

# l warp

LATEXHTML5

## The l warp package

v0.23 — 2017/03/02

LATEX to HTML5

© 2016–2017 Brian Dunn  
[bd@BDTechConcepts.com](mailto:bd@BDTechConcepts.com)

### Abstract

The l warp package allows LATEX to directly produce HTML5 output, using external utility programs only for the final conversion of text and images. Math may be represented by SVG files or MathJax.

Documents may be produced by LATEX, LuaLATEX, or XeLATEX. A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `texmfk`. Configuration is automatic at the first manual compile.

Print and HTML versions of each document may coexist, each with its own set of auxiliary files. Support files are self-generated on request.

A modular package-loading system uses the l warp version of a package for HTML when available. Several dozen LATEX packages are supported with these high-level source compatibility replacements.

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

To update existing projects, see section 1, Updates.

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

### License:

This work may be distributed and/or modified under the conditions of the LaTeP 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 LaTeP version 2005/12/01 or later.

# Contents

<b>1</b>	<b>l warp.sty</b>	<b>17</b>
<b>1</b>	<b>Updates</b>	<b>17</b>
<b>2</b>	<b>Introduction</b>	<b>20</b>
2.1	Supported packages and features . . . . .	21
<b>3</b>	<b>Alternatives</b>	<b>25</b>
3.1	Internet class . . . . .	25
3.2	TeX4ht . . . . .	25
3.3	Translators . . . . .	25
3.4	AsciiDoc . . . . .	26
3.5	Pandoc . . . . .	26
3.6	Word Processors . . . . .	26
3.7	Commercial Systems . . . . .	27
3.8	Comparisons . . . . .	27
<b>4</b>	<b>Installation</b>	<b>28</b>
4.1	Installing the <code>l warp</code> package . . . . .	28
4.2	Installing the <code>l warpmk</code> utility . . . . .	31
4.2.1	Using a local copy of <code>l warpmk</code> . . . . .	32
4.3	Installing additional utilities . . . . .	33
<b>5</b>	<b>Tutorial</b>	<b>35</b>
5.1	Starting a new project . . . . .	35
5.2	Compiling the print version with <code>l warpmk</code> . . . . .	39
5.3	Compiling the HTML version with <code>l warpmk</code> . . . . .	40
5.4	Generating the SVG images . . . . .	41
5.5	Using MathJax for math . . . . .	42
5.6	Changing the CSS style . . . . .	43
5.7	Customizing the HTML output . . . . .	44
5.8	Using <code>latexmk</code> . . . . .	48
5.9	Using XeLaTeX or LuaLaTeX . . . . .	49
5.10	Cleaning auxiliary files . . . . .	50
5.11	Cleaning auxiliary and output files . . . . .	50
5.12	Processing multiple projects in the same directory . . . . .	50
5.13	Using the <code>make</code> utility . . . . .	50
<b>6</b>	<b>Additional details</b>	<b>51</b>
6.1	Font and UTF-8 support . . . . .	51
6.2	<code>l warp</code> package loading and options . . . . .	52
6.3	Selecting the operating system . . . . .	52
6.4	Selecting actions for print or HTML output . . . . .	52
6.5	Commands to be placed into the <code>warpprint</code> environment . . . . .	53

6.6	Commands for a successful HTML conversion . . . . .	54
6.7	Title page . . . . .	55
6.8	HTML page meta descriptions . . . . .	56
6.9	CSS . . . . .	56
6.10	Special cases and limitations . . . . .	57
6.10.1	Text formatting . . . . .	57
6.10.2	Math . . . . .	57
6.10.3	Graphics . . . . .	59
6.10.4	Xcolor . . . . .	60
6.10.5	Tikz . . . . .	60
6.10.6	Tabular . . . . .	60
6.10.7	Minipages . . . . .	61
6.10.8	Mdframed . . . . .	62
6.10.9	float, trivfloat, and/or algorithmicx together . . . . .	62
6.10.10	caption and subcaption packages . . . . .	63
6.10.11	floatrow package . . . . .	63
6.10.12	abstract package . . . . .	63
6.10.13	cleveref and varioref packages . . . . .	63
6.10.14	verse and memoir . . . . .	63
6.10.15	siunitx package . . . . .	64
6.10.16	newclude package . . . . .	64
6.10.17	newtxmath package . . . . .	64
6.10.18	babel package . . . . .	65
6.10.19	enumitem package . . . . .	65
6.10.20	enumerate package . . . . .	65
7	<b>Expanding the code base</b>	65
7.1	Creating an lwarf version of a package . . . . .	66
8	<b>Troubleshooting</b>	67
8.1	Using the lwarf.sty package . . . . .	67
8.1.1	Debug tracing output . . . . .	68
8.2	Compiling the lwarf.dtx file . . . . .	69
9	<b>Implementation</b>	69
10	<b>Stack depths</b>	70
11	<b>Source Code</b>	71
12	<b>Detecting the T<sub>E</sub>X Engine — pdflatex, lualatex, xelatex</b>	72
13	<b>Unicode Input Characters</b>	72
14	<b>Handling package options</b>	72
15	<b>Misplaced packages</b>	75

<b>16</b>	<b>Required packages</b>	<b>77</b>
<b>17</b>	<b>Loading Packages</b>	<b>82</b>
<b>18</b>	<b>Copying a file</b>	<b>85</b>
<b>19</b>	<b>Debugging messages</b>	<b>85</b>
<b>20</b>	<b>Remembering original formatting macros</b>	<b>86</b>
<b>21</b>	<b>Operating-System portability</b>	<b>87</b>
21.1	Common portability code . . . . .	87
21.2	Unix, Linux, and Mac OS . . . . .	87
21.3	MS-Windows . . . . .	87
<b>22</b>	<b>Stacks</b>	<b>88</b>
22.1	Assigning depths . . . . .	88
22.2	Closing actions . . . . .	89
22.3	Closing depths . . . . .	89
22.4	Pushing and popping the stack . . . . .	90
<b>23</b>	<b>Data arrays</b>	<b>91</b>
<b>24</b>	<b>HTML entities</b>	<b>92</b>
<b>25</b>	<b>HTML filename generation</b>	<b>93</b>
<b>26</b>	<b>Homepage link</b>	<b>95</b>
<b>27</b>	<b>\PrintStack diagnostic tool</b>	<b>96</b>
<b>28</b>	<b>Closing stack levels</b>	<b>96</b>
<b>29</b>	<b>HTML tags, spans, divs, elements</b>	<b>97</b>
29.1	Mapping L <sup>A</sup> T <sub>E</sub> X Sections to HTML Sections . . . . .	97
29.2	HTML tags . . . . .	98
29.3	Block tags and comments . . . . .	99
29.4	Div class and element class . . . . .	100
29.5	Single-line elements . . . . .	101
29.6	HTML5 semantic elements . . . . .	102
29.7	High-level block and inline classes . . . . .	102
29.8	Closing HTML tags . . . . .	103
<b>30</b>	<b>Paragraph handling</b>	<b>104</b>
<b>31</b>	<b>Paragraph start/stop handling</b>	<b>107</b>
<b>32</b>	<b>Page headers and footers</b>	<b>110</b>

<b>33 CSS</b>	<b>111</b>
<b>34 HTML meta description</b>	<b>111</b>
<b>35 Footnotes</b>	<b>112</b>
35.1 Regular page footnotes . . . . .	112
35.2 Minipage footnotes . . . . .	112
35.3 Titlepage thanks . . . . .	112
35.4 Regular page footnote implementation . . . . .	113
<b>36 Marginpars</b>	<b>114</b>
<b>37 Splitting HTML files</b>	<b>115</b>
<b>38 Sectioning</b>	<b>119</b>
38.1 Book class commands . . . . .	120
38.2 Sectioning support macros . . . . .	120
38.3 \section and friends . . . . .	125
<b>39 Starting a new file</b>	<b>127</b>
<b>40 Starting HTML output</b>	<b>129</b>
<b>41 Ending HTML output</b>	<b>132</b>
<b>42 Titles and the <i>titling</i> package</b>	<b>134</b>
42.1 Setting the title, etc. . . . .	135
42.2 Changes to HTML titlepage and <i>titlingpage</i> . . . . .	137
42.3 Printing the title, etc. in HTML . . . . .	138
42.4 Printing the title, etc. in print form . . . . .	141
42.5 \maketitle for print output . . . . .	142
42.6 \maketitle for HTML output . . . . .	144
<b>43 Abstract</b>	<b>147</b>
<b>44 Quote and verse</b>	<b>148</b>
44.1 Citations and attributions . . . . .	148
44.2 Quotes, quotations . . . . .	148
44.3 Verse . . . . .	149
<b>45 Verbatim</b>	<b>150</b>
<b>46 Fancyvrb</b>	<b>150</b>
<b>47 Theorems</b>	<b>156</b>
<b>48 Lists</b>	<b>157</b>
48.1 Itemize . . . . .	157

48.2	Enumerate . . . . .	158
48.3	Description . . . . .	159
<b>49</b>	<b>Tabular</b>	<b>160</b>
49.1	Token lookahead . . . . .	161
49.2	Booleans . . . . .	162
49.3	Handling & and ! . . . . .	163
49.4	Handling \\ . . . . .	165
49.5	Variables . . . . .	166
49.6	Parsing @, >, <, ! columns . . . . .	167
49.7	Parsing 'l', 'c', or 'r' columns . . . . .	169
49.8	Parsing 'p', 'm', or 'b' columns . . . . .	169
49.9	Parsing 'D' columns . . . . .	170
49.10	Parsing the column specifications . . . . .	170
49.11	Starting a new row . . . . .	174
49.12	Data opening tag . . . . .	174
49.13	Midrules . . . . .	176
49.14	Multicolumns . . . . .	177
49.14.1	Parsing multicolumns . . . . .	177
49.14.2	High-level multicolumn interface . . . . .	180
49.14.3	Longtable captions . . . . .	180
49.14.4	\tabledata{multicolumn}{tag} . . . . .	183
49.15	Multirow . . . . .	183
49.16	Utility macros inside a table . . . . .	184
49.17	Checking for a new table cell . . . . .	184
49.18	\mrowcell . . . . .	185
49.19	New \tabular definition . . . . .	186
49.20	Array . . . . .	188
<b>50</b>	<b>Cross-references</b>	<b>188</b>
50.1	Setup . . . . .	188
50.2	Zref setup . . . . .	190
50.3	Labels . . . . .	191
50.4	References . . . . .	193
50.5	Hyper-references . . . . .	195
<b>51</b>	<b>Floats</b>	<b>198</b>
51.1	Float captions . . . . .	199
51.1.1	Caption inside a float environment . . . . .	200
51.1.2	Caption and LOF linking and tracking . . . . .	200
<b>52</b>	<b>Table of Contents, LOF, LOT</b>	<b>203</b>
52.1	Reading and printing the TOC . . . . .	204
52.2	High-level TOC commands . . . . .	206
52.3	Side TOC . . . . .	206
52.4	Low-level TOC line formatting . . . . .	208

<b>53 Index</b>	<b>209</b>
<b>54 Math</b>	<b>211</b>
54.1 Inline and display math . . . . .	214
54.2 MathJax support . . . . .	215
54.3 Equation environment . . . . .	218
54.4 AMS Math environments . . . . .	220
54.4.1 Support macros . . . . .	220
54.4.2 Environment patches . . . . .	221
<b>55 Lateximages</b>	<b>225</b>
<b>56 center, flushleft, flushright</b>	<b>231</b>
<b>57 Siunitx</b>	<b>232</b>
<b>58 Graphics</b>	<b>232</b>
58.1 \graphicspath . . . . .	233
58.2 Length conversions and graphics options . . . . .	233
58.3 \includegraphics . . . . .	236
58.4 \rotatebox, \scalebox, \reflectbox . . . . .	240
58.5 Null functions . . . . .	242
<b>59 Cleverref</b>	<b>243</b>
<b>60 Picture</b>	<b>244</b>
<b>61 Tikzpicture</b>	<b>245</b>
<b>62 Boxes and Minipages</b>	<b>246</b>
62.1 Counters and lengths . . . . .	246
62.2 Footnote handling . . . . .	247
62.3 Minipage handling . . . . .	248
62.4 Parbox, makebox, framebox, fbox, raisebox . . . . .	251
<b>63 Direct formatting</b>	<b>252</b>
<b>64 Skips, spaces, font sizes</b>	<b>255</b>
<b>65 \phantomsection</b>	<b>262</b>
<b>66 \LaTeX and other logos</b>	<b>262</b>
66.1 HTML logos . . . . .	262
66.2 Print logos . . . . .	264
<b>67 \AtBeginDocument, \AtEndDocument</b>	<b>265</b>
<b>68 lwarpmk.conf</b>	<b>266</b>

69 Trademarks	267
<b>2 l warp-newproject.sty</b>	<b>268</b>
70 l warp-newproject	268
70.1 project_html.tex . . . . .	270
70.2 project.lwarpmkconf . . . . .	270
70.3 l warp.css . . . . .	271
70.4 l warp_sagebrush.css . . . . .	291
70.5 l warp_formal.css . . . . .	296
70.6 sample_project.css . . . . .	301
70.7 l warp_html.xdy . . . . .	301
70.8 l warp_mathjax.txt . . . . .	301
70.9 Lwarpmk option . . . . .	303
<b>3 l warp-abstract.sty</b>	<b>313</b>
71 Abstract	313
<b>4 l warp-afterpage.sty</b>	<b>315</b>
72 Afterpage	315
<b>5 l warp-algorithmicx.sty</b>	<b>316</b>
73 Algorithmicx	316
<b>6 l warp-alltt.sty</b>	<b>316</b>
74 Alltt	317
<b>7 l warp-bookmark.sty</b>	<b>317</b>
75 Bookmark	317
<b>8 l warp-booktabs.sty</b>	<b>318</b>
76 Booktabs	318

<b>9 lwarf-changepage.sty</b>	<b>318</b>
<b>77 Changepage</b>	<b>318</b>
<b>10 lwarf-dcolumn.sty</b>	<b>318</b>
<b>78 Dcolumn</b>	<b>319</b>
<b>11 lwarf-draftwatermark.sty</b>	<b>319</b>
<b>79 Draftwatermark</b>	<b>319</b>
<b>12 lwarf-enumerate.sty</b>	<b>320</b>
<b>80 Enumerate</b>	<b>320</b>
<b>13 lwarf-epigraph.sty</b>	<b>321</b>
<b>81 Epigraph</b>	<b>321</b>
<b>14 lwarf-eso-pic.sty</b>	<b>322</b>
<b>82 Eso-pic</b>	<b>322</b>
<b>15 lwarf-everypage.sty</b>	<b>323</b>
<b>83 Everypage</b>	<b>323</b>
<b>16 lwarf-extramarks.sty</b>	<b>323</b>
<b>84 Extramarks</b>	<b>323</b>
<b>17 lwarf-fancyhdr.sty</b>	<b>323</b>
<b>85 Fancyhdr</b>	<b>324</b>
<b>18 lwarf-float.sty</b>	<b>325</b>
<b>86 Float and \newfloat</b>	<b>325</b>

<b>19 lwarf-floatrow.sty</b>	<b>327</b>
<b>87 Floatrow</b>	<b>327</b>
<b>20 lwarf-fontenc.sty</b>	<b>332</b>
<b>88 Fontenc</b>	<b>332</b>
<b>21 lwarf-fontspec.sty</b>	<b>332</b>
<b>89 Fontspec</b>	<b>332</b>
<b>22 lwarf-ftnright.sty</b>	<b>333</b>
<b>90 Ftnright</b>	<b>333</b>
<b>23 lwarf-geometry.sty</b>	<b>333</b>
<b>91 Geometry</b>	<b>333</b>
<b>24 lwarf-graphics.sty</b>	<b>334</b>
<b>92 Graphics</b>	<b>334</b>
<b>25 lwarf-graphicx.sty</b>	<b>334</b>
<b>93 Graphicx</b>	<b>334</b>
<b>26 lwarf-hyperref.sty</b>	<b>335</b>
<b>94 Hyperref</b>	<b>335</b>
<b>27 lwarf-indentfirst.sty</b>	<b>338</b>
<b>95 Indentfirst</b>	<b>338</b>
<b>28 lwarf-inputenc.sty</b>	<b>338</b>
<b>96 Inputenc</b>	<b>338</b>

<b>29 l warp-keyfloat.sty</b>	<b>339</b>
<b>97 Keyfloat</b>	<b>339</b>
<b>30 l warp-layout.sty</b>	<b>339</b>
<b>98 Layout</b>	<b>340</b>
<b>31 l warp-letterspace.sty</b>	<b>340</b>
<b>99 Letterspace</b>	<b>340</b>
<b>32 l warp-listings.sty</b>	<b>341</b>
<b>100 Listings</b>	<b>341</b>
<b>33 l warp-longtable.sty</b>	<b>345</b>
<b>101 Longtable</b>	<b>345</b>
<b>34 l warp-lscape.sty</b>	<b>347</b>
<b>102 Lscape</b>	<b>347</b>
<b>35 l warp-ltcaption.sty</b>	<b>347</b>
<b>103 Ltcaption</b>	<b>347</b>
<b>36 l warp-mcaption.sty</b>	<b>347</b>
<b>104 Mcaption</b>	<b>348</b>
<b>37 l warp-mdframed.sty</b>	<b>349</b>
<b>105 Mdframed</b>	<b>349</b>
<b>38 l warp-microtype.sty</b>	<b>356</b>
<b>106 Microtype</b>	<b>356</b>

<b>39 l warp-multicol.sty</b>	<b>357</b>
<b>107 Multicol</b>	<b>357</b>
<b>40 l warp-multirow.sty</b>	<b>359</b>
<b>108 Multirow</b>	<b>359</b>
<b>41 l warp-nameref.sty</b>	<b>359</b>
<b>109 Nameref</b>	<b>359</b>
<b>42 l warp-needspace.sty</b>	<b>360</b>
<b>110 Needspace</b>	<b>360</b>
<b>43 l warp-newclude.sty</b>	<b>360</b>
<b>111 Newclude</b>	<b>360</b>
<b>44 l warp-newunicodechar.sty</b>	<b>361</b>
<b>112 Newunicodechar</b>	<b>361</b>
<b>45 l warp-nextpage.sty</b>	<b>361</b>
<b>113 Nextpage</b>	<b>361</b>
<b>46 l warp-nowidow.sty</b>	<b>361</b>
<b>114 Nowidow</b>	<b>362</b>
<b>47 l warp-pagenote.sty</b>	<b>362</b>
<b>115 Pagenote</b>	<b>362</b>
<b>48 l warp-parskip.sty</b>	<b>363</b>
<b>116 Parskip</b>	<b>363</b>

<b>49 l warp-placeins.sty</b>	<b>363</b>
<b>117 Placeins</b>	<b>363</b>
<b>50 l warp-ragged2e.sty</b>	<b>364</b>
<b>118 Ragged2e</b>	<b>364</b>
<b>51 l warp-rotating.sty</b>	<b>365</b>
<b>119 Rotating</b>	<b>365</b>
<b>52 l warp-setspace.sty</b>	<b>366</b>
<b>120 Setspace</b>	<b>366</b>
<b>53 l warp-showkeys.sty</b>	<b>366</b>
<b>121 Showkeys</b>	<b>367</b>
<b>54 l warp-sidecap.sty</b>	<b>368</b>
<b>122 Sidecap</b>	<b>368</b>
<b>55 l warp-tabularx.sty</b>	<b>369</b>
<b>123 Tabularx</b>	<b>369</b>
<b>56 l warp-textpos.sty</b>	<b>370</b>
<b>124 Textpos</b>	<b>370</b>
<b>57 l warp-threeparttable.sty</b>	<b>371</b>
<b>125 Threeparttable</b>	<b>371</b>
<b>58 l warp-titleps.sty</b>	<b>372</b>
<b>126 Titleps</b>	<b>372</b>

<b>59 lwarf-titlesec.sty</b>	<b>374</b>
<b>127 Titlesec</b>	<b>374</b>
<b>60 lwarf-tocloft.sty</b>	<b>375</b>
<b>128 Tocloft</b>	<b>375</b>
<b>61 lwarf-titling.sty</b>	<b>376</b>
<b>129 Titling</b>	<b>376</b>
<b>62 lwarf-trivfloat.sty</b>	<b>377</b>
<b>130 Tocloft</b>	<b>377</b>
<b>63 lwarf-trivfloat.sty</b>	<b>382</b>
<b>131 Trivfloat</b>	<b>382</b>
131.1 Combining \newfloat, \trivfloat, and algorithmicx . . . . .	382
<b>64 lwarf-verse.sty</b>	<b>384</b>
<b>132 Verse</b>	<b>384</b>
<b>65 lwarf-wallpaper.sty</b>	<b>386</b>
<b>133 Wallpaper</b>	<b>386</b>
<b>66 lwarf-wrapfig.sty</b>	<b>387</b>
<b>134 Wrapfig</b>	<b>387</b>
<b>67 lwarf-xcolor.sty</b>	<b>389</b>
<b>135 Xcolor</b>	<b>389</b>

**68 l warp-xfrac.sty** 392

**136 Xfrac** 392

**Change History and Index** 395

## List of Figures

1	tutorial.tex listing	36
---	----------------------	----

## List of Tables

1	L <sup>A</sup> T <sub>E</sub> X–HTML generation — l warp package — Supported functions	21
2	Additional supported packages	24
3	Required software programs	29
4	Files created in a new project	38
5	Section depths and HTML headings	70
6	Tabular baseline	171
7	Tabular HTML column conversions	171
8	Cross-referencing data structures	189
9	Float data structures	198

## Package 1

### **l warp.sty**

#### **1 Updates**

For a detailed list of changes, see the Change History on page [395](#).

##### **v0.22:**

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

##### **v0.21:**

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

##### **v0.20:**

- The `makefile` and related infrastructure has been replaced by the `l warpmk` utility. This provides increased portability, reduced dependencies, and much simpler installation and setup.
- The `l warp-newproject` package is now used to locally create support files.
- The print and HTML versions of a document may co-exist with their own sets of auxiliary files.
- Package handling is now controlled by a modular system which looks for and loads an `l warp-<package>` version if available.

- High-level source compatibility is provided for all supported packages, almost totally eliminating the need for `warpprint` and `warpHTML` environments.
- A large number of additional packages are supported.
- A new tutorial is included in the documentation, and many obsolete sections have been removed.
- `\NewHTMLdescription` sets the HTML meta description tag for each file. See section 6.8.
- `\HTMLFileName` may now be empty, allowing filenames without a prefix. Lwarp no longer automatically appends a - character. For existing projects, add a - to the end of `\HTMLFileName`.
- ⚠️ • `\HomeHTMLFileName` and `\HTMLFileName` no longer use escaped underscore characters. Underscores may be used in filenames as-is.
- ⚠️ • Lwarp now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows.
- Tabular column types @, >, and < are now supported.
- `BlockClass` and `\InlineClass` add an optional style.
- The `sidebar` and `example` environments have been moved to the test suite, and are no longer included in `l warp`.

#### v0.19:

- ⚠️ • MATHJAX now may be used to display math via the `mathjax` option. See sections 6.2 and 6.10.2. To use MathJax with a pre-existing project, copy or link the file `l warp_mathjax.txt` to the project's directory.
- `\rule` added, supporting width, height, raise, `\textcolor`.
- `\LateximageFontSizeName` provides user-adjustable font size for math and `lateximages`.
- `\minipagefullwidth` requests that the next minipage be full-width in HTML, but still the assigned width in print.
- `minipage` improved side-by-side rendering.
- CSS class `tablenotes` is provided for table note items.
- ⚠️ • `\warpprintonly` replaces `\rowprintedonly`, and `\warpHTMLonly` is added. These behave like the `warpprint` and `warpHTML` environments, and are generally useful, so they replace the previously table-specific syntax.
- `cleveref` is loaded `\AtEndPreamble` for improved reliability. See section 59.
- `\xfracHTMLfontsize` controls `xfrac` font size in HTML.
- `Tikz` improved catcode handling.

**v0.18:**

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

**v0.17:**

- `mdframed` package is supported.

**v0.16:**

- Font and input encoding are now controlled by the user, and `lwarp` is loaded after fonts have been selected.
- Support for X<sub>E</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X. See section 6.1.

## 2 Introduction

The lwarf project aims to allow a rich L<sup>A</sup>T<sub>E</sub>X document to be converted to a reasonable HTML interpretation. No attempt has been made to force L<sup>A</sup>T<sub>E</sub>X to provide for every HTML-related possibility, and HTML cannot exactly render every possible L<sup>A</sup>T<sub>E</sub>X concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

Several “modern” features of HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

`pdflatex`, `xelatex`, or `lualatex` is used, allowing lwarf to process the usual image formats. While generating HTML output, SVG files are used in place of PDF. Other formats such as JPG are used as-is.

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments, as this format has better browser and e-book support than MathML (as of this writing), while still allowing for the high-quality display and printing of images (again, subject to potentially bug-ridden<sup>1</sup> browser support).

Furthermore, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by ALT tags holding the L<sup>A</sup>T<sub>E</sub>X source for the expression, allowing it to be copy/pasted into other documents.<sup>2</sup> Custom L<sup>A</sup>T<sub>E</sub>X macros may be used as-is in math expressions, since the math is evaluated entirely inside L<sup>A</sup>T<sub>E</sub>X.

The MATHJAX JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files, one per math expression. lwarf maintains L<sup>A</sup>T<sub>E</sub>X control for cross-referencing and equation numbering / formatting.

---

The lwarf package allows L<sup>A</sup>T<sub>E</sub>X to directly generate HTML5 tags from a L<sup>A</sup>T<sub>E</sub>X source document, with only minor intervention on the user’s part. A `texlua` program called `lwarpmk` is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from

---

<sup>1</sup>Firefox has had an on-again/off-again bug for quite some time regarding printing svgs at high resolution.

<sup>2</sup>There seems to be some debate as to whether MathML is actually an improvement over L<sup>A</sup>T<sub>E</sub>X for sharing math. The author has no particular opinion on the matter, except to say that in this case L<sup>A</sup>T<sub>E</sub>X is much easier to implement!

a L<sup>A</sup>T<sub>E</sub>X-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

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

When requesting packages during HTML conversion, `l warp` first looks to see if it has its own modified version to use instead of the usual L<sup>A</sup>T<sub>E</sub>X version. These `l warp-pagename.sty` files contain code used to emulate or replace functions for HTML output.

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 21.)

## 2.1 Supported packages and features

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

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

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

Package or Macro	Status
Engines:	pdfL <sup>A</sup> T <sub>E</sub> X, X <sub>H</sub> L <sup>A</sup> T <sub>E</sub> X, LuaL <sup>A</sup> T <sub>E</sub> X
Classes:	<code>book</code> , <code>report</code> , or <code>article</code> . <code>memoir</code> is planned.
Sectioning:	Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>filedepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames.

## l warp Supported Functions — continued

Package or Macro	Status
Table of Contents, Figures, Tables:	Supported, with hyperlinks.
\maketitle, titlepage, titling:	Supported, with optional <code>titling</code> -based commands for published and subtitle.
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleveref</code> com- mands, and index entries.
Footnotes:	Supported, emulated as pagenotes per HTML page.
Indexing:	<code>texindy</code> is used, with hyperlinks.
Bibliography:	Supported, without hyperlinks so far.
Math:	Supported. Converted to SVG images with HTML ALT tags containing the L <sup>A</sup> T <sub>E</sub> X source for the math expression. MathJax supported as an alternative. <i>AMS</i> environments are supported. User-defined macros are available during conversion, due to native L <sup>A</sup> T <sub>E</sub> X processing.
Floats:	Supported, appear where declared. <code>float</code> is emu- lated, <code>newfloat</code> is supported, <code>caption</code> and <code>subcap- tion</code> are supported, <code>capt-of</code> is supported, <code>placeins</code> is ignored, <code>trivfloat</code> is supported, <code>floatrow</code> is emu- lated, <code>keyfloat</code> is supported, <code>wrapfig</code> is emulated.
<code>tabular</code> :	Emulated. <code>\multirow</code> and <code>\multicolumn</code> are available, but cannot be used at the same time. Nested tables are not supported.
<code>array</code> :	Supported inside math environments, emulated elsewhere.
<code>multirow</code> :	Emulated.
<code>longtable</code> :	Emulated. Converted to a tabular. Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>

## l warp Supported Functions — continued

Package or Macro	Status
booktabs:	Emulated. \toprule and \bottomrule form black rules, \midrule forms silver rules, as demonstrated on this table. \cmidrule, demonstrated at this line, does not use width or trim options.
threeparttable:	Emulated
graphics, graphicx:	Emulated. \includegraphics supports width, height, origin, angle, and scale tags, and adds class. References to PDF files are changed to SVG, other image types are accepted as well. \rotatebox and \scalebox are supported as well as HTML can handle.
rotating:	Emulated. All objects are displayed unrotated.
Lists:	Supported
enumitem:	Supported, although spacing is still controlled by CSS.
Environments:	Standard L <sup>A</sup> T <sub>E</sub> X environments are supported.
theorem:	Supported, with added css tags.
picture and tikz:	Converted to an SVG image.
minipage:	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
fancyvrb:	Supported except for verbatim footnotes.
mdframed:	Supported
multicol:	Emulated, with css3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
siunitx:	Supported except for per-mode=fraction.
xfrac:	Supported
Direct formatting:	\emph, \textsuperscript, \textbf, etc are supported. \bfseries, etc. are not yet supported.

## l warp Supported Functions — continued

Package or Macro	Status
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, <code>\enskip</code> , <code>\quad</code> , <code>\qquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&amp;</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
xcolor:	<b>Supported.</b> Full package color names, any color models, and <code>mixing</code> is converted to hex web colors via <code>\convertcolorspec</code> . Patched commands are <code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> . <code>\pagecolor</code> is not supported.
Where:	
<b>Supported:</b>	The existing L <sup>A</sup> T <sub>E</sub> X package is used.
<b>Emulated:</b>	The L <sup>A</sup> T <sub>E</sub> X package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

Supported packages include everything listed in the table of contents, plus each of the following in table 2, and certainly others as well. Not all have been thoroughly tested. Many are simply nullified during HTML output. Others are not affected by the output mode and thus work as-is.

These packages and features probably works with little or no change to the user's source code. Special environments are provided to mark blocks of code which are for print only, HTML only, or both, should it be necessary.

Table 2: Additional supported packages

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

### 3 Alternatives

Summarized below are several other ways to convert a L<sup>A</sup>T<sub>E</sub>X or other document to HTML. Where an existing L<sup>A</sup>T<sub>E</sub>X document is to be converted to HTML, lwarf may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

#### 3.1 Internet class

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

#### 3.2 TeX4ht

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

This system uses native L<sup>A</sup>T<sub>E</sub>X processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In some cases, lwarf provides a better HTML conversion, and it supports a different set of packages. TeX4ht produces several other forms of output beyond HTML.

#### 3.3 Translators

These systems use external programs to translate a subset of L<sup>A</sup>T<sub>E</sub>X syntax into HTML. Search for each on CTAN (<http://ctan.org>).

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

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

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

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

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

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

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

Finally, Glad<sub>TEX</sub> may used to directly insert L<sub>A</sub>T<sub>E</sub>X math into HTML:

Prog Glad<sub>TeX</sub> Glad<sub>TEX</sub>: <http://humenda.github.io/GladTeX/>

### 3.4 AsciiDoc

AsciiDoc is one of the most capable markup languages, providing enough features to produce the typical technical-writing document with cross-references, and it writes L<sub>A</sub>T<sub>E</sub>X and HTML.

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

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

The Asciidoctor-LaTeX project is adding additional L<sub>A</sub>T<sub>E</sub>X-related features.

#### Asciidoctor-LateX:

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

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

### 3.5 Pandoc

Prog Pandoc

A markup system which also reads and writes L<sub>A</sub>T<sub>E</sub>X and HTML.

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

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

### 3.6 Word Processors

Prog Word It should be noted that the popular word processors have advanced through the

Prog LibreOffice years in their abilities to represent math with a L<sub>A</sub>T<sub>E</sub>X-ish input syntax, unicode  
Prog OpenOffice math fonts, and high-quality output, and also generate HTML with varying success.

See recent developments in Microsoft® Word® and LibreOffice™ Writer.

### 3.7 Commercial Systems

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

Prog    **FrameMaker**

Prog    **InDesign**

Prog    **Flare**

Prog    **Madcap**

### 3.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X is comparably easy to learn, while L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X back end, yielding high-quality results especially when the L<sup>A</sup>T<sub>E</sub>X template is adjusted, but they lose the ability to use L<sup>A</sup>T<sub>E</sub>X macros and other L<sup>A</sup>T<sub>E</sub>X source-document features.

The effort required to customize the output of each markup system varies. For print output, L<sup>A</sup>T<sub>E</sub>X configuration files are usually used. For HTML output, a CSS file will be available, but additional configuration may require editing some form of control file with a different syntax, such as XML. In the case of lwarf, CSS is used, and much HTML output is adjusted through the usual L<sup>A</sup>T<sub>E</sub>X optional macro parameters, but further customization may require patching L<sup>A</sup>T<sub>E</sub>X code.

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

## 4 Installation

Table 3 shows the tools which are used for the L<sup>A</sup>T<sub>E</sub>X to HTML conversion. In most cases, these will be available via the standard package-installation tools.

### 4.1 Installing the lwarf package

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

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

Enter ⇒ `kpsewhich lwarf.sty`

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

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

Enter ⇒ `tlmgr install lwarf`

**MiK<sub>T</sub>E<sub>X</sub>:** If using MiK<sub>T</sub>E<sub>X</sub>, try using the package installer.

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

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

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

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

Enter ⇒ `kpsewhich -var-value TEXMFLOCAL`

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

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

This should be something like:

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

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

Enter ⇒ `mktexlsr` — or —

Enter ⇒ `texhash`

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.

Table 3: Required software programs

**Provided by your L<sup>A</sup>T<sub>E</sub>X distribution:**

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

**L<sup>A</sup>T<sub>E</sub>X:** pdflatex, xelatex, or lualatex.

**The lwarf package:** This package.

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

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

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

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

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

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

**Poppler PDF utilities:**

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

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

For Windows, also see:

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

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

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

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

**Perl:**

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

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

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

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

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

**CTAN .dtx and .ins files:** Another form of TeX package is the .dtx and .ins source files. These files are used to create the documentation and .sty files.

1. See <http://ctan.org/pkg/lwarp> for the l warp 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 TeX Live local tree found in the previous CTAN TDS section, under the subdirectory:  
`<texlocal>/tex/latex/local/lwarp`
8. Copy the documentation `lwarp.pdf` to a source directory in the local tree, such as:  
`<texlocal>/doc/local/lwarp`
9. Renew the cache:  
Enter ⇒ `mktexlsr` — or —  
Enter ⇒ `texhash`  
Or, for Windows MiKTeX, start the program called **MiKTeX Settings (Admin)** and click on the button called **Refresh FNDB**.
10. See section 4.2.1 to generate your local copy of `lwarpmk`.
11. Once the local version of `lwarpmk.lua` is installed, it may be made available system-wide as per section 4.2.

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

Just testing!

## 4.2 Installing the `lwarpmk` utility

(Note: If `lwarpmk` is not already installed, it is easiest to use a local copy instead of installing it system-wide. See section 4.2.1.)

After the `l warp` package is installed, you may need to setup the `lwarpmk` utility:

1. At a command line, try executing `lwarpmk`. If the `lwarpmk` help message appears, then `lwarpmk` is already set up. If not, it is easiest to generate and use a local copy. See section 4.2.1.

Otherwise, continue with the following:

2. Locate the file `lwarpmk.lua`, which should be in the `scripts` directory of the TDS tree. On a TeX Live or MiKTeX system you may use

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

3. Create `lwarpmk`:

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

- (a) Locate the TeX Live binaries:

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

This will be something like:

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

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

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

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

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

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

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

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

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

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

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

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

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

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

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

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

#### 4.2.1 Using a local copy of `lwarpmk`

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

1. When compiling the tutorial in section 5, use the `lwarpmk` option for the `l warp-newproject` package:  
`\usepackage[lwarpmk]{l warp-newproject}`
2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.
3. `lwarpmk.lua` may be used for this project:

##### Unix:

- (a) Make `lwarpmk.lua` executable:

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

- (b) Compile documents with

```
Enter ⇒ ./lwarpmk.lua html
```

```
Enter ⇒ ./lwarpmk.lua print etc.
```

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

##### Windows:

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

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

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

Or:

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print etc.
```

### 4.3 Installing additional utilities

The TeX utilities `latexmk` and `pdfcrop` may be provided by the installation of additional TeX-relatex operating-system packages.

The tools from the POPPLER project may also be provided by your operating system's package-installation tools. For Windows, see table 3 on page 29.

Perl may require installation as well. See table 3.

**To test for the existence of the additional utilities:** Enter the following in a command line:

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

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

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

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

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

8. Click "Ok" when done.

Prog perl

**To add 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.

## 5 Tutorial

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

### 5.1 Starting a new project

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

#### **⚠ Note: .txt suffix!**

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

3. Compile the project:

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

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

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

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created while the `lwarp-newproject` package is used, the configuration files for `lwarpmk` are updated to record the operating system, L<sup>A</sup>T<sub>E</sub>X program (`pdflatex`, `xelatex`, or `lualatex`), the filenames of the source code and HTML output, and whether the additional helper program `latexmk` will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

% --- LOAD FONT SELECTION AND ENCODING BEFORE LOADING LWARP ---
\ifPDFTeX
\usepackage{lmodern}           % pdflatex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\else
\usepackage{fontspec}          % XeLaTeX or LuaLaTeX
\fi

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

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

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

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

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarf Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2}        % Include subsections in the TOC.
\setcounter{secnumdepth}{2}      % Number down to subsections.
\setcounter{FileDepth}{1}        % Split HTML files at sections
\booltrue{CombineHigherDepths} % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1}     % Include subsections in the sideTOC
\SetFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\SetPageTop{\fbox{LOGO}}
\SetPageBottom{Contact Information and Copyright}
\NewCSS{lwarf_sagebrush.css}
```

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

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

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}
```

Table 4: Files created in a new project

**tutorial.pdf:** The PDF output from L<sup>A</sup>T<sub>E</sub>X.

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

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

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

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

**.css files:** `lwarp.css`, `lwarp_formal.css`, `lwarp_sagebrush.css` These files are standard for `lwarp`, and are not meant to be modified by the user.

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

**lwarp\_html.xdy:** Used by `lwarp` while creating an index. This file should not need to be modified by the user.

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

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

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

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

On Unix-related operating systems this file must be made executable:  
`chmod u+x lwarpmk.lua`

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

## 5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk print`

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

5. Process the index.

Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

Enter ⇒ `lwarpmk print`

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

---

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

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

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

1. Compile the HTML version:

Enter ⇒ `lwarpmk html`

- (a) `lwarpmk` uses L<sup>A</sup>T<sub>E</sub>X to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert into the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`<sup>4</sup>, `Some-math.html`, and the document's index in `_Index.html`<sup>5</sup>.

2. View the homepage in a web browser.

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

`math`

Note that math is still displayed as its plain-text L<sup>A</sup>T<sub>E</sub>X source until the images of the math expressions have not yet been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 5.4 and 5.5.

3. Force a recompile:

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile:

Enter ⇒ `lwarpmk htmlindex`

Enter ⇒ `lwarpmk html`

`_Index.html`, is updated for the new L<sup>A</sup>T<sub>E</sub>X index.

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

---

<sup>4</sup> `First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

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

## 5.4 Generating the SVG images

### math as SVG images

By default `l warp` represents math as SVG images with the L<sup>A</sup>T<sub>E</sub>X source included in `alt` tags. In this way, the math displays as it was drawn by L<sup>A</sup>T<sub>E</sub>X, and the L<sup>A</sup>T<sub>E</sub>X source may be copied and pasted into some other document.

### picture and Tikz

`l warp` uses the same mechanism for `picture` and `Tikz` environments.

1. Create the SVG images:

Enter ⇒ `l warpmk limages`

Enter ⇒ `l warpmk 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 L<sup>A</sup>T<sub>E</sub>X source.

### ⚠ Adding/removing

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

Expressing math as SVG images has the advantage of representing the math exactly as L<sup>A</sup>T<sub>E</sub>X would, but has the disadvantage of requiring an individual file for each

### ⚠ Lots of files!

math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time `$x$` is used, for example, yet another file is created. For a document with a large amount of math, see section 5.5 to use MathJax instead.

## 5.5 Using MathJax for math

[math with MathJax](#) Math may also be represented using the MathJax Javascript project.

1. In the tutorial's source code, change

```
\usepackage{l warp}  
      to  
\usepackage[mathjax]{l warp}
```

2. Recompile

Enter ⇒ l warpmk 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 L<sup>A</sup>T<sub>E</sub>X source.

While using MathJax has many advantages, it may not be able to represent complex expressions or spacing adjustments as well as L<sup>A</sup>T<sub>E</sub>X.

## 5.6 Changing the CSS style

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

The styles provided by lwarf include:

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

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

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

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

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

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

<project>.css has an entry which loads lwarf.css, and this entry may be changed to load lwarf\_formal.css or lwarf\_sagebrush.css if desired. Additional changes to the CSS may be made by making entries later in the <project>.css file.

## 5.7 Customizing the HTML output

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

**⚠ Placement!** Note that if changes are made, it is best to first use `lwarpmk clearall` to clear all the HTML, PDF, and auxiliary files. Also, if `\HomeHTMLFilename`, `\HTMLFileName`, or `\UseLatexmk` are changed, it is necessary to reenable the `lwarpmk-newproject` package and then recompile the print version in order to recreate the configuration files for `lwarpmk`.

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

<code>\HomeHTMLFilename</code>	<code>\HomeHTMLFilename</code> : Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is: <code>\newcommand{\HomeHTMLFilename}{index}</code> causing the homepage to be the file <code>index.html</code> . Underscores are allowed in filenames.
<code>\HTMLFileName</code>	<code>\HTMLFileName</code> : A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.
<code>\UseLatexmk</code>	<code>\UseLatexmk</code> : A macro which controls whether <code>lwarp</code> uses <code>latexmk</code> to compile the document. This setting is written to <code>lwarpmk</code> 's configuration files. Defaults to false. Set to true with: <code>\newcommand*{\UseLatexmk}{true}</code>

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

<code>Ctr tocdepth</code>	<code>tocdepth</code> : Sectioning depth of the table of contents. See section 10 for a list of LATEX stack depths.
<code>Ctr SideTOCDepth</code>	<code>SideTOCDepth</code> : Sectioning depth of the sidetOC. Defaults to 1, causing the sidetoc to show sections but not subsections.
<code>sideTOC</code>	Each subpage of the website has its own small table of contents on the side (the “sidetOC”). Its depth is set by <code>SideTOCDepth</code> . This sidetOC is only shown if the web page is wide enough. When using a narrow web browser window, “responsive web design” is used to show the sidetOC at the top of the page and a link back to “Home” at the bottom.  It is recommended to set: <code>SideTOCDepth=FileDepth</code> , or <code>SideTOCDepth=FileDepth+1</code>
<b>⚠</b>	If <code>SideTOCDepth &lt; FileDepth</code> , web pages will be inaccessible via the sidetOC.
<code>Ctr FileDepth</code>	<code>FileDepth</code> : Sectioning depth of file splits. Defaults to -5, causing the entire HTML website to be one single file.

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



Bool `CombineHigherDepths`

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

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

Inaccesible pages!

Lost in an old page!

Bool `FileSectionNames`

HTML filenames

**FileSectionNames:** If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the `\HTMLFileName` is prefixed.

Example HTML filenames:

#### Numbered HTML nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`. (See `\SetHTMLFileName` to number grouped by chapter, for example.)

---

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

---

#### Named HTML sections, no prefix:

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

---

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

---

---

```
\usepackage{lwarp}
\booltrue{FileSectionNames}
```

---

**Named HTML sections, with prefix:**

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

---

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

---

`\abstractname`

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

`\MetaLanguage`

**\MetaLanguage:** The HTML language meta header. Defaults to en-US.

`\SetFirstPageTop`

**\SetFirstPageTop:** `{<contents>}` A user-definable custom action applied to the top of the home page. Useful for logos, etc. Defaults empty. Ignored in print output.

`\SetPageTop`

**\SetPageTop:** `{<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.

`\SetPageBottom`

**\SetPageBottom:** `{<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 section break:**

`\tableofcontents`

**\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 before any sectioning command which causes a file break:**

`\NewCSS`

**\NewCSS:** `{<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 `\NewCSS` command. If `\NewCSS` is not used, a default plain style is used to mimic printed L<sup>A</sup>T<sub>E</sub>X output. `lwarp_sagebrush.css` is a semi-fancy colored style as shown in this tutorial. Change it to `lwarp_formal.css` for a more formal look, or comment out the `\NewCSS` command to see the default. `\NewCSS` may be used before each file break to set the css for individual pages of the website.

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

**Placed in the document wherever necessary:**

Env \warpprint	<b>warpprint:</b> An environment which is only used while generating print output. Place here anything which does not apply to HTML and which may cause problems with l warp. If l warp 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 l warp using this environment.
Env \warpHTML	<b>warpHTML:</b> An environment which is only used while generating HTML output. This is useful for website logos and other items which have no purpose in printed output.
\warpprintonly	\warpprintonly: {<contents>} A macro version of the warpprint environment.
\warpHTMLonly	\warpHTMLonly: {<contents>} A macro version of the warpHTML environment.

## 5.8 Using `latexmk`

`latexmk` is a L<sup>A</sup>T<sub>E</sub>X utility used to monitor changes in source files and recompile as needed.

1. In the tutorial's source code, at "HTML FILENAME AND LATEXMK SETTINGS", insert the line:

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

2. Directly recompile the printed version of the document.

⚠ NOT `lwarpmk`!

Enter ⇒ `pdflatex tutorial.tex`

(Or `xelatex` or `lualatex`)

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

3. Recompile the document.

Enter ⇒ `lwarpmk print` and/or

Enter ⇒ `lwarpmk html`

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

## 5.9 Using XeLaTeX or LuaLaTeX

X<sub>E</sub>LaTeX or LuaLaTeX may be used instead of L<sup>A</sup>T<sub>E</sub>X.

1. Remove the auxiliary files for the project:

Enter ⇒ lwarfpmk cleanall

2. Be sure that

\usepackage{lwarf-newproject}  
is still enabled in `tutorial.tex`.

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

Enter ⇒ xelatex `tutorial.tex` -or-

Enter ⇒ lualatex `tutorial.tex`

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

4. To recompile the document:

Enter ⇒ lwarfpmk print -and-

Enter ⇒ lwarfpmk html

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

## 5.10 Cleaning auxiliary files

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

Enter ⇒ lwarpmk clean

## 5.11 Cleaning auxiliary and output files

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

Enter ⇒ lwarpmk cleanall

## 5.12 Processing multiple projects in the same directory

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

To create each project:

Enter ⇒ pdflatex project\_a

Enter ⇒ pdflatex project\_b

Each project is given its own configuration file:

project\_a.lwarpmkconf, project\_b.lwarpmkconf

To compile each project with lwarkmk:

Enter ⇒ lwarpmk print project\_a

Enter ⇒ lwarpmk html project\_b

## 5.13 Using the make utility

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

lwarpmk pdftohtml [project]

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

## 6 Additional details

### 6.1 Font and UTF-8 support

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

**vector fonts** 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.

 `XETATEX` and `LuaETATEX` users must use the `fontspec` package. Do NOT use `fontenc`.

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

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

- For `XETATEX` or `LuaETATEX`:

Pkg `fontspec`

- `fontspec` and font choices

`ligatures`

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

---

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

---

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File `glyptounicode`

- `\input glyptounicode.tex`

- `\input glyptounicode-cmr.tex%` from the `pdfx` package

- `\pdffgentounicode=1`

Pkg	cmap	– cmap
Pkg	textcomp	– textcomp
Pkg	microtype ligatures	– microtype is automatically used by lwarf 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.

3. \usepackage{lwarf} (section 6.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

## 6.2 lwarf package loading and options

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

Pkg lwarf Load the lwarf package immediately after the font and UTF-8 setup commands.

Opt lwarf warpprint Select the warpprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarf-newproject is loaded in print mode, it creates `<project>.html.tex`, which sets the warpHTML option before calling the user's source code `<project>.tex`. In this way, `<project>.tex` can \usepackage{lwarf} without any options to create a printed version, while `<project>.html.tex` will create an HTML version.

Opt lwarf mathsvg For math display, select mathsvg (default), or mathjax. For more information about the math options, see section 6.10.2.

## 6.3 Selecting the operating system

lwarf tries to detect which operating system is being used.

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

## 6.4 Selecting actions for print or HTML output

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

For most of built-in LATEX and many additional packages there is user-level source code support or emulation, so no special handling will be required. For those cases

---

which `lwarp` does not handle by itself, the following environments and macros may be used to isolate sections of code for print-only or HTML-only.

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

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

---

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

---

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

---

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

---

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

---

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

---

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

`\warpprintonly {<actions>}`

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

`\warpHTMLonly {<actions>}`

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

## 6.5 Commands to be placed into the `warpprint` environment

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

- Paragraph formatting: `\parindent \parskip`
- Variable spaces such as `\vspace`. `\hfill` is turned into a `\quad`. Fixed spaces such as `\quad` are emulated correctly.
- Manual page positions such as the `textpos` package, which is emulated but only in a limited way.

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

Also see section 8: [Troubleshooting](#).

## 6.6 Commands for a successful HTML conversion

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

**Page references:** The printed page does not translate to the HTML page, so references to page numbers are converted to parentheses containing `\pageref{PageFor}`, which defaults to “see”, followed by a hyperlink to the appropriate object. Ex: “Sec. 1.23 on page (see sec. 1.23)”. `\pageref{PageFor}` may be redefined to “page for”, empty, etc.

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

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

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

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

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

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

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

**\fbox around a minipage:**

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

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

- Place the `\fbox` command and its closing brace inside `\warpprint` environments.
- Use `\mdframed` instead.

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

---

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

---

Also see section 8: [Troubleshooting](#).

## 6.7 Title page

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

---

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

---

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

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

`\title {\langle title\rangle}`

⚠ Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use the `\subtitle` command instead. In HTML, the title will appear in a heading “h1”.

`\author {\langle author\rangle}`

⚠ In `\author`, use `\protect` before formatting commands such as `\textsc`. In HTML,

the author will appear in a div class `author`. `\affiliation` is a new addition to `\lwarf`.

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

`\date` works as expected. In HTML, this will appear in a div class “`titledate`”.

`\subtitle`  $\{\langle subtitle \rangle\}$

A new command which sets a subtitle. Newlines are allowed. The default is empty. In HTML, this will appear in a div class “`subtitle`”.

`\published`  $\{\langle published \rangle\}$

A new command which sets a publisher. The default is empty. In HTML, this will appear in a div class “`published`”.

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

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

## 6.8 HTML page meta descriptions

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

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

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

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

`\NewCSSdescription{}`

## 6.9 CSS

File `lwarf.css` It is best to make a local project-specific CSS file such as `project.css`, containing

File `project.css`

File `sample_project.css`

---

only things which are different from `lwarp.css`. `project.css` should refer to `lwarp.css` as follows:

---

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

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

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

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

---

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

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

## 6.10 Special cases and limitations

### 6.10.1 Text formatting

\textbf{, etc.} are supported, but \bfseries{, etc.} are not yet supported.

### 6.10.2 Math

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

- SVG math option** For SVG math, math is rendered as usual by L<sup>A</sup>T<sub>E</sub>X into the initial PDF file using the current font<sup>6</sup>, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by L<sup>A</sup>T<sub>E</sub>X with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the L<sup>A</sup>T<sub>E</sub>X code which generated the math, allowing copy/paste of the L<sup>A</sup>T<sub>E</sub>X math expression into other documents.

---

<sup>6</sup>See section 136 regarding fonts and fractions.

**SVG image font size** The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

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

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

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

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

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

Prog MathJax

When MathJax is enabled, math is rendered twice:

1. As regular L<sup>A</sup>T<sub>E</sub>X PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of L<sup>A</sup>T<sub>E</sub>X, and
2. As detokenized printed L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X values, and the MathJax cross-referencing system is ignored in favor of the L<sup>A</sup>T<sub>E</sub>X internal system, seamlessly integrating with the rest of the L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X. \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 L<sup>A</sup>T<sub>E</sub>X math expression to manually insert tags, but this has not yet been done.

[footnotes in math](#)

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

[lateximage](#)

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

[siunitx](#)

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

[https://github.com/mathjax/MathJax-third-party-extensions/  
tree/master/siunitx](https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx)

[L<sup>A</sup>T<sub>E</sub>X macros](#)

- MathJax does not automatically support custom L<sup>A</sup>T<sub>E</sub>X macros, but they may be set up by the user.

[custom MathJax macros](#) For an example of using custom L<sup>A</sup>T<sub>E</sub>X macros with MathJax, see page 213.

### 6.10.3 Graphics

[\graphicspath](#) \graphicspath only works for a single directory; all graphics must be in this directory.

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

[options](#) \includegraphics accepts width and height, origin, rotate and scale, plus a new class key.

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

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

[\rotatebox](#) \rotatebox accepts the optional origin key.

⚠ **browser support**

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

#### 6.10.4 Xcolor

support	Color definitions, models, and mixing are fully supported without any changes required.
tables	Colored tables are ignored so far. Use css to style tables.
colored text and boxes	\textcolor, \colorbox, and \fcolorbox are supported.
\color and \pagecolor	\color and \pagecolor are ignored. Use css or \textcolor where possible.

#### 6.10.5 Tikz

⚠ **preamble**

For recent versions of Tikz (v3.0.0 — 2013/12/20 and later), include in the document preamble:

```
\usetikzlibrary{babel} % v3.0.0 and up
```

For older versions of Tikz without the Tikz babel library, lwarf automatically handles the catcode change for the \$ math shift.

See section 61.

#### 6.10.6 Tabular

column types

- Vertical rules are not yet supported.
- \* in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of @, !, >, and < may be used at each column, and they are used in that order.
- \newcolumntype is ignored; unknown column types are set to 1.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.
- Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)

⚠ **\multicolumn &**

\multicolumn

⚠ `\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 with rules`

`rule at last row`

- If a multirow reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.

⚠ `paragraphs`

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

`\cmidrule width, trim`

`longtable headings`








#### 6.10.7 Minipages

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

⚠ `inline`

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

`in a span`

There is limited support for minipages inside an HTML `<span>`. An HTML `<div>`

cannot appear inside a `<span>`. While in a `<span>`, minipages and parboxes are ignored. Use `\newline` or `\par` for an HTML break.

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

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

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

### 6.10.8 Mdframed

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

**⚠ loading** When used, l warp loads `mdframed` in HTML with `framemethod=none`.

For title font, use

`frametitlefont=\textbf,`

**font** instead of

`frametitlefont=\bfseries,`

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

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

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

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

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

**⚠ package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 131.1.

### 6.10.10 **caption** and **subcaption** packages

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

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

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

### 6.10.11 **floatrow** package

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

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

### 6.10.12 **abstract** package

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

### 6.10.13 **cleveref** and **varioref** packages

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

### 6.10.14 **verse** and **memoir**

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

output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

---

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

---

Pkg `lwarp`  
 △ `\leftskip`  
`\leftmargini`  
`\TMLleftskip`  
`\TMLleftmargini`

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

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

### 6.10.15 `siunitx` package

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

### 6.10.16 `newclude` package

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

Ex:

---

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

---

### 6.10.17 `newtxmath` package

Pkg `newtxmath` The proper load order is:

△ `loading sequence`

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

#### 6.10.18 **babel** package

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

 **French** to preserve the special HTML and enumitem list handling.

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

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

#### 6.10.19 **enumitem** package

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

#### 6.10.20 **enumerate** package

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

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

## 7 Expanding the code base

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

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

An environment may be converted to a `\teximage` then displayed with an image of the resulting L<sup>A</sup>T<sub>E</sub>X output. See section 60 for an example of the `picture` environment.

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

## 7.1 Creating an l warp version of a package

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

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

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

Each `l warp-` package should first call either `\LWR@ProvidesPackageDrop` or `\LWR@ProvidesPackagePass`. If `Dropped`, the original print-version package is ignored, and only the `l warp-` version is used. Use this where the original print version is useless for HTML. If `Passed`, the original package is loaded first, with the supplied options, then the `l warp-` version continues loading as well. 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 `l warp-` version.

## 8 Troubleshooting

### 8.1 Using the `lwarf.sty` package

Also see:

[Section 6.5: Commands to be placed into the `warpprint` environment](#)

[Section 6.6: Commands for a successful HTML conversion](#)

[Section 6.10: Special cases and limitations](#)

**Text is not converting:**

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

**Undefined HTML settings:**

See the warning regarding the placement of the HTML settings at section [5.7](#).

**Obscure error messages:**

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

**Missing sections:**

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

**Missing HTML files:**

See the warning regarding changes to the HTML settings at section [5.7](#).

**Missing / incorrect cross-references:**

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

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

Use X<sup>L</sup>A<sub>T</sub>E<sub>X</sub> or Lua<sup>L</sup>A<sub>T</sub>E<sub>X</sub>.

**Floats out of sequence:**

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

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

**Print document contains HTML tags:**

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

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

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

**Images are appearing in strange places:**

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

**“Leaders not followed by proper glue”:** This can be caused by a missing `1@<floattype>` or `1@<sectiontype>` definition. See `lwarp`’s definitions for examples.

**Plain-looking document:**

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

**Broken fragments of HTML:**

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

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

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

**Un-matched conditional compiles:**

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

### 8.1.1 Debug tracing output

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

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

## 8.2 Compiling the lwarf.dtx file

Illogical error messages caused by an out-of-sync lwarf.sty file:

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

Un-nested environments:

Be sure to properly nest:

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

## 9 Implementation

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

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

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

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

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

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

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

Table 5: Section depths and HTML headings

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

## 10 Stack depths

Stacks are created to track depth inside the LATEX document structure. This depth is translated to HTML headings as shown in table 5. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the 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 will push the HTML headings down a step, and also cause `subsubsection` to become a `div` due to a limit of six HTML headings.

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

## 11 Source Code

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

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

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

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

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

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

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

**for HTML output:** 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:**

`lwarf` source code begins on the following page.

## 12 Detecting the T<sub>E</sub>X Engine — pdflatex, lualatex, xelatex

```

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

```

## 13 Unicode Input Characters

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

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

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

```

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

```

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

```

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

```

## 14 Handling package options

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

```

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

```

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

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

Pkg **comment** Provides conditional code blocks.

```
25 \RequirePackage{comment}%
26 \excludecomment{testing}
```

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

```
27 \RequirePackage{kvoptions}%
28 \SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool **warpingprint** Set to true/false depending on the package option selections for print/HTML output

Bool **warpingHTML**:

Bool **mathjax**

```
29 \newbool{warpingprint}%
30 \newbool{warpingHTML}%
31 \newbool{mathjax}
```

`\warpprintonly {<contents>}`

Only process the contents if producing printed output.

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

`\warpHTMLonly {<contents>}`

Only process the contents if producing HTML output.

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

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

```
34 \includecomment{warpall}
```

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

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

```
35 \DeclareVoidOption{warpprint}{%
36 \PackageInfo{lwarp}{Using option 'warpprint'}%
37 \includecomment{warpprint}%
38 \excludecomment{warpHTML}%
}
```

---

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

**Opt l warp** `BaseJobname` Option `BaseJobname` sets the `\BaseJobname` for this document.

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

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

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

```

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

```

```

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

```

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

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

```
\@namedef {⟨name⟩} {⟨defn⟩}
```

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

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

## 15 Misplaced packages

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

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

```
l warp-⟨packagename⟩.sty
```

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

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

```

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

```

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

```

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

```

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

Packages which should only be loaded after `lwarf` are tested here to trip an error if they have already been loaded.

The following packages must be loaded after `lwarf`:

```
86 \LWR@loadafter{abstract}
87 \LWR@loadafter{afterpage}
88 \LWR@loadafter{algorithmicx}
89 \LWR@loadafter{alltt}
90 \LWR@loadafter{bookmark}
91 \LWR@loadafter{booktabs}
92 \LWR@loadafter{changepage}
93 \LWR@loadafter{dcolumn}
94 \LWR@loadafter{draftwatermark}
95 \LWR@loadafter{enumerate}
96 \LWR@loadafter{epigraph}
97 \LWR@loadafter{eso-pic}
98 \LWR@loadafter{everypage}
99 \LWR@loadafter{extramarks}
100 \LWR@loadafter{fancyhdr}
101 \LWR@loadafter{floatrow}
102 \LWR@loadafter{float}
103 \LWR@loadafter{ftnright}
104 \LWR@loadafter{geometry}
105 \LWR@loadafter{graphics}
106 \LWR@loadafter{graphicx}
107 \LWR@loadafter{hyperref}
108 \LWR@loadafter{indentfirst}
109 \LWR@loadafter{keyfloat}
110 \LWR@loadafter{layout}
111 \LWR@loadafter{letterspace}
112 \LWR@loadafter{listings}
113 \LWR@loadafter{longtable}
114 \LWR@loadafter{lscape}
115 \LWR@loadafter{ltcaption}
116 \LWR@loadafter{mcaption}
117 \LWR@loadafter{mdframed}
118 \LWR@loadafter{microtype}
119 %\LWR@loadafter{multicol} loaded by ltxdoc
120 \LWR@loadafter{multirow}
121 \LWR@loadafter{nameref}
122 \LWR@loadafter{needspace}
123 \LWR@loadafter{newtxmath}
124 \LWR@loadafter{nextpage}
125 \LWR@loadafter{nowidow}
```

---

```

126 \LWR@loadaft{pagenote}
127 \LWR@loadaft{parskip}
128 \LWR@loadaft{placeins}
129 \LWR@loadaft{ragged2e}
130 \LWR@loadaft{rotating}
131 \LWR@loadaft{setspace}
132 \LWR@loadaft{showkeys}
133 \LWR@loadaft{sidecap}
134 \LWR@loadaft{tabularx}
135 \LWR@loadaft{textpos}
136 \LWR@loadaft{threeparttable}
137 \LWR@loadaft{titleps}
138 \LWR@loadaft{titlesec}
139 \LWR@loadaft{titletoc}
140 \LWR@loadaft{tocloft}
141 \LWR@loadaft{trivfloat}
142 \LWR@loadaft{varioref}
143 \LWR@loadaft{verse}
144 \LWR@loadaft{wallpaper}
145 \LWR@loadaft{wrapfig}
146 \LWR@loadaft{xcolor}
147 \LWR@loadaft{xfrac}

```

## 16 Required packages

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

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

**for HTML & PRINT:** 148 `\begin{warpall}`

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

Detects X<sub>E</sub>T<sub>E</sub>X and LuaE<sub>T</sub>E<sub>X</sub>:

```

149 \RequirePackage{iftex}
150 \newif\ifxetexorluatex
151 \ifXeTeX
152     \xetexorluatextrue
153 \else
154     \ifLuaTeX
155         \xetexorluatextrue

```

```

156     \else
157         \xetexorluatexfalse
158     \fi
159 \fi

160 \end{warpall}

```

**for HTML output:** 161 \begin{warpHTML}

```

162 \ifxetexorluatex
163 % ^~A \usepackage[no-math]{fontspec}

```

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

```

164 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
165 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
166 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
167 \else

```

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

Pkg `microtype`

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

```

168 \RequirePackage {microtype}
169
170 \microtypesetup{
171   protrusion=false,
172   expansion=false,
173   tracking=false,
174   kerning=false,
175   spacing=false}
176
177 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
178 \fi
179 \end{warpHTML}

```

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

- Uses a very long and wide page to minimize page breaks and margin overflow.
- Will also use a scriptsize font.

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

**for HTML output:** 180 \begin{warpHTML}  
 181 \RequirePackage[paperheight=190in,paperwidth=20in,%  
 182 left=7in,right=7in,%  
 183 top=1in,bottom=1in,%  
 184 ]{geometry}  
 185 \end{warpHTML}

**for HTML & PRINT:** 186 \begin{warpall}

Pkg **xparse**  
 L<sup>A</sup>T<sub>E</sub>X3 command argument parsing  
 187 \RequirePackage{xparse}  
 188 \end{warpall}

**for HTML output:** 189 \begin{warpHTML}

Pkg **expl3**  
 L<sup>A</sup>T<sub>E</sub>X3 programming  
 190 \RequirePackage{expl3}

Pkg **getttitlestring**

Used to emulate \nameref.

191 \RequirePackage{getttitlestring}

Pkg **everyhook**

everyhook is used to patch paragraph handling.

192 \RequirePackage{everyhook}  
 193 \end{warpHTML}

**for HTML & PRINT:** 194 \begin{warpall}

Pkg **fancyvrb**  
 Used for Verbatim, verse.  
 195 \RequirePackage{fancyvrb}

```
196 \end{warpall}

for HTML output: 197 \begin{warpHTML}

Pkg  xifthen
      198 \RequirePackage{xifthen}

Pkg  xstring
      199 \RequirePackage{xstring}

Pkg  makeidx
      200 \RequirePackage{makeidx}
      201 \makeindex

Pkg  calc
      202 \RequirePackage{calc}

Pkg  newfloat
      203 \RequirePackage{newfloat}

Pkg  caption
      204 \RequirePackage{caption}

Pkg  subcaption
      205 \RequirePackage{subcaption}

Pkg  enumitem
      enumitem is patched to support \newlist with HTML.
      206 \RequirePackage{enumitem}
      207 \setlist[itemize]{leftmargin=0em}
      208 \setlist[enumerate]{leftmargin=0em}
      209 \setlist[description]{leftmargin=0em}

Pkg  pagenote
      Used to convert footnotes into pagenotes.

      210 \RequirePackage[continuous]{pagenote}
      211 \makepagenote
```

```
212 \let\LWR@origmakepagenote\makepagenote  
213 \let\LWR@origprintnotes\printnotes  
214 \let\LWR@origpagenote\pagenote  
215 \end{warpHTML}
```

**for HTML & PRINT:** 216 \begin{warpall}

Pkg **titling**

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

```
217 \RequirePackage{titling}  
  
218 \end{warpall}
```

**for HTML output:** 219 \begin{warpHTML}

Pkg **zref**

Used for cross-references.

```
220 \RequirePackage{zref}
```

Pkg **amsmath**

Equation numbers are placed to the left for HTML.

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

```
221 \PassOptionsToPackage{leqno,fleqn}{amsmath}  
222 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
223 \RequirePackage{environ}
```

Pkg **titleps**

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

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

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

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

```
\pagestyle {⟨style⟩}

225 \let\LWR@origpagestyle\pagestyle
226 \renewcommand*\pagestyle[1]{}

\thispagestyle {⟨style⟩}

227 \let\LWR@origthispagestyle\thispagestyle
228 \renewcommand*\thispagestyle[1]{}

\pagenumbering {⟨commands⟩}

229 \let\LWR@origpagenumbering\pagenumbering
230 \renewcommand*\pagenumbering[1]{}
```

Pkg xfrac

Patched for HTML use. See section 136.

```
231 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
232 \RequirePackage{printlen}
```

```
233 \end{warpHTML}
```

## 17 Loading Packages

for HTML output: 234 \begin{warpHTML}

Remember the original \RequirePackage:

```
235 \let\LWR@origRequirePackage\RequirePackage
```

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

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

\LWR@findword [⟨1: separator⟩] {⟨2: list⟩} {⟨3: index⟩} [⟨4: destination⟩]

Note that argument 4 is passed directly to \StrBetween.

---

```
237 \newcommand*\LWR@findword[3][,]{%
238     \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
239 }
```

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

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

Find the n'th package name from the list:

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

See if the package name was found:

```
242 \IfStrEq{\LWR@strresult}{}
243 {}% no filename
244 {}% yes filename
```

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

```
245 \IfExists{l warp-\LWR@strresult.sty}
246 {}% latex_html_ file found
247 \StrSubstitute
248 {\LWR@requirepackagenames}
249 {\LWR@strresult}
250 {l warp-\LWR@strresult}[\LWR@requirepackagenames]
251 }
252 {}% no latex_html_* file
253 {}% yes filename
254 }
```

`\RequirePackage [(<1: options>) ] {(<2: package names>) } [(<3: version>) ]`

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

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

Redirect up to nine names:

```
256 \renewcommand*{\LWR@requirepackagenames}{#2}
257 \LWR@lookforpackagename{1}
258 \LWR@lookforpackagename{2}
259 \LWR@lookforpackagename{3}
260 \LWR@lookforpackagename{4}
261 \LWR@lookforpackagename{5}
```

```

262 \LWR@lookforpackagename{6}
263 \LWR@lookforpackagename{7}
264 \LWR@lookforpackagename{8}
265 \LWR@lookforpackagename{9}

```

\RequirePackage depending on the options and version:

```

266 \IfValueTF{#1}
267 {%
268   \IfValueTF{#3}{%
269     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}{#3}}
270     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
271   }%
272   {%
273     \IfValueTF{#3}{%
274       {\LWR@origRequirePackage{\LWR@requirepackagenames}{#3}}
275       {\LWR@origRequirePackage{\LWR@requirepackagenames}}
276     }%
277   }%
278   \let\usepackage\RequirePackage

```

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

Provides a new package which passes its options to the original.

```

279 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{%
280   \PackageInfo{lwarp}{Using the lwarp version of package '#1, including options,}%
281   \IfValueTF{#2}{%
282     {\ProvidesPackage{lwarp-#1}{#2}}%
283     {\ProvidesPackage{lwarp-#1}}%
284     \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}%
285     \ProcessOptions\relax
286   }%
287   \IfValueTF{#2}{%
288     {\LWR@origRequirePackage[#1]{#2}}%
289     {\LWR@origRequirePackage[#1]}%
290   }%

```

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

Provides a new package which drops/discards its options.

```

291 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{%
292   \PackageInfo{lwarp}{Using the lwarp version of package '#1, discarding options,}%
293   \IfValueTF{#2}{%
294     {\ProvidesPackage{lwarp-#1}{#2}}%
295     {\ProvidesPackage{lwarp-#1}}%
296     \DeclareOption*{}%
297     \ProcessOptions\relax

```

---

```

298 }

299 \end{warpHTML}

```

## 18 Copying a file

**for HTML output:** 300 \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.

```

301 \newcommand*\{\LWR@copyfile\}[2]{%
302 \newwrite\copyfile % open the file to write to
303 \immediate\openout\copyfile=\#2
304 \newread\file % open the file to read from
305 \openin\file=\#1
306 \begingroup\endlinechar=-1
307 \makeatletter
308 \loop\unless\ifeof\file
309   \read\file to\fileline % Read one line and store it into \fileline
310 %   \fileline\par % print the content into the pdf
311 % print the content:
312   \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
313 \repeat
314 \closeout\copyfile
315 \endgroup
316 }

317 \end{warpHTML}

```

## 19 Debugging messages

Bool LWR@tracinglwarp True if tracing is turned on.

318 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.

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

---

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

```

320 \newcommand{\LWR@traceinfo}[1]{%
321 \ifbool{LWR@tracing}{\warp{}}{%
322 {\PackageInfo{lwarf}{#1 : }}}%
323 {}%
324 }
```

## 20 Remembering original formatting macros

**for HTML output:** 325 \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.

```

326 \let\LWR@origtextit{textit}
327 \let\LWR@origtextbf{textbf}
328 \let\LWR@origtexttt{texttt}
329 \let\LWR@origtextsc{textsc}
330 \let\LWR@origtextsf{textsf}
331 \let\LWR@origtextrm{textrm}
332 \let\LWR@origbfseries{bfseries}
333 \let\LWR@origrmfamily{rmfamily}
334 \let\LWR@origttfamily{ttfamily}
335 \let\LWR@orignormalfont{normalfont}
336
337 \let\LWR@origraggedright{raggedright}
338 \let\LWR@origonecolumn{onecolumn}
339
340 \let\LWR@origtextsuperscript{textsuperscript}
341 \let\LWR@origtextsubscript{textsubscript}
342
343 \let\LWR@origscriptsize{scriptsize}
344
345 \let\LWR@orignewpage{newpage}
346
347 \let\LWR@origminipage{minipage}
348 \let\LWR@origendminipage{endminipage}
349
350 \let\LWR@orignewline{newline}
351
352 \let\LWR@origitem{item}
353
354 \let\LWR@origpar{par}
355
```

---

```

356
357 \let\LWR@origfootnote\footnote
358 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
359
360 \let\LWR@origclearpage\clearpage
361 \let\clearpage\relax
362 \let\cleardoublepage\relax

363 \end{warpHTML}

```

## 21 Operating-System portability

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

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

**for HTML & PRINT:** 364 `\begin{warpall}`

### 21.1 Common portability code

`Bool usingOSwindows` Set if `\warpOSwindows`.

```

365 \newbool{usingOSwindows}
366 \boolfalse{usingOSwindows}

```

### 21.2 Unix, Linux, and Mac OS

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

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

### 21.3 MS-Windows

For MS-Windows:

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

---

```

368 \newcommand*{\warpOSwindows}
369 {
370 \booltrue{usingOSwindows}
371 \renewcommand*{\OSPathSymbol}{\@backslashchar}
372 }
```

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

```

373 \ifwindows
374 \warpOSwindows
375 \fi
376 \end{warpall}
```

## 22 Stacks

**for HTML output:** 377 `\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.

### 22.1 Assigning depths

initial depths for empty stack entries:

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

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

```

379 \newcommand*{\LWR@depthfinished}{-4}
380 \newcommand*{\LWR@depthpart}{-1}
381 \newcommand*{\LWR@depthchapter}{0}
382 \newcommand*{\LWR@depthsection}{1}
```

---

```
383 \newcommand*{\LWR@depthsubsection}{2}
384 \newcommand*{\LWR@depthsubsubsection}{3}
385 \newcommand*{\LWR@depthparagraph}{4}
386 \newcommand*{\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

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

## 22.2 Closing actions

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

```
389 \newcommand*{\LWR@closeone}{}% top of the stack
390 \newcommand*{\LWR@closetwo}{}%
391 \newcommand*{\LWR@closethree}{}%
392 \newcommand*{\LWR@closefour}{}%
393 \newcommand*{\LWR@closefive}{}%
394 \newcommand*{\LWR@closesix}{}%
395 \newcommand*{\LWR@closeseven}{}%
396 \newcommand*{\LWR@closeeight}{}%
397 \newcommand*{\LWR@closenine}{}%
398 \newcommand*{\LWR@closeten}{}%
399 \newcommand*{\LWR@closeeleven}{}%
400 \newcommand*{\LWR@closetwelve}{}%
```

## 22.3 Closing depths

A stack to record the depth of each level:

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

Ex:

---

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

---

```
401 \newcommand*{\LWR@closedepthonen}{\LWR@depthnone}%
402 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
403 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
404 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
405 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
406 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
407 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
408 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
409 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
410 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
411 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
412 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

## 22.4 Pushing and popping the stack

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

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

```
413 \NewDocumentCommand{\pushclose}{m m}
414 {
415 \let\LWR@closedephtwelve\LWR@closeeleven
416 \let\LWR@closeeleven\LWR@closeten
417 \let\LWR@closeten\LWR@closenine
418 \let\LWR@closenine\LWR@closeeight
419 \let\LWR@closeeight\LWR@closeseven
420 \let\LWR@closeseven\LWR@closesix
421 \let\LWR@closesix\LWR@closefive
422 \let\LWR@closefive\LWR@closefour
423 \let\LWR@closefour\LWR@closethree
424 \let\LWR@closethree\LWR@closetwo
425 \let\LWR@closetwo\LWR@closeone
426 \let\LWR@closeone#1
427 \let\LWR@closedephtwelve\LWR@closedeptheleven
428 \let\LWR@closedeptheleven\LWR@closedephten
429 \let\LWR@closedephten\LWR@closedepthnine
```

---

```

430 \let\LWR@closedepthnine\LWR@closedeptheight
431 \let\LWR@closedeptheight\LWR@closedeptseven
432 \let\LWR@closedeptseven\LWR@closedeptsix
433 \let\LWR@closedeptsix\LWR@closedeptfive
434 \let\LWR@closedeptfive\LWR@closedeptfour
435 \let\LWR@closedeptfour\LWR@closedeptthree
436 \let\LWR@closedeptthree\LWR@closedepttwo
437 \let\LWR@closedepttwo\LWR@closedeptone
438 \let\LWR@closedeptone#2
439 }

```

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

```

440 \newcommand*{\popclose}{}
441 {
442 \let\LWR@closeone\LWR@closetwo
443 \let\LWR@closetwo\LWR@closethree
444 \let\LWR@closethree\LWR@closefour
445 \let\LWR@closefour\LWR@closefive
446 \let\LWR@closefive\LWR@closesix
447 \let\LWR@closesix\LWR@closeseven
448 \let\LWR@closeseven\LWR@closeeight
449 \let\LWR@closeeight\LWR@closenine
450 \let\LWR@closenine\LWR@closeten
451 \let\LWR@closeten\LWR@closeeleven
452 \let\LWR@closeeleven\LWR@closetwelve
453 \let\LWR@closedeptone\LWR@closedepttwo
454 \let\LWR@closedepttwo\LWR@closedeptthree
455 \let\LWR@closedeptthree\LWR@closedeptfour
456 \let\LWR@closedeptfour\LWR@closedeptfive
457 \let\LWR@closedeptfive\LWR@closedeptsix
458 \let\LWR@closedeptsix\LWR@closedeptseven
459 \let\LWR@closedeptseven\LWR@closedepteight
460 \let\LWR@closedepteight\LWR@closedeptnine
461 \let\LWR@closedeptnine\LWR@closedeptten
462 \let\LWR@closedeptten\LWR@closedepteleven
463 \let\LWR@closedepteleven\LWR@closedepttwelve
464 }

465 \end{warpHTML}

```

## 23 Data arrays

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

---

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

To assign an empty value:

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

**for HTML output:** 466 `\begin{warpHTML}`

```
\LWR@setexpparray {name} {index} {contents}
467 \NewDocumentCommand{\LWR@setexpparray}{m m m}%
468 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
469 }

\LWR@getexpparray {name} {index}
470 \newcommand*{\LWR@getexpparray}[2]{\csuse{#1#2}}
471 \end{warpHTML}
```

## 24 HTML entities

**for HTML output:** 472 `\begin{warpHTML}`

HTML entites and HTML Unicode entities:

```
473 \let\LWR@origampersand\&
\HTMLentity {entitytag}
474 \newcommand*{\HTMLentity}[1]{\LWR@origampersand#1;}

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

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

\textless
\textgreater
```

---

```

477 \let\LWR@origtextless\textless
478 \renewcommand*\{\textless}{\HTMLentity{lt}}
479
480 \let\LWR@origtextgreater\textgreater
481 \renewcommand*\{\textgreater}{\HTMLentity{gt}}
482 \end{warpHTML}

```

## 25 HTML filename generation

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

**for HTML & PRINT:** 483 `\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.

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

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

```
485 \providecommand*\{\HTMLFileName}{}
```

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

```
486 \providecommand*\{\HomeHTMLFileName}{\BaseJobname}
```

`\SetHTMLFileName` `\{\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.

```

487 \newcommand*\{\SetHTMLFileName}{[1]{%
488 \setcounter{LWR@htmlfilename}{#1}%
489 }}
```

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

Defaults to use section names in the filenames.

```
490 \newbool{FileSectionNames}  
491 \booltrue{FileSectionNames}
```

```
492 \end{warpall}
```

**for HTML output:** 493 \begin{warpHTML}

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

```
494 \newcounter{LWR@htmlfilename}  
495 \setcounter{LWR@htmlfilename}{0}
```

\LWR@htmlsectionfilename {<htmlfilename or name>}

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

```
496 \newcommand*{\LWR@htmlsectionfilename}[1]{%  
497 \LWR@traceinfo{\LWR@htmlsectionfilename A}%
```

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

```
498 \LWR@traceinfo{about to assign temp}%
499 \edef\LWR@tempone{\#1}%
500 \LWR@traceinfo{about to compare with ??}%
501 \ifthenelse{\equal{\LWR@tempone}{??}}{%
502 }{%
503 \LWR@traceinfo{found ??}%
504 }{%
505 \LWR@traceinfo{not found ??}%
506 }{%
507 \LWR@traceinfo{about to compare with zero or empty}%
508 \ifthenelse{%
509 \equal{\LWR@tempone}{0}}{%
510 \OR \equal{\LWR@tempone}{}}{%
511 \OR \equal{\LWR@tempone}{??}}{%
512 }{%
513 }{%
514 \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
515 \HomeHTMLFileName.html}%
516 }%
```

For a L<sup>A</sup>T<sub>E</sub>X section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```
517 {%
518 \LWR@traceinfo{\LWR@htmlsectionfilename C}%
519 \ifthenelse{%
520 \equal{\HTMLFileName}{} \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
521 }%
522 {\_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```
523 {\HTMLFileName#1.html}%
524 }%
525 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
526 }
```

`\LWR@htmlrefsectionfilename {⟨label⟩}`

Prints the filename for the given label

```
527 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
528 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
529 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
530 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
531 }

532 \end{warpHTML}
```

## 26 Homepage link

**for HTML output:** 533 `\begin{warpHTML}`

`\LinkHome` `\LinkHome` may be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

```
534 \newcommand*{\LinkHome}{%
535 \LWR@subhyperrefclass{%
536 \HomeHTMLFileName.html}%
537 {Home}{linkhome}%
538 }
```

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

---

```
539 \newcommand*\{LWR@topnavigation}{  
540 \LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}  
541 }
```

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

```
542 \newcommand*\{LWR@botnavigation}{  
543 \LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}  
544 }  
  
545 \end{warpHTML}
```

## 27 \PrintStack diagnostic tool

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

**for HTML output:** 546 \begin{warpHTML}

`\PrintStack` Prints the closedepth stack.

```
547 \newcommand*\{PrintStack}{  
548 \LWR@startpars  
549 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}  
550 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}  
551 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}  
552 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}  
553 }  
  
554 \end{warpHTML}
```

## 28 Closing stack levels

**for HTML output:** 555 \begin{warpHTML}

Close one nested level:

```
556 \newcommand*\{LWR@closeoneprevious}{%  
557  
558 \LWR@closeone{}}
```

---

```
559
560 \popclose{}
561 }
```

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

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

Close any pending paragraph:

```
563 \LWR@stopars
```

Close anything nested deeper than the desired depth:

```
564 \whiledo{\not\{\LWR@closedepthone<\#1\}}{\LWR@closeoneprevious}
565 }
```

```
566 \end{warpHTML}
```

## 29 HTML tags, spans, divs, elements

**for HTML output:** 567 \begin{warpHTML}

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

```
568 \newcommand*\{\LWR@tagpart\}{h2}
569 \newcommand*\{\LWR@tagpartend\}{/h2}
570 \newcommand*\{\LWR@tagchapter\}{h3}
571 \newcommand*\{\LWR@tagchapterend\}{/h3}
572 \newcommand*\{\LWR@tagsection\}{h4}
573 \newcommand*\{\LWR@tagsectionend\}{/h4}
574 \newcommand*\{\LWR@tagsubsection\}{h5}
575 \newcommand*\{\LWR@tagsubsectionend\}{/h5}
576 \newcommand*\{\LWR@tagsubsubsection\}{h6}
577 \newcommand*\{\LWR@tagsubsubsectionend\}{/h6}
578 \newcommand*\{\LWR@tagparagraph\}{span class="paragraph"{}}
579 \newcommand*\{\LWR@tagparagraphend\}{/span}
580 \newcommand*\{\LWR@tag subparagraph\}{span class="subparagraph"{}}
581 \newcommand*\{\LWR@tag subparagraphend\}{/span}
582
583 \newcommand*\{\LWR@tagregularparagraph\}{p}
```

## 29.2 HTML tags

\LWR@htmltagc {*tag*} Break ligatures in HTML tags.

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

```
584 \newcommand*\LWR@htmltagc[1]{%
585 {%
586 \LWR@origttfamily%
587 % \protect\LWR@origtexttt{%
588 % \protect\LWR@origtextbf{%
589 \LWR@origtextless#1\LWR@origtextgreater{%
590 }%
591 }%
592 }%
593 }
```

Env LWR@nestspan Disable minipage, \parbox inside a span.

⚠ \begin{LWR@nestspan} must follow the opening <span> tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

⚠ \end{LWR@nestspan} must follow the /span or an extra <p> may appear.

```
594 \newenvironment*\LWR@nestspan{%
595 {%
596 \addtocounter{LWR@spandepth}{1}%
597 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{\{}{\}}%
598 }%
599 {\addtocounter{LWR@spandepth}{-1}}}
```

\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.

```
600 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
601 \LWR@ensuredoingapar%
602 \LWR@htmltagc{#1}%
603 \begin{LWR@nestspan}%
604 #2%
605 \LWR@htmltagc{/#1}%
606 \end{LWR@nestspan}%
607 }
```

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

```

608 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
609 \LWR@ensuredoingapar%
610 \LWR@subhtmlelementclass{span}{#1}[#2]%
611 \begin{LWR@nestspan}%
612 #3%
613 \LWR@htmtagc{/span}%
614 \end{LWR@nestspan}%
615 }

\LWR@htmltag {⟨tag⟩}
print an HTML tag: <tag>

616 \newcommand*{\LWR@htmltagb}[1]{%
617 \LWR@htmtagc{#1}%
618 \endgroup%
619 }
620
621 \newcommand*{\LWR@htmltag}{%
622 \begingroup\catcode`\_=12
623 \LWR@htmltagb%
624 }

```

### 29.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
625 \newcommand*{\LWR@htmlopencomment}{%
626 % \LWR@origtexttt{\LWR@origtextbf{\LWR@origtextless{}!{-}{-}}}\%
627 \begingroup\LWR@origttfamily\LWR@origtextless{}!{-}{-}\endgroup%
628 }
629
630 \newcommand*{\LWR@htmlclosecomment}{%
631 % \LWR@origtexttt{\LWR@origtextbf{{-}{-}\LWR@origtextgreater{}}}\%
632 \begingroup\LWR@origttfamily{-}{-}\LWR@origtextgreater{}\endgroup%
633 }

\LWR@htmlcomment {⟨comment⟩}
634 \newcommand{\LWR@htmlcomment}[1]{%
635 \LWR@htmlopencomment{}%
636 \LWR@origextrm{#1}%
637 \LWR@htmlclosecomment{}}

```

---

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

638 \newcommand{\LWR@htmlblockcommentb}[1]
639 {\LWR@stoppars\LWR@htmlcomment{\#1}\LWR@startpars\endgroup}
640
641 \newcommand{\LWR@htmlblockcomment}
642 {%
643 \begingroup\catcode`\_=12%
644 \LWR@htmlblockcommentb%
645 }

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

646 \newcommand*\LWR@htmlblocktag[1]{%
647 \LWR@stoppars%
648 \LWR@htmlltag{\#1}%
649 \LWR@startpars%
650 }

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

651 \NewDocumentCommand{\LWR@htmlblocktagcomment}{m +m}{%
652 \LWR@stoppars%
653 \LWR@htmlltag{\#1}\LWR@htmlcomment{\#2}\LWR@orignewline%
654 \LWR@startpars%
655 }
```

## 29.4 Div class and element class

\LWR@subhtmlelementclass {\langle element\rangle} {\langle class\rangle} [{\langle style\rangle}]

Factored and reused in several places.

```
656 \NewDocumentCommand{\LWR@subhtmlelementclass}{m m o}{%
657 \IfValueTF{\#3}{%
658 {%
659 \ifthenelse{\equal{\#3}{}}{%
660 {\LWR@htmlltag{\#1 class="#2"}% empty option
661 {\LWR@htmlltag{\#1 class="#2" style="#3"}% non-empty option
662 }% option
663 {\LWR@htmlltag{\#1 class="#2"}% no option
664 }}
```

\LWR@htmlelementclass {\langle element\rangle} {\langle class\rangle} [{\langle style\rangle}]

```

665 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
666 \LWR@stoppars%
667 \LWR@subhtmlelementclass{#1}{#2}{#3}%
668 \LWR@startpars%
669 }

\LWR@htmlelementclassend {⟨element⟩} {⟨class⟩}

670 \newcommand*{\LWR@htmlelementclassend}[2]{%
671 \LWR@stoppars%
672 \LWR@htmlltag{/#1}%
673 \LWR@htmlcomment{End of #1 ``#2''}%
674 \LWR@startpars%
675 }

\LWR@htmldivclass {⟨class⟩} [⟨style⟩]

676 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
677 \LWR@htmlelementclass{div}{#1}{#2}%
678 }

\LWR@htmldivclassend {⟨class⟩}

679 \newcommand*{\LWR@htmldivclassend}[1]{%
680 \LWR@htmlelementclassend{div}{#1}%
681 }

```

## 29.5 Single-line elements

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

```

\LWR@htmlelementclassline {⟨element⟩} {⟨class⟩} [⟨style⟩] {⟨text⟩}

682 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
683 \LWR@stoppars%
684 \LWR@subhtmlelementclass{#1}{#2}{#3}%
685 #4%
686 \LWR@htmlltag{/#1}
687 \LWR@startpars%
688 }

```

## 29.6 HTML5 semantic elements

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

689 \newcommand*{\LWR@htmlelement}[1]{%
690 \LWR@htmlblocktag{#1}%
691 }

\LWR@htmlelementend  {\langle element\rangle}

692 \newcommand*{\LWR@htmlelementend}[1]{%
693 \LWR@stoppars%
694 \LWR@htmltag{/#1}%
695 \LWR@startpars%
696 }%
697
698 \end{warpHTML}
```

## 29.7 High-level block and inline classes

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

For other direct-formatting commands, see section 63.

**Env BlockClass** {\⟨class⟩} [⟨style⟩]    High-level interface for div classes.

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

**for HTML output:** 699 \begin{warpHTML}
700 \NewDocumentEnvironment{BlockClass}{m o}
701 {
702 \LWR@htmldivclass{#1}[#2]
703 }
704 {
705 \LWR@htmldivclassend{#1}
706 }
707 \end{warpHTML}

**for PRINT output:** 708 \begin{warpprint}
709 \NewDocumentEnvironment{BlockClass}{m o}{\relax}{\relax}
710 \end{warpprint}

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

**for HTML output:**

```

711 \begin{warpHTML}
712 \newcommand{\BlockClassSingle}[2]{%
713 \LWR@htmlelementclassline{div}{#1}{#2}%
714 }
715 \end{warpHTML}

for PRINT output: 716 \begin{warpprint}
717 \newcommand{\BlockClassSingle}[2]{#2}
718 \end{warpprint}

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

for HTML output: 719 \begin{warpHTML}
720 \NewDocumentCommand{\InlineClass}{m o m}{%
721 \LWR@htmlspanclass{#1}{#2}{#3}%
722 }
723 \end{warpHTML}

for PRINT output: 724 \begin{warpprint}
725 \NewDocumentCommand{\InlineClass}{m o m}{#3}
726 \end{warpprint}

```

## 29.8 Closing HTML tags

**for HTML output:** 727 \begin{warpHTML}

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

```

728 \newcommand*{\LWR@printclosepart}%
729   {\LWR@htmlcomment{Closing part}}
730 \newcommand*{\LWR@printclosechapter}%
731   {\LWR@htmlcomment{Closing chapter}}
732 \newcommand*{\LWR@printclosesection}%
733   {\LWR@htmlcomment{Closing section}}
734 \newcommand*{\LWR@printclosesubsection}%
735   {\LWR@htmlcomment{Closing subsection}}
736 \newcommand*{\LWR@printclosesubsubsection}%
737   {\LWR@htmlcomment{Closing subsubsection}}
738 \newcommand*{\LWR@printcloseparagraph}%
739   {\LWR@htmlcomment{Closing paragraph}}
740 \newcommand*{\LWR@printclosesubparagraph}%
741   {\LWR@htmlcomment{Closing subparagraph}}

```

Lists require closing HTML tags:

---

```

742 \newcommand*{\LWR@printcloselistitem}
743   {\LWR@htmltag{/li}}
744 \newcommand*{\LWR@printclosedescitem}
745   {\LWR@htmltag{/dd}}
746 \newcommand*{\LWR@printcloseitemize}
747   {\LWR@htmltag{/ul}}
748 \newcommand*{\LWR@printcloseenumerate}
749   {\LWR@htmltag{/ol}}
750 \newcommand*{\LWR@printclosedescription}
751   {\LWR@htmltag{/dl}}
752 \end{warpHTML}

```

## 30 Paragraph handling

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

Paragraph tags are or are not allowed depending on many conditions. Section 31 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a L<sup>A</sup>T<sub>E</sub>X 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:** 753 `\begin{warpHTML}`

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

```

754 \newcounter{LWR@spandepth}
755 \setcounter{LWR@spandepth}{0}

```

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

```

756 \newbool{LWR@doingstartpars}
757 \boolfalse{LWR@doingstartpars}

```

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

```

758 \newbool{LWR@doingapar}
759 \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:

---

```

760 \newcommand*{\LWR@ensuredoingapar}{%
761 \ifbool{\LWR@doingstartpars}{%
762 {\global\booltrue{\LWR@doingapar}}}{%
763 {}{}}{%
764 }

```

\LWR@openparagraph

```

765 \newcommand*{\LWR@openparagraph}{%
766 {}{%

```

See if paragraph handling is enabled:

```

767 \ifbool{\LWR@doingstartpars}{%
768 {}{%
    handling pars

```

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

```

769 \ifthenelse{}{%
770 \cnttest{\value{\LWR@lateximagedepth}}{>}{0} \OR%
771 \cnttest{\value{\LWR@spandepth}}{>}{0}{%
772 }{%
    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`.

```

773 {}{%
    no nested par tags

```

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

```

774 {}{%
    yes nest par tags
775 \LWR@htmlltagc{\LWR@tagregularparagraph}{%

```

Manually indent item list labels to avoid left margin intrustion:

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

See if are nested inside an item list:

```

776 \ifnumcomp{\@listdepth}{>}{0}{%
777 {}{%

```

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

```

778 \LWR@orighspace{1in}{%
779 }{}{%

```

Now have started a paragraph.

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

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

```
781 \let\par\LWR@closeparagraph%
782 }% end of yes nest par tags
783 }% end of handling pars
784 {}% not handling pars
785 }
```

`\LWR@closeparagraph`

```
786 \newcommand*\LWR@closeparagraph{%
787 }%
```

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
789 {}% handling pars
```

See if already started a `lateximage` or a `span`:

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

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

```
794 {}% no nested par tags
795 \ifthenelse{%
796 \cnttest{\value{LWR@spandepth}}>}{0}%
797 \AND%
798 \cnttest{\value{LWR@lateximagedepth}}=}{0}%
799 }%
800 {}%
801 \ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}%
802 }%
803 {}%
804 }% no nested par tags
```

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

```
805 { % yes nest par tags
```

Print a closing tag:

```
806 \unskip%
807 \LWR@htmltagc{/\LWR@tagregularparagraph}%
```

No longer doing a paragraph:

```
808 \global\boolfalse{LWR@doingapar}%
809 % Disable the special \env{minipage} \& \cs{hspace} interaction
810 % until a new minipage is found:
811 %   \begin{macrocode}
812 \global\boolfalse{LWR@minipagethispar}%
813 }% end of yes nest par tags
814 }% end of handling pars
```

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

```
815 { % not handling pars
816 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}{%
817 {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
818 {}%
819 }% not handling pars
```

Finish with regular paragraph processing

```
820 \LWR@origpar%
821 }

822 \end{warpHTML}
```

## 31 Paragraph start/stop handling

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

Section 30 has the commands which actually generate the tags.

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

**for HTML output:** 823 `\begin{warpHTML}`

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

```
824 \newcommand*\LWR@startpars{%
825 {%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
827 {}%
```

If not currently in paragraph mode:

```
828 {%
```

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

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

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

```
830 \let\par\LWR@closeparagraph
831
832 }% an intentionally blank line
```

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

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

No <par> tag yet to undo:

```
834 \global\boolfalse{\LWR@doingapar}{%
835 }
```

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

```
836 \newcommand*\LWR@stoppars{%
837 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

839 {%

Print a closing tag:

840 \unskip%

841 \LWR@htmlltagc{/\LWR@tagregularparagraph}%

No longer have an open HTML paragraph:

842 \global\boolfalse{LWR@doingapar}%

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

843 \global\boolfalse{LWR@minipagethispar}

844

845 }% an intentionally blank line

If was not in an HTML paragraph:

846 {}%

See if currently allowing HTML paragraphs:

847 \ifbool{LWR@doingstartpars}{}%

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

848 {%

849 \ClearPreHook{par}%

850 }%

Else: do nothing

851 {}%

no longer in paragraph mode

852 \global\setbool{LWR@doingstartpars}{false}{}%

no `<p>` tag to undo:

853 \global\boolfalse{LWR@doingapar}{}%

854 }

855 \end{warpHTML}

## 32 Page headers and footers

**for HTML & PRINT:** 856 \begin{warpall}

In the following, catcode is manually changes back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
857 \newcommand{\LWR@firstpagetop}{} % for the home page alone
858 \newcommand{\LWR@pagetop}{} % for all other pages
859 \newcommand{\LWR@pagebottom}{}
860
861 \newcommand{\LWR@setfirstpagetopb}[1]{%
862 \renewcommand{\LWR@firstpagetop}{#1}
863 \catcode`\_=8
864 }
```

\SetFirstPageTop {*text and logos*}

```
865 \newcommand{\SetFirstPageTop}{%
866 \catcode`\_=12
867 \LWR@setfirstpagetopb
868 }

869 \newcommand{\LWR@setpagetopb}[1]{%
870 \renewcommand{\LWR@pagetop}{#1}
871 \catcode`\_=8
872 }
```

\SetPageTop {*text and logos*}

```
873 \newcommand{\SetPageTop}{%
874 \catcode`\_=12
875 \LWR@setpagetopb
876 }

877 \newcommand{\LWR@setpagebottomb}[1]{%
878 \renewcommand{\LWR@pagebottom}{#1}
879 \catcode`\_=8
880 }
```

\SetPageBottom {*text and logos*}

```
881 \newcommand{\SetPageBottom}{%
882 \catcode`\_=12
883 \LWR@setpagebottomb
884 }
```

```
885 \end{warpall}
```

## 33 CSS

**for HTML output:** 886 \begin{warpHTML}

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

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

```
\NewCSS {\langle new-css-filename.css \rangle}
```

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

```
888 \newcommand*{\LWR@newcssb}[1]{%
889 \renewcommand*{\LWR@currentcss}{#1}
890 \catcode`\_=8
891 }
892
893 \newcommand*{\NewCSS}{
894 \catcode`\_=12
895 \LWR@newcssb
896 }
897 \end{warpHTML}
```

**for PRINT output:** 898 \begin{warpprint}
899 \newcommand\*{\NewCSS}[1]{}
900 \end{warpprint}

## 34 HTML meta description

**for HTML output:** 901 \begin{warpHTML}

\LWR@currentHTMLdescription The HTML meta description to use.

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

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

```
902 \newcommand{\LWR@currentHTMLdescription}{}  
  
\NewHTMLdescription {\langle New HTML meta description.\rangle}  
Assigns the HTML file's description meta tag  
  
903 \newcommand{\NewHTMLdescription}[1]{%  
904 \renewcommand{\LWR@currentHTMLdescription}{#1}  
905 }  
906  
907 \end{warpHTML}  
  
for PRINT output: 908 \begin{warpprint}  
909 \newcommand{\NewHTMLdescription}[1]{}  
910 \end{warpprint}
```

## 35 Footnotes

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

### 35.1 Regular page footnotes

In L<sup>A</sup>T<sub>E</sub>X 2<sub><</sub> printed documents, footnotes are placed at the bottom of the printed page. For HTML, footnotes will be replaced by page notes using the `pagenote` package. Footnotes will then appear at the bottom of the HTML page.

See section 35.4 for the implementation.

### 35.2 Minipage footnotes

See section 62.2 for minipage footnotes.

### 35.3 Titlepage thanks

See section 42.6 for titlepage footnotes.

### 35.4 Regular page footnote implementation

**for HTML output:** 911 \begin{warpHTML}

Delayed in case the user's preamble loaded and adjusted pagenote:

912 \AtEndPreamble{

\notedivision Prints the footnote heading at the end of the page.

913 \renewcommand\*\{\notedivision\}\  
914 {\BlockClassSingle{footnoteheader}{}{}}

To print the word "Notes:" at the top of the footnotes:

---

\renewcommand\*\{\notedivision\}| \\  
{\BlockClassSingle{footnoteheader}{Notes:}}

---

\pagenotesubhead {\langle num \rangle} {\langle title \rangle}

Don't print page note headings in HTML page notes.

915 \renewcommand{\pagenotesubhead}[2]{}{}

\footnote {[id]} {\langle text \rangle} Footnotes now become page notes.

916 \let\footnote\LWR@origpagenote

\notenuminnotes {\langle number \rangle}

Sets the page notes to use superscripts to match standard classes.

917 \renewcommand{\notenuminnotes}[1]{%  
918 \textsuperscript{\#1} }

\noteentry {\langle 1: note number \rangle} {\langle 2: id \rangle} {\langle 3: text \rangle} {\langle 4: pagenum \rangle}

919 \let\LWR@orignoteentry\noteentry  
920 \renewcommand{\noteentry}[4]{%

\@currentlabel A label may be used inside a note, so the proper \@currentlabel must be set  
\cref@currentlabel

inside a group. Likewise for `cleveref`'s `\cref@currentlabel`.

```
921 \begingroup%
922 \renewcommand*{\@currentlabel}{#1}%
923 \renewcommand*{\cref@currentlabel}{[pagenote] [#1] [] {#1}}%
```

Re-use `pageref`'s code, with an extra endgroup:

```
924 \LWR@orignoteentry{#1}{#2}{#3}{#4}%
925 \endgroup%
926 }
```

`\LWR@printpendingnotes` Prints the page notes at the end of the section.

```
927 \newcommand*{\LWR@printpendingnotes}{
```

If there are footnotes to print, open a new `div` and print the notes.

```
928 \ifthenelse{\cnttest{\value{pagenote}}{>}{0}}{%
929 {
930 \begin{BlockClass}{footnotes}
```

Print any pending endnotes, allowing new endnotes to accumulate.

```
931 \LWR@origprintnotes*
```

Start over with new page notes.

```
932 \setcounter{pagenote}{0}
933 \end{BlockClass}
934 }
935 {}
936 }
```

```
937 }% \AtEndPreamble
```

```
938 \end{warpHTML}
```

## 36 Marginpars

**for HTML output:** 939 `\begin{warpHTML}`

```
\marginpar [⟨left⟩] {⟨right⟩}
```

---

```

940 \renewcommand{\marginpar}[2] []{%
941 \LWR@htmlspanclass{marginpar}{#2}%
942 }

943 \end{warpHTML}

```

## 37 Splitting HTML files

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

**for HTML & PRINT:** 944 `\begin{warpall}`

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

```

945 \newcounter{FileDepth}
946 \setcounter{FileDepth}{-5}

```

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

```

947 \newbool{CombineHigherDepths}
948 \booltrue{CombineHigherDepths}

949 \end{warpall}

```

**for HTML output:** 950 `\begin{warpHTML}`

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

```

951 \newcommand*{\LWR@thisfilename}{}%
```

`\LWR@thisnewfilename` The filename being sanitized.

```

952 \newcommand*{\LWR@thisnewfilename}{}%
```

`\LWR@filenamenoblanks` {*filename*}

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



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

```
953 \newcommand*{\LWR@filenamenoblanks}[1]{%
954 \begingroup
```

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

```
955 \renewcommand*{\HTMLunicode}[1]{}
956 \renewcommand*{\HTMLentity}[1]{##1}
957 \renewcommand*{\LWR@htmntagc}[1]{}
958 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

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

```
959 \renewcommand*{\&}{and}
960 \renewcommand{\textit}[1]{##1}
961 \renewcommand{\textsc}[1]{##1}
962 \renewcommand{\textsl}[1]{##1}
963 \renewcommand{\textbf}[1]{##1}
964 \renewcommand{\texttt}[1]{##1}
965 \renewcommand{\textsf}[1]{##1}
966 \renewcommand{\textrm}[1]{##1}
967 \renewcommand{\textsuperscript}[1]{##1}
968 \renewcommand{\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
969 \edef\LWR@thisfilename{#1}
970 \fullexpandarg
```

Convert spaces into hyphens:

```
971 \StrSubstitute{\LWR@thisfilename}{ }{-}[\LWR@thisfilename]
```

Convert punctuation into hyphens:

```
972 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
973 \StrSubstitute{\LWR@thisfilename}{'}{-}[\LWR@thisfilename]
974 \StrSubstitute{\LWR@thisfilename}{%}
975 {\LWR@origampersand}{-}[\LWR@thisfilename]
976 \StrSubstitute{\LWR@thisfilename}{+}{-}[\LWR@thisfilename]
977 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
978 \StrSubstitute{\LWR@thisfilename}{/}{-}[\LWR@thisfilename]
979 \StrSubstitute{\LWR@thisfilename}{:}{-}[\LWR@thisfilename]
980 \StrSubstitute{\LWR@thisfilename}{;}{-}[\LWR@thisfilename]
```

```

981 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
982 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
983 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
984 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
985 \StrSubstitute{\LWR@thisnewfilename}{%}
986 {\textless}{-}[\LWR@thisnewfilename]
987 \StrSubstitute{\LWR@thisnewfilename}{%}
988 {\textgreater}{-}[\LWR@thisnewfilename]
989 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
990 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
991 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
992 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
993 \StrSubstitute{\LWR@thisnewfilename}{\|}{-}[\LWR@thisnewfilename]
994 \StrSubstitute{\LWR@thisnewfilename}{%}
995 {\textbackslash}{-}[\LWR@thisnewfilename]
996 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
997 \StrSubstitute{\LWR@thisnewfilename}{~{}{-}[\LWR@thisnewfilename]
998 %      "~-{}" for babel
999 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
1000 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
1001 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
```

Convert short words:

```

1002 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
1003 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
1004 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
1005 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
1006 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
1007 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
1008 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
1009 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
1010 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
1011 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
1012 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
1013 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
1014 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
1015 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
1016 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
1017 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1018 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
1019 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```

1020 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
1021 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
1022 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
1023 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1024 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
```

```

1025 %     emdash
1026 \StrSubstitute{\LWR@thisfilename}{-}{-}[\LWR@thisfilename]
1027 %     endash
1028 \global\let\LWR@thisfilename\LWR@thisfilename% return a global result
1029 \endgroup
1030 }

```

\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.

```
1031 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

1032 \LWR@htmlelementclassend{section}{textbody}
1033
1034 \LWR@printpendingnotes
1035
1036 \LWR@htmlelement{footer}
1037
1038 \LWR@pagebottom
1039
1040 \LWR@htmlelementend{footer}

```

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

```
1041 \ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{{0}}{\LWR@botnavigation}{}
```

End of this HTML file:

```

1042 \LWR@stopars
1043 \LWR@htmlltag{/body}\LWR@orignewline
1044 \LWR@htmlltag{/html}\LWR@orignewline
1045
1046 \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.

```

1047 \ifbool{FileSectionNames}{%
1048 {\LWR@filenamenoblanks{#1}}%
1049 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%

```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
1050 \LWR@htmlblockcomment{%
1051 \LWR@origtexttt{|Start file|%
1052 \LWR@htmlsectionfilename{\LWR@thisfilename}|}%
1053 }
```

At the top of the starting file:

```
1054 \LWR@stoppars
1055
1056 \LWR@filestart{ -- #1}%
there is an EMdash in front of the #1
1057
1058 \LWR@topnavigation
1059
1060 \LWR@htmlelement{header}
1061
1062 \LWR@pagetop
1063
1064 \LWR@htmlelementend{header}
1065
1066 \ifcsvvoid{thetitle}{}{\LWR@printthetitle}
1067
1068 \LWR@sidetoc
1069
1070 \LWR@htmlelementclass{section}{textbody}
1071
1072 \LWR@stoppars
1073
1074 \setcounter{LWR@latestautopage}{\value{page}}
1075 }

1076 \end{warpHTML}
```

## 38 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L<sup>A</sup>T<sub>E</sub>X code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

- Unicode Section names and the resulting filenames with accented characters are partially supported, depending on the ability of `pdflatex` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that

`pdflatex` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.

X<sub>E</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X directly support accented section and file names.

**for HTML output:** 1077 `\begin{warpHTML}`

## 38.1 Book class commands

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

```
1078 \newbool{LWR@mainmatter}
1079 \DeclareDocumentCommand{\mainmatter}{}{%
1080 \booltrue{LWR@mainmatter}}%
1081 }
```

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

```
1082 \DeclareDocumentCommand{\frontmatter}{}{%
1083 \boolfalse{LWR@mainmatter}}%
1084 }
```

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

```
1085 \DeclareDocumentCommand{\backmatter}{}{%
1086 \boolfalse{LWR@mainmatter}}%
1087 }
```

## 38.2 Sectioning support macros

`\LWR@sectionnumber {<section type>}`

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

```
1088 \newcommand*{\LWR@sectionnumber}[1]{%
1089 \InlineClass{sectionnumber}{#1} %
1090 }
```

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

`\LWR@createautosec {<section type>}`

Create an autosection tag.

```
1091 \newcommand*{\LWR@createautosec}[1]{%
1092 \LWR@htmlltag{#1 id="autosec-\thepage"}{}%
1093 }
```

**\LWR@pushoneclose** {*depth*} {*printclose*} Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

 **\LWR@stoppars** must be executed first.

```
1094 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

**\LWR@startnewdepth** {*depth*} {*printclose*}

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 **\LWR@stoppars** must be executed first.

```
1095 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1096 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1097 \LWR@pushoneclose{#1}{#2}%
1098 }
```

**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.

```
1099 \newcounter{LWR@prevFileDepth}
1100 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

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

The common actions for the high-level sectioning commands.

```
1101 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
1102 \LWR@traceinfo{LWR@section}%
1103 \LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
1104 \global\boolefalse{LWR@minipagethispar}%
```

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

```
1105 \LWR@orignewpage%
1106
```

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

```
1107 \LWR@traceinfo{LWR@section: testing about to start a new HTML file}%
1108 \IfBooleanTF{#1}{}{%
1109 \ifthenelse{%
1110 \cnttest{\csuse{LWR@depth#4}}{\leq}{\value{FileDepth}}}{%
1111 \AND%
1112 \((%
1113 \NOT\boolean{CombineHigherDepths})\OR%
1114 \cnttest{\csuse{LWR@depth#4}}{\leq}{\value{LWR@prevFileDepth}}}{%
1115 \))%
1116 }%
```

If so: start a new HTML file:

```
1117 {%
1118 \LWR@traceinfo{LWR@section: new HTML file}%
}
```

See if there was an optional TOC name entry:

```
1119 \IfNoValueTF{#2}{%
```

If no optional entry

```
1120 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
1121 {\LWR@newhtmlfile{#2}}%
1122 }% new file
```

Else: No new HTML file:

```
1123 {}% not new file
1124 }% not starred
```

Remember this section's name for `\nameref`:

```
1125 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
1126 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
```

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

```
1127
1128 \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}
1129
```

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

```
1130 \ifthenelse{%
1131 \cnttest{\csuse{LWR@depth#4}}{>}{\LWR@depthparagraph}%
1132 }%
1133 {\LWR@startpars}
1134 {}
```

Create the opening tag with an autosec:

```
1135 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

If not starred, step counter and add to TOC:

```
1136 \IfBooleanTF{#1}%
1137 {}% starred
1138 {}% not starred
```

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

```
1139 \ifthenelse{%
1140 \cnttest{\csuse{LWR@depth#4}}{<}{\value{secnumdepth}}%
1141 }%
1142 {}% if secnumdepth
```

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

```
1143 \LWR@traceinfo{LWR@section: about to test main matter}%
1144 \ifbool{LWR@mainmatter}%
1145 {}%
1146 \LWR@traceinfo{LWR@section: yes mainmatter}%
1147 \refstepcounter{#4}%
```

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

```
1148 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1149 \addcontentsline{toc}{#4}%
1150 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1151 \IfValueTF{#2}{#2}{#3}}%
```

```
1152 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1153 }% end of if main matter
```

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

```
1154 {%
1155 \LWR@traceinfo{LWR@section: no main matter}%
1156 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1157 }% end of not main matter
1158 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
1159 {%
1160 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1161 }%
```

For part, print the section type:

```
1162 \ifbool{LWR@mainmatter}%
1163 {%
1164 \ifthenelse{%
1165 {(\cnttest{\csuse{LWR@depth#4}}{<=})}%
1166 {\value{secnumdepth}}}\ \AND
1167 {(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart})}%
1168 }%
1169 {\csuse{#4name}-%}%
1170 {}%
```

Print the section number:

```
1171 \LWR@traceinfo{LWR@section: about to print section number}%
1172 \ifthenelse{%
1173 {(\cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}})}%
1174 }%
1175 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1176 {}%
1177 \LWR@traceinfo{LWR@section: finished print section number}%
1178 }{}%
1179 }% end of not starred
```

Print the section name:

```
1180 #3
```

close the heading tag, such as /H2

```
1181 \LWR@htmlltag{\csuse{LWR@tag#4end}}%
```

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

```
1182 \label{autopage-\thepage} %
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
1183 \ifthenelse{%
1184 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
1185 {\LWR@startpars}%
1186 {}%
```

If not starred, remember the previous depth to possibly trigger a new HTML page.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a `\listoftables` was tested in the middle of the document. The `\chapter*` for the list was not allowing a new HTML page for the section following it while `CombineHigherDepths` was true.

```
1187 \IfBooleanTF{#1}{}{%
1188 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
1189 }% not starred
1190 \LWR@traceinfo{LWR@section: done}%
1191 }
```

### 38.3 \section and friends

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

```
1192 \@ifundefined{chapter}%
1193 {}%
1194 {}%
1195 \DeclareDocumentCommand{\part}{s o m}{%
1196 \LWR@stoppars%
1197 %
1198 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
1199 %
1200 \LWR@section{#1}{#2}{#3}{part}%
1201 }%
1202 }
```

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

```
1203 \@ifundefined{chapter}%
1204 {}%
1205 {}%
```

```
1206 \DeclareDocumentCommand{\chapter}{s o m}{%
1207   \LWR@traceinfo{chapter #3}%
1208   \LWR@stoppars%
1209   \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
1210
1211   \LWR@traceinfo{chapter: about to \LWR@section}%
1212   \LWR@section{#1}{#2}{#3}{chapter}%
1213   \LWR@traceinfo{chapter: done}%
1214 }
1215 }

\section * [<TOC name>] {<name>}

1216 \DeclareDocumentCommand{\section}{s o m}{%
1217   \LWR@stoppars%
1218
1219   \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
1220
1221   \LWR@section{#1}{#2}{#3}{section}%
1222 }

\subsection * [<TOC name>] {<name>}

1223 \DeclareDocumentCommand{\subsection}{s o m}{%
1224   \LWR@stoppars%
1225
1226   \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1227
1228   \LWR@section{#1}{#2}{#3}{subsection}%
1229 }

\subsubsection * [<TOC name>] {<name>}

1230 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1231   \LWR@stoppars%
1232
1233   \LWR@startnewdepth{\LWR@depthsubsubsection}{%
1234     \LWR@printclosesubsubsection}%
1235
1236   \LWR@section{#1}{#2}{#3}{subsubsection}%
1237 }

\paragraph * [<TOC name>] {<name>}

1238 \DeclareDocumentCommand{\paragraph}{s o m}{%
1239   \LWR@stoppars%
1240
```

```

1241 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1242
1243 \LWR@section{#1}{#2}{#3}{paragraph}%
1244 }

\subparagraph * [<TOC name>] {<name>}

1245 \DeclareDocumentCommand{\subparagraph}{s o m}{%
1246 \LWR@stoppars%
1247
1248 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
1249
1250 \LWR@section{#1}{#2}{#3}{subparagraph}%
1251 }

1252 \end{warpHTML}

```

## 39 Starting a new file

**for HTML output:** 1253 \begin{warpHTML}

\MetaLanguage Default language for the HTML lang tag.

```
1254 \newcommand*{\MetaLanguage}{en-US}
```

\LWR@filestart {<title\\_suffix>}

Creates the opening HTML tags.

```
1255 \newcommand*{\LWR@filestart}[1]{
```

Locally temporarily disable direct-formatting commands:

```

1256 \begingroup
1257 \renewcommand{\textit}[1]{##1}% not used in filenames
1258 \renewcommand{\textsc}[1]{##1}
1259 \renewcommand{\textsl}[1]{##1}
1260 \renewcommand{\textbf}[1]{##1}
1261 \renewcommand{\texttt}[1]{##1}
1262 \renewcommand{\textsf}[1]{##1}
1263 \renewcommand{\textrm}[1]{##1}
1264 \renewcommand{\textsuperscript}[1]{##1}
1265 \renewcommand{\textsubscript}[1]{##1}
1266 \renewcommand*{\HTMLunicode}[1]{}

```

```
1267 \renewcommand*{\HTMLentity}[1]{}
1268 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1269 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

Create the page's HTML header:

```
1270 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
1271 \LWR@htmltag{html lang="\MetaLanguage"{} }\LWR@orignewline
```

Start of the meta data:

```
1272 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
1273 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

`\lwarf` is the generator:

```
1274 \LWR@htmltag{meta name="generator" content="LaTeX lwarf package" /}%
1275     \LWR@orignewline
```

If there is a description, add it now:

```
1276 \ifdefempty{\LWR@currentHTMLdescription}{}{%
1277 \LWR@htmltag{%
1278 meta name="description" content="\LWR@currentHTMLdescription" /}%
1279     \LWR@orignewline
1280 }%
```

Mobile-friendly viewport:

```
1281 \LWR@htmltag{meta name="viewport" %
1282 content="width=device-width, initial-scale=1.0" /}%
1283     \LWR@orignewline
```

IE patch:

```
1284 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
1285 \LWR@htmltag{%
1286 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
1287 \LWR@htmltag{/script}\LWR@orignewline
1288 \LWR@htmltag{! [endif]{-}{-}}\LWR@orignewline
```

The page's title:

---

```

1289 \ifcsvoid{thetitle}{}{%
1290 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
1291 }%

```

The page's stylesheet:

```

1292 \LWR@htmltag{%
1293 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1294 \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.

```

1295 \ifbool{mathjax}{%
1296 {%
1297 \boolfalse{LWR@verbtags}%
1298 \VerbatimInput{l warp_mathjax.txt}%
1299 \booltrue{LWR@verbtags}%
1300 \LWR@stopars
1301 }% end of mathjax
1302 {}}%

```

End of the header:

```

1303 \LWR@htmltag{/head}\LWR@orignewline

```

Start of the body:

```

1304 \LWR@htmltag{body}\LWR@orignewline
1305 \endgroup
1306 }%

```

```

1307 \end{warpHTML}

```

## 40 Starting HTML output

**for HTML output:** 1308 \begin{warpHTML}

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

```

1309 \newcommand*\LWR@LwarpStart{%
1310 {%

```

Expand and detokenize \HomeHTMLFileName and \HTMLFileName:

```
1311 \edef\LWR@strresult{\HomeHTMLFileName}
1312 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1313 \edef\LWR@strresult{\HTMLFileName}
1314 \edef\HTMLFileName{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
1315 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
1316 \LWR@origscriptsize%
```

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

```
1317 \LWR@origraggedright%
1318 \let\\ \LWR@endofline%
```

Spread the lines for pdftotext to read them well:

```
1319 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
1320 \setlength{\parindent}{0pt}
1321 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
1322 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1323 \setlength{\belowcaptionskip}{-3ex}
```

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

```
1324 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 1325 \renewcommand*{\centering}{}%
1326 \renewcommand*{\raggedleft}{}%
1327 \renewcommand*{\raggedright}{}%
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
1328 \let\LWR@origtabular\tabular
1329 \let\LWR@origendtabular\endtabular
1330 \let\tabular\LWR@tabular
1331 \let\endtabular\endLWR@tabular
```

Float captions:

```
1332 \let\LWR@origcaption\caption
```

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

```
1333 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
1334 \let\TeX\LWR@TeX
1335 \let\LaTeX\LWR@LaTeX
1336 \let\LuaTeX\LWR@LuaTeX
1337 \let\LuaLaTeX\LWR@LuaLaTeX
1338 \let\XeTeX\LWR@XeTeX
1339 \let\XeLaTeX\LWR@XeLaTeX
1340 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
1341 \let\textcolor\LWR@textcolor
1342 \let\pagecolor\LWR@pagecolor
1343 \let\colorbox\LWR@colorbox
1344 \let\fcolorbox\LWR@fcolorbox
1345 \let\rotatebox\LWR@rotatebox
1346 \let\scalebox\LWR@scalebox
1347 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
1348 \global\boolfalse{LWR@doingapar}
1349 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
1350 \LWR@filestart{}
1351 \LWR@htmntag{header}\LWR@orignewline
1352 \LWR@startpars
1353 \LWR@firstpagetop
1354 \LWR@stoppars
1355 \LWR@htmntag{/header}\LWR@orignewline
```

---

```
1356 \LWR@htmltag{section class="textbody"{}}
1357 \LWR@origpagestyle{empty}
```

Document and page settings:

```
1358 \mainmatter
1359 \LWR@origpagenumbering{arabic}
```

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

```
1360 \if@titlepage
1361   \thanksmarkseries{arabic}
1362 \else
1363   \thanksmarkseries{fnsymbol}
1364 \fi
```

Initial default patch for fancyvrb:

```
1365 \fvset{frame=none}%
```

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

```
1366 \catcode`\&=\active
```

Allow HTML paragraphs to begin:

```
1367 \LWR@startpars
1368 }
```

```
1369 \end{warpHTML}
```

## 41 Ending HTML output

**for HTML output:** 1370 \begin{warpHTML}

\LWR@requesttoc {⟨boolean⟩} {⟨suffix⟩} Requests that a toc, lof, or lot be generated.

```
1371 \newcommand*\LWR@requesttoc[2]{%
1372 \ifbool{#1}{%
1373 {%
1374 \expandafter\newwrite\csuse{tf@#2}%
1375 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
1376 }{}%
1377 }}
```

\LWR@LwarpEnd Final stop of all HTML output:

```
1378 \newcommand*\{\LWR@LwarpEnd}
1379 {
1380 \LWR@stopars
1381 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
1382 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
1383 \LWR@printpendingnotes
```

Create the footer:

```
1384 \LWR@htmlelement{footer}
1385
1386 \LWR@pagebottom
1387
1388 \LWR@htmlelementend{footer}
```

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

```
1389 \ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}{}
```

```
1390 \LWR@stopars% final stop of all paragraphs
```

Finish the HTML file:

```
1391 \LWR@htmlltag{/body}\LWR@orignewline
1392 \LWR@htmlltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
1393 \LWR@orignewpage
```

For `lateximage` commands:

```
1394 \immediate\closeout\LWR@file
1395 }
```

```
1396 \end{warpHTML}
```

## 42 Titles and the **titling** package

Supports and extends the **titling** package.

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

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

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

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

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

`\maketitlehookaaa` **\maketitlehookaaa:** Between the title and the subtitle.

`\prepublished` **\prepublished:** Before the “published” field.

`\postpublished` **\postpublished:** After the “published” field.

`\presubtitle` **\presubtitle:** Before the subtitle.

`\postsubtitle` **\postsubtitle:** After the subtitle.

`\printthanks` **\printthanks:** has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` is not used.

 **No footnotes!** Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

---

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

---

## 42.1 Setting the title, etc.

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

```
\published    \@title, \@subtitle, \@author, etc. store the values as originally assigned, in-
               \title   cluding any \thanks, \and, or \affiliation. These are low-level macros intended
               \subtitle to be used by other macros only inside a titlepage or titlingpage, and are used
               \author  by \maketitle. The author is printed inside a single-column table, which becomes
               \date    multiple single-column tables if multiples authors are included.

\printpublished \printtitle, \printsubtitle, etc. are user-level macros intended to be used
\printtitle     in titlepage and titlingpage environments in cases where \maketitle is not
\printsubtitle desired. These commands preserve the \thanks, etc., and should not be used in
\printauthor   the main text. The author is printed inside a single-column table, which becomes
\printdate     multiple single-column tables if multiples authors are included.

\thepublished \thetitle, \thesubtitle, \theauthor, etc. are user-level sanitized versions which
\thetitle      have removed the \thanks and \affiliation, and \and is changed for inline text
\thesubtitle   usage. The author is printed inline without \affiliation or \thanks, with \and
\theauthor    placing commas between multiple authors. Thus, these commands are to be used
\thedate      in the main text whenever the user wishes to refer to the document's title and
               such. One practical use for this is to place the authors at the bottom of each HTML
               page, such as:

\SetPageBottom {\langle text\rangle}
```

---

```
\SetPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

---

`\author {\langle author\rangle}` While using `\maketitle`, the author is treated as a single-column table
`\and` and the `\and` feature finishes the current table then starts a new one for the next
author. Each author thus is placed into its own table, and an affiliation may be
placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

After `\maketitle` has completed, `\theauthor` retains the definition of the author,
but `\and` is changed to become a comma and a space, intending to print the authors

names separated by spaces. This fails when affiliations are included on their own table rows.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

```
\affiliation {<text>}
```

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

**for HTML output:** 1397 `\begin{warpHTML}`  
 1398 `\newrobustcmd{\affiliation}[1]{\ \ \ \InlineClass{affiliation}{#1}}`  
 1399 `\end{warpHTML}`

**for PRINT output:** 1400 `\begin{warpprint}`  
 1401 `\newrobustcmd{\affiliation}[1]{\ \ \ \textsc{\small#1}}`  
 1402 `\end{warpprint}`

The following are based on the original titling code:

**for HTML & PRINT:** 1403 `\begin{warpall}`

```
\author {<text>}
```

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

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

```
1404 \renewcommand{\author}[1]{%  

  1405 \gdef\@author{#1}  

  1406 \begingroup  

  1407 \renewcommand{\thanks}{}  

  1408 \renewcommand{\and}{\unskip, }  

  1409 \renewcommand{\thanksmark}{}  

  1410 \renewcommand{\thanksgap}{}  

  1411 \renewcommand{\affiliation}{}  

  1412 \protected@xdef\theauthor{#1}  

  1413 \endgroup}
```

```
\published {<text>}
```

```
1414 \newcommand{\published}[1]{%  

  1415 \gdef\@published{#1}}
```

```

1416 \begingroup
1417   \renewcommand{\thanks}{\relax}
1418   \renewcommand{\thanksmark}{\relax}
1419   \renewcommand{\thanksgap}{\relax}
1420   \protected@xdef\thepublished{\#1}
1421 \endgroup
1422 }
1423 \newcommand{\@published}{}
1424 \newcommand{\thepublished}{}

\subtitle {\langle text\rangle}

1425 \newcommand{\subtitle}[1]{%
1426   \gdef\@subtitle{\#1}
1427   \begingroup
1428     \renewcommand{\thanks}{\relax}
1429     \renewcommand{\thanksmark}{\relax}
1430     \renewcommand{\thanksgap}{\relax}
1431     \protected@xdef\thesubtitle{\#1}
1432   \endgroup
1433 }
1434 \newcommand{\@subtitle}{}
1435 \newcommand{\thesubtitle}{}

1436 \end{warpall}

```

## 42.2 Changes to HTML titlepage and titlingpage

for HTML output: 1437 \begin{warpHTML}

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

```

1438 \renewenvironment*{titlepage}%
1439   {\BlockClass{titlepage}\LWR@subminipage}%
1440   {\LWR@endsubminipage\endBlockClass}

```

Env **titlingpage**

```

1441 \renewenvironment*{titlingpage}%
1442 {%

```

Start an HTML titlepage div:

```
1443 \begin{titlepage}
```

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

```
1444 \LWR@maketitlesetup
1445 \let\maketitle\LWR@titlingmaketitle
1446 }
1447 {
```

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

```
1448 \end{titlepage}
```

Reset the footnote counter:

```
1449 \@bscontmark
1450 }
```

```
1451 \end{warpHTML}
```

**for HTML & PRINT:** 1452 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

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

```
1453 \newcommand*\printthanks{\@thanks}
```

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

```
1454 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1455 \end{warpall}
```

### 42.3 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

**for HTML output:** 1456 `\begin{warpHTML}`

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

```
1457 \newcommand{\prepublished}[1]{%
1458 \def\@bsprepublished{\BlockClass{published}\#1}%
1459 }
1460
```

```
1461 \newcommand{\postpublished}[1]{%
1462 \def\@bspostpublished{\#1\endBlockClass}%
1463 }
1464
1465 \renewcommand{\pretitle}[1]{%
1466 \def\@bspretitle{\#1\LWR@stoppars\LWR@htmltag{h1}}%
1467 }
1468
1469 \renewcommand{\posttitle}[1]{%
1470 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
1471 }
1472
1473 \newcommand{\presubtitle}[1]{%
1474 \def\@bspresubtitle{\BlockClass{subtitle}\#1}%
1475 }
1476
1477 \newcommand{\postsubtitle}[1]{%
1478 \def\@bspostsubtitle{\#1\endBlockClass}%
1479 }
1480
1481 \renewcommand{\preauthor}[1]{%
1482 \def\@bspreauthor{\BlockClass{author}\#1}%
1483 }
1484
1485 \renewcommand{\postauthor}[1]{%
1486 \def\@bspostauthor{\#1\endBlockClass}%
1487 }
1488
1489 \renewcommand{\predate}[1]{%
1490 \def\@bspredate{\#1\BlockClass{titledate}}%
1491 }
1492
1493 \renewcommand{\postdate}[1]{%
1494 \def\@bspostdate{\endBlockClass\#1}%
1495 }
1496
1497 \prepublished{\begin{center}}
1498 \postpublished{\par\end{center}}
1499
1500 \pretitle{\begin{center}}
1501 \posttitle{\par\end{center}}
1502
1503 \presubtitle{\begin{center}}
1504 \postsubtitle{\par\end{center}}
1505
1506 \preauthor{\begin{center}}%
1507 \begin{tabular}[t]{c}%
1508 }%
1509 \postauthor{\end{tabular}\par\end{center}}%
1510
```

```
1511 \predate{\begin{center}}
1512 \postdate{\par\end{center}}
```

\printpublished

```
1513 \newcommand*\printpublished{%
1514 \ifthenelse{\equal{\thepublished}{}}%
1515 {}%
1516 {%
1517 \begin{BlockClass}{published}%
1518 \@published%
1519 \end{BlockClass}%
1520 }%
1521 }
```

\printtitle

```
1522 \newcommand*\printtitle{%
1523 {%
1524 \LWR@stoppars%
1525 \LWR@htmlltag{h1}%
1526 \@title%
1527 \LWR@htmlltag{/h1}%
1528 \LWR@startpars%
1529 }}
```

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

```
1530 \newcommand*\LWR@printthetitle{%
1531 {%
1532 \LWR@stoppars%
1533 \LWR@htmlltag{h1}%
1534 \@thetitle%
1535 \LWR@htmlltag{/h1}%
1536 \LWR@startpars%
1537 }}
```

\printsubtitle

```
1538 \newcommand*\printsubtitle{%
1539 \ifthenelse{\equal{\thesubtitle}{}}%
1540 {}%
1541 {%
1542 \begin{BlockClass}{subtitle}%
1543 \@subtitle%
1544 \end{BlockClass}}
```

```

1545 }
1546 }

\printauthor

1547 \newcommand*{\printauthor}{%
1548 \begin{BlockClass}{author}
1549 \begin{tabular}{c}\@author\end{tabular}
1550 \end{BlockClass}
1551 }

\printdate

1552 \newcommand*{\printdate}{%
1553 \begin{BlockClass}{titledate}
1554 \@date
1555 \end{BlockClass}
1556 }

1557 \end{warpHTML}

```

#### 42.4 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

**for PRINT output:** 1558 `\begin{warpprint}`

```

\printpublished

1559 \newcommand*{\printpublished}{{\Large\scshape\@published}}

\printtitle

1560 \newcommand*{\printtitle}{{\Huge\@title}}


\printsubtitle

1561 \newcommand*{\printsubtitle}{{\Large\itshape\@subtitle\bigskip}}


\printauthor

1562 \newcommand*{\printauthor}{%
1563   {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}}

```

```
\printdate
1564 \newcommand*{\printdate}{{\small\textrit{\@date}}}
```

## 42.5 \maketitle for print output

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

```
1565 \providecommand{\maketitle}{}
1566 \if@titlepage
1567   \renewcommand{\maketitle}{\begin{titlepage}%
1568     \let\footnotesize\small
1569     \let\footnoterule\relax
1570     \let\footnote\thanks
1571     \obsmarkseries
1572     \def\@makefnmark{\rlap{\textsuperscript{\%}}
1573       \normalfont\obstthanksheadpre \tmark \obstthanksheadpost}}\%
1574     \long\def\@makefntext##1{\makethanksmark ##1}
1575   \null\vfil
1576   \vskip 60\p@
1577   \vspace*{\droptitle}
1578   \maketitlehooka
1579   \ifcsempty{@published}
1580 {}
1581 {\obsprepublished \published \obspostpublished}\maketitlehookaa}
1582 {\obspretitle \title \obsposttitle}
1583 \ifcsempty{@subtitle}
1584 {}
1585 {\maketitlehookaaa{\obspresubtitle \subtitle \obspostsubtitle}}
1586 \maketitlehookb
1587 {\obspreauthor \author \obspostauthor}
1588 \maketitlehookc
1589 {\obspredate \date \obspostdate}
1590 \maketitlehookd
1591 \par
1592 \thanks
1593 \vfil\null
1594 \end{titlepage}%
1595 \obscntmark % \setcounter{footnote}{0}%
1596 %% \obsmtitleempty
1597 } % end titlepage defs
1598 \else
1599   \renewcommand{\maketitle}{\par
1600   \begin{group}
1601     \obsmarkseries
1602     \def\@makefnmark{\rlap{\textsuperscript{\%}}
1603       \normalfont\obstthanksheadpre \tmark \obstthanksheadpost}}\%
```

```
1604     \long\def\@makefntext##1{\makethanksmark ##1}
1605     \if@twocolumn
1606         \ifnum \col@number=\@ne
1607             \@maketitle
1608         \else
1609             \twocolumn[\@maketitle]%
1610         \fi
1611     \else
1612         \newpage
1613         \global\@topnum\z@
1614         \@maketitle
1615     \fi
1616     \thispagestyle{plain}\@thanks
1617     \endgroup
1618     \@bscontmark % \setcounter{footnote}{0}%
1619 %%% \@bsmtitleempty
1620 } % end non-titlepage
1621
1622 \def\@maketitle{%
1623     \newpage
1624     \null
1625     \vskip 2em%
1626     \vspace*{\droptitle}
1627     \maketitlehooka
1628     \ifcsempty{@published}
1629     {}
1630 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1631 {\@bspretitle \@title \@bsposttitle}
1632 \ifcsempty{@subtitle}
1633 {}
1634 {\maketitlehookaaa{@bspresubtitle \@subtitle \@bspostsubtitle}}
1635 \maketitlehookb
1636 {\@bspreauthor \@author \@bspostauthor}
1637 \maketitlehookc
1638 {\@bspredate \@date \@bspostdate}
1639 \maketitlehookd
1640 \par
1641 \vskip 1.5em}
1642 \fi
1643
1644 \providecommand{\maketitlehookaa}{}
1645
1646 \providecommand{\maketitlehookaaa}{}
1647
1648 \newcommand{\prepublished}[1]{%
1649 \def\@bsprepublished{#1}%
1650 }
1651
1652 \newcommand{\postpublished}[1]{%
1653 \def\@bspostpublished{#1}%

```

```

1654 }
1655
1656 \newcommand{\presubtitle}[1]{%
1657 \def\@bspresubtitle{\#1}%
1658 }
```

\presubtitle Hook after printing the subtitle.

```

1659 \newcommand{\postsubtitle}[1]{%
1660 \def\@bspostsubtitle{\#1}%
1661 }
```

Initial settings:

```

1662 \if@titlepage
1663 \prepublished{
1664 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1665 \begin{center}}
1666 \postpublished{\par\end{center}\vskip 2em}
1667
1668 \presubtitle{\unskip\begin{center}\unskip}
1669 \postsubtitle{\par\end{center}\vskip 2em}
1670 \else
1671 \prepublished{\begin{center}}
1672 \postpublished{\par\end{center}\vskip 0.5em}
1673
1674 \presubtitle{\begin{center}\unskip}
1675 \postsubtitle{\par\end{center}\vskip 0.5em}
1676 \fi

1677 \end{warpprint}
```

## 42.6 \maketitle for HTML output

An HTML div of class `titlepage` is created, inside of which a L<sup>A</sup>T<sub>E</sub>X PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 35 for other kinds of footnotes.

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

**for HTML output:** 1678 `\begin{warpHTML}`

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

```
1679 \newcommand*{\LWR@maketitlesetup}{%
```

Select which kind of footnote marks to use:

```
1680 \Cbsmarkseries
1681 \Cmpbsmarkseries
```

Redefine the footnote mark:

```
1682 \def\@makefnmark{\textsuperscript{\thefootnote}}%
```

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

Redefine the footnote text:

```
1683 \long\def\@makefntext##1{%
```

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

```
1684 \makethanksmark \LWR@orighspace{1in}
```

```
\makethanksmark \thanksfootmark \tamark \Ctfnmark \itshape a (or similar)
```

Print the text and a closing paragraph tag:

```
1685 ##1\LWR@htmlltagc{\LWR@tagregularparagraph}%
1686 }%
1687 }
```

```
\@fnsymbol {\langle counter\rangle}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being found by pdftotext.

```
1688 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
1689 \mathsection\or \mathparagraph\or \text{\HTMLUnicode{2016}}\or
1690 **\or \dagger\dagger\or \ddagger\ddagger \else\ctrerr\fi}}
```

\maketitle Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
1691 \renewcommand*{\maketitle}{%
```

An HTML titlepage div is used for all classes.

1692 \begin{titlepage}

Set up special patches:

1693 \LWR@maketitlesetup

Typeset the title, etc:

1694 \@maketitle

Immediately generate any \thanks footnotes:

1695 \@thanks

Close the HTML titlepage div:

1696 \end{titlepage}

Reset the footnote counter:

1697 \@bscontmark

1698 }

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

```
1699 \DeclareDocumentCommand{\@maketitle}{}{%
1700   \maketitlehooka
1701   \ifcsempty{@published}{}
1702   {}
1703   {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1704   {\@bspretitle \@title \@bsposttitle}
1705   \ifcsempty{@subtitle}{}
1706   {}
1707   {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1708   \maketitlehookb
1709   {\@bspreauthor \@author \@bspostauthor}
1710   \maketitlehookc
1711   {\@bspredate \@date \@bspostdate}
1712   \maketitlehookd
1713 }

1714 \providecommand{\maketitlehookaa}{}
1715 \providecommand{\maketitlehookaaa}{}
```

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

---

```
1716 \newcommand*{\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

```
1717 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
1718 \@thanks
1719 }
```

`\thanksmarkseries {<series>}`

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

```
1720 \let\LWR@origthanksmarkseries\thanksmarkseries
1721 \renewcommand{\thanksmarkseries}[1]{%
1722 \def\@mpbsmarkseries{%
1723 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}
1724 \LWR@origthanksmarkseries{#1}
1725 }
```

  

```
1726 \end{warpHTML}
```

## 43 Abstract

The following code replaces the L<sup>A</sup>T<sub>E</sub>X default, and will itself be replaced later if the `abstract` package is loaded.

for HTML output: 1727 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the `babel` package.

```
1728 \providecommand*{\abstractname}{Abstract}
```

Env `abstract`

```
1729 \DeclareDocumentEnvironment{abstract}{}{%
1730 {%
1731 \BlockClass{abstract}}
```

---

```

1732 \BlockClassSingle{abstracttitle}{\abstractname}
1733 }
1734 {
1735 \endBlockClass
1736 }

1737 \end{warpHTML}

```

## 44 Quote and verse

### 44.1 Citations and attributions

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

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

**for HTML output:** 1738 `\begin{warpHTML}`  
 1739 `\newcommand{\attribution}[1]{%`  
 1740 `\InlineClass{\attribution}{--\,#1}}% emdash`  
 1741 `\end{warpHTML}`

**for PRINT output:** 1742 `\begin{warpprint}`  
 1743 `\newcommand{\attribution}[1]{\textsc{---\,#1}}`  
 1744 `\end{warpprint}`

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

**for HTML output:** 1745 `\begin{warpHTML}`  
 1746 `\newcommand{\citetitle}[1]{%`  
 1747 `\InlineClass{\citetitle}{--\,#1}}% emdash`  
 1748 `\end{warpHTML}`

**for PRINT output:** 1749 `\begin{warpprint}`  
 1750 `\newcommand{\citetitle}[1]{\textsl{---\,#1}}`  
 1751 `\end{warpprint}`

### 44.2 Quotes, quotations

**for HTML output:** 1752 `\begin{warpHTML}`

Env quote

---

```

1753 \renewenvironment*{quote}
1754 {\LWR@htmlblocktag{blockquote}}
1755 {\LWR@htmlblocktag{/blockquote}}
1756
1757 \renewenvironment*{quotation}
1758 {\LWR@htmlblocktag{blockquotation}}
1759 {\LWR@htmlblocktag{/blockquotation}}


1760 \end{warpHTML}

```

### 44.3 Verse

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

---

```

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

```

---

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

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

**for HTML & PRINT:** 1761 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

**Len \TMLvleftskip** Sets `\vleftskip` inside a `verse` environment in HTML.

```
1762 \newlength{\HTMLvleftskip}
```

---

```
1763 \setlength{\HTMLvleftskip}{1em}
```

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

```
1764 \newlength{\HTMLleftmargini}
1765 \setlength{\HTMLleftmargini}{4.5em}
```

```
1766 \end{warpall}
```

## 45 Verbatim

**for HTML output:** 1767 \begin{warpHTML}

Env `verbatim`

```
1768 \AfterEndPreamble{
1769 \AtBeginEnvironment{verbatim}{\LWR@atbeginverbatim{verbatim}\unskip\vspace*{-\baselineskip}}
1770 \AfterEndEnvironment{verbatim}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
1771 }
```

```
1772 \end{warpHTML}
```

## 46 Fancyvrb

**for HTML & PRINT:** 1773 \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.

```
1774 \newlength{\VerbatimHTMLWidth}
1775 \setlength{\VerbatimHTMLWidth}{4in}
1776 \end{warpall}
```

**for HTML output:** 1777 \begin{warpHTML}

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `VerbatimInput` in the HTML head.

```
1778 \newbool{LWR@verbtags}
1779 \booltrue{LWR@verbtags}
```

For \VerbatimFootnotes:

```
1780 \renewcommand{\VerbatimFootnotes}{  
1781 \PackageError{l warp}{  
1782 {Verbatim footnotes are not yet supported by l warp.}  
1783 {This may be improved when l warp uses LaTeX footnotes instead of pagenotes.}  
1784 }}
```

\LWR@atbeginverbatim {*class*}

Encloses a verbatim environment with the given CSS class.

```
1785 \newcommand*\LWR@atbeginverbatim[1]  
1786 {%
```

Avoid excessive space between lines:

```
1787 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1788 \LWR@stopars%
```

Create a new pre of the given class:

```
1789 \ifbool{\LWR@verbtags}{\LWR@htmlltag{pre class="#1"{})){}}
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the css for the *verse* class.

```
1790 \LWR@origttfamily%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain L<sup>A</sup>T<sub>E</sub>X \hspace functionality:

```
1791 \let\hspace\LWR@orighspace%  
1792 }
```

\LWR@afterendverbatim Finishes enclosing a verbatim environment.

```
1793 \newcommand*\LWR@afterendverbatim{%
```

Remove excess vertical space at the end of the pre:

```
1794 \unskip%
```

At the end of the environment, close the pre:

---

```

1795 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
1796
1797 }{}%

```

Resume regular paragraph handling:

```

1798 \LWR@startpars%
1799 }

```

**\LWR@Verbatimclass** Holds the class of the following verbatim.

```
1800 \newcommand*{\LWR@Verbatimclass}{fancyvrb}
```

**Env VerbatimClass** {*class*} [*Verbatim options*]

Creates a **Verbatim** enclosed in a DIV of the given class.

```

1801 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1802 {%
1803 \renewcommand*{\LWR@Verbatimclass}{#1}%
1804 \LWR@origVerbatim[#2]%
1805 }
1806 {\endVerbatim}

```

After the preamble is loaded, after any patches to **Verbatim**:

```
1807 \AfterEndPreamble{
```

Remember the original defintion of **Verbatim**:

```
1808 \let\LWR@origVerbatim\Verbatim
```

**Env Verbatim** Patched to place the environment in a **fancyvrb** div, and the label in a **fancyvrblabel** div. Also corrects the left margin for line numbers. Also uses **VerbatimHTMLWidth** to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```

1809 \renewcommand*{\Verbatim}{%
1810 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
1811 \LWR@origVerbatim%
1812 }

```

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.

```

1813 \newcommand*{\LWR@fvstartnone}{%
1814 \LWR@traceinfo{fvstartnone}%
1815 \ifbool{\LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}}%
1816 \hbox to\z@{\LWR@atbeginverbatim{verbatim}%
1817 }%
1818 \newcommand*{\LWR@fvendnone}{%
1819 \LWR@traceinfo{fvendnone}%
1820 \hbox to\z@{\LWR@afterendverbatim}%
1821 \ifbool{\LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}}%
1822 }%
1823 \newcommand*{\LWR@fvstartsingle}{%
1824 \LWR@traceinfo{fvstartsingle}%
1825 \LWR@fvstartnone%
1826 \FV@BeginListFrame@Single%
1827 }%
1828 \newcommand*{\LWR@fvendsingle}{%
1829 \LWR@traceinfo{fvendsingle}%
1830 \FV@EndListFrame@Single%
1831 \LWR@fvendnone%
1832 }%
1833 \newcommand*{\LWR@fvstartline}{%
1834 \LWR@traceinfo{fvstartline}%
1835 \LWR@fvstartnone%
1836 \FV@BeginListFrame@Lines%
1837 }%
1838 \newcommand*{\LWR@fvendline}{%
1839 \LWR@traceinfo{fvendline}%
1840 \FV@EndListFrame@Lines%
1841 \LWR@fvendnone%
1842 }%
1843 \newcommand*{\LWR@fvnewline}{%
1844 \LWR@traceinfo{fvnewline}%
1845 \FV@EndListFrame@Lines%
1846 \LWR@fvendnone%
1847 }

```

The following patches select the start/left/right/end behaviors depending on `frame`.  
Original code is from the `fancyvrb` package.

```

1848 \def\FV@Frame@none{%
1849 \let\FV@BeginListFrame\LWR@fvstartnone%
1850 \let\FV@LeftListFrame\relax%
1851 \let\FV@RightListFrame\relax%
1852 \let\FV@EndListFrame\LWR@fvendnone}%
1853 \def\FV@Frame@singl{%
1854 \let\FV@BeginListFrame\LWR@fvstartsingle%
1855 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1856 \let\FV@RightListFrame\FV@RightListFrame@Single%
1857 }

```

```

1858 \let\fv@EndListFrame\lwr@fvendsingle}
1859
1860 \def\fv@Frame@lines{%
1861 \let\fv@BeginListFrame\lwr@fvstartline%
1862 \let\fv@LeftListFrame\relax%
1863 \let\fv@RightListFrame\relax%
1864 \let\fv@EndListFrame\lwr@fvendline}
1865
1866 \def\fv@Frame@topline{%
1867 \let\fv@BeginListFrame\lwr@fvstartline%
1868 \let\fv@LeftListFrame\relax%
1869 \let\fv@RightListFrame\relax%
1870 \let\fv@EndListFrame\lwr@fvendnone}
1871
1872 \def\fv@Frame@bottomline{%
1873 \let\fv@BeginListFrame\lwr@fvstartnone%
1874 \let\fv@LeftListFrame\relax%
1875 \let\fv@RightListFrame\relax%
1876 \let\fv@EndListFrame\lwr@fvendline}
1877
1878 \def\fv@Frame@leftline{%
1879 % To define the \fv@FrameFillLine macro (from \fv@BeginListFrame)
1880 \ifx\fancyverb@fillcolor\relax%
1881 \let\fv@FrameFillLine\relax%
1882 \else%
1883 \tempdima\fv@FrameRule\relax%
1884 \multiply\tempdima-\tw@%
1885 \edef\fv@FrameFillLine{%
1886 {\noexpand\fancyverb@fillcolor{\rule{\width}{\number\tempdima sp}}%
1887 \kern-\number\tempdima sp}%
1888 \fi%
1889 \let\fv@BeginListFrame\lwr@fvstartnone%
1890 \let\fv@LeftListFrame\fvc@LeftListFrame@Single%
1891 \let\fv@RightListFrame\relax%
1892 \let\fv@EndListFrame\lwr@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyverb package.

```

1893 \def\fv@SingleFrameLine#1{%
1894   \hbox to\z@{%
1895 %     \kern\leftmargin
1896     \ifnum#1=\z@\relax
1897       \let\fvc@Label\fvc@LabelBegin
1898     \else
1899       \let\fvc@Label\fvc@LabelEnd
1900     \fi
1901     \ifx\fvc@Label\relax
1902 %       \fancyverb@rulecolor{\rule{\width}{\linewidth \height\fvc@FrameRule}%

```

```

1903      \else
1904          \ifnum#1=\z@
1905 %           \setbox\z@\hbox{\strut\enspace\@LabelBegin\enspace\strut}%
1906           \ifx\@LabelPositionTopLine\relax
1907 \else
1908     \LWR@htmltagc{div class="fancyvrblabel"}
1909 \LWR@origtextrm{\@LabelBegin}%
1910 \LWR@htmltagc{/div}
1911 \fi
1912     \else
1913 %           \setbox\z@\hbox{\strut\enspace\@LabelEnd\enspace\strut}%
1914           \ifx\@LabelPositionBottomLine\relax
1915 \else
1916     \LWR@htmltagc{div class="fancyvrblabel"}
1917 \LWR@origtextrm{\@LabelEnd}
1918 \LWR@htmltagc{/div}
1919 \fi
1920     \fi
1921
1922     \fi
1923     \hss
1924 }
1925 }
```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

1926 \def\@ListProcessLine#1{%
1927     \hbox to \hsize{%
1928 %         \kern\leftmargin
1929         \hbox to \VerbatimHTMLWidth {%
1930             \ifcsvoid{\@LeftListNumber}{}{\kern 2.5em}%
1931 \@LeftListNumber%
1932 %         \FV@LeftListFrame
1933         \FancyVerbFormatLine{#1}%
1934         \hss%
1935 %         \FV@RightListFrame
1936         \FV@RightListNumber%
1937 }%
1938         \hss% required to avoid underfull hboxes
1939 }
1940 }
```

Env `BVerbatim`

```

1941 \AtBeginEnvironment{BVerbatim}
1942 {
1943 \LWR@atbeginverbatim{bverbatim}
1944
```

```

1945 }
1946
1947 \AfterEndEnvironment{BVerbatim}
1948 {
1949 \leavevmode\par\vspace{-\baselineskip}
1950 \LWR@afterendverbatim
1951 }

```

**Env LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```
1952 } % \AfterEndPreamble
```

```
\UseVerbatim {<text>}
```

No changes required.

```
1953 \end{warpHTML}
```

## 47 Theorems

```
\newtheorem {<text>} [<counter>] -or- [<oldname>] {<text>}
```

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a div of class **theorem**.
- The label for each theorem is placed inside a span of class **theoremlabel**.
- The contents are placed inside a div of class **theoremcontents**.

**for HTML output:** 1954 \begin{warpHTML}

```
\@begintheorem {<name>} {<number>}

1955 \renewcommand{\@begintheorem}[2]{%
1956 \LWR@htmldivclass{theorem}
1957 \InlineClass{theoremlabel}{#1\ #2\ }
1958 \LWR@htmldivclass{theoremcontents}
1959 }
```

```
\@opargbegintheorem {⟨name⟩} {⟨number⟩} {⟨oparg⟩}

1960 \renewcommand{\@opargbegintheorem}[3]{%
1961 \BlockClass{theorem}
1962 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
1963 \BlockClass{theoremcontents}
1964 }

\@endtheorem

1965 \renewcommand*{\@endtheorem}{%
1966 \endBlockClass% theoremcontents
1967 \endBlockClass% theorem
1968 }

1969 \end{warpHTML}
```

## 48 Lists

If using `babel` with French, use

```
\frenchbsetup{StandardLists=true}
```

⚠ **French** to preserve the special HTML and `enumitem` list handling.

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

### 48.1 Itemize

**for HTML output:** 1970 \begin{warpHTML}

```
1971 \let\LWR@origitem\item
```

```
\LWR@itemizeitem [⟨label⟩]
```

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
1972 \newcommand*{\LWR@itemizeitem}{%
1973 \LWR@stoppars%
1974 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
```

---

```

1975 \LWR@htmntag{li}%
1976 \LWR@startpars%
1977 \LWR@origitem%
1978 }

```

To have a blank item, use `\mbox{}`. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

---

```

begin{itemize}
item \mbox{}
\begin{itemize}
...

```

Env **itemize** [*enumitem options*]

```

1979 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
1980
1981 \newcommand*{\LWR@itemizestart}{%
1982 \LWR@stoppars%
1983 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
1984 \LWR@htmntag{ul style="list-style-type:none"}{%
1985 \LWR@startpars%
1986 \let\item\LWR@itemizeitem%
1987 }
1988
1989 \AtEndEnvironment{itemize}{\LWR@itemizeend}
1990
1991 \newcommand*{\LWR@itemizeend}{%
1992 \LWR@stoppars%
1993 \LWR@closeprevious{\LWR@depthlistitem}%
1994 \LWR@closeoneprevious{}}%
1995 \LWR@startpars%
1996 }

```

## 48.2 Enumerate

An HTML unordered list is used with customized L<sup>A</sup>T<sub>E</sub>X-generated labels.

Env **enumerate** [*enumitem options*]

```

1997 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
1998
1999 \newcommand*{\LWR@enumeratestart}{%

```

```

2000 \LWR@stoppars%
2001 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}
2002 \LWR@htmlltag{ul style="list-style-type:none"}%
2003 \LWR@startpars%
2004 \let\item\LWR@itemizeitem%
2005 }
2006
2007
2008 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
2009
2010 \newcommand*{\LWR@enumerateend}%
2011 \LWR@stoppars%
2012 \LWR@closeprevious{\LWR@depthlistitem}%
2013 \LWR@closeoneprevious{}%
2014 \LWR@startpars%
2015 }

```

### 48.3 Description

\LWR@descitem [*<label>*] Handles an \item inside a description.

```

2016 \newcommand*{\LWR@descitem}[1] []
2017 {%
2018 \LWR@stoppars%
2019 \LWR@setlatestname{#1}%
2020 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}
2021 \LWR@origitem[]%

```

Be sure the label doesn't print to the left of the rest of the file:

```

2022 \LWR@orighspace{1in}
2023 \LWR@htmlltag{dt}#1\LWR@htmlltag{/dt}%
2024 \LWR@orignewline%
2025 \LWR@htmlltag{dd}%
2026 \LWR@startpars%
2027 }

```

Env description [*enumitem options*]

```

2028 \AtBeginEnvironment{description}{\LWR@descriptionstart}
2029
2030 \newcommand*{\LWR@descriptionstart}%
2031 \LWR@stoppars%
2032 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}
2033 \LWR@htmlltag{dl}%
2034 \LWR@startpars%

```

```

2035 \let\item\LWR@descitem%
2036 }
2037
2038 \AtEndEnvironment{description}{\LWR@descriptionend}
2039
2040 \newcommand*\LWR@descriptionend{%
2041 \LWR@stoppars%
2042 \LWR@closeprevious{\LWR@depthlistitem}%
2043 \LWR@closeoneprevious{}%
2044 \LWR@startpars%
2045 }

\newlist  {\langle name\rangle} {\langle type\rangle} {\langle maxdepth\rangle}

\renewlist {\langle name\rangle} {\langle type\rangle} {\langle maxdepth\rangle}

```

For `enumitem` lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```

2046 \let\LWR@orignewlist\newlist
2047
2048 \renewcommand*\newlist[3]{%
2049 \LWR@orignewlist{\#1}{\#2}{\#3}%
2050 \AtBeginEnvironment{\#1}{\csuse{\LWR@#2start}}%
2051 \AtEndEnvironment{\#1}{\csuse{\LWR@#2end}}%
2052 }

2053 \end{warpHTML}

```

## 49 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

- |              |  |
|--------------|--|
| column types | <ul style="list-style-type: none"> <li>• Vertical rules are not yet supported.</li> <li>• * in a column specification is not used (so far). Repeat the column type the correct number of times.</li> <li>• Only one each of @, !, &gt;, and &lt; may be used at each column, and they are used in that order.</li> <li>• <code>\newcolumntype</code> is ignored; unknown column types are set to 1.</li> </ul> |
|--------------|--|

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

⚠ `\multirow & \multicolumn`

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

⚠ `\multirow`

`\multirow with rules`

`rule at last row`

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

- If a multirow reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.

- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.

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

- `\cmidrule` does not support width or trim options due to CSS limitations.

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

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

`This & That \endhead`

to

`\warpprintonly{This & That \endhead}`

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

⚠ `S columns`

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

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

## 49.1 Token lookahead

Used by `\LWR@futureonospacelet` to look at the next token.

**for HTML output:** 2054 `\begin{warpHTML}`

```
\LWR@mynexttoken
```

```
2055 \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
```

```
2056 \def{\LWR@futurenonspacelet#1{\def{\LWR@cs{#1}}%  
2057 \afterassignment{\LWR@fnalone\let\nexttoken= }  
2058 \def{\LWR@fnalone{\expandafter{\futurelet{\LWR@cs{\LWR@fnltwo}}%  
2059 \def{\LWR@fnltwo{\%  
2060 \expandafter{\ifx{\LWR@cs{\@sptoken}\let\next=\@BTfnlthree}{%  
2061 \else\let\next=\nexttoken\fi\next}  
2062 \def{\@BTfnlthree{\afterassignment{\LWR@fnalone\let\next= }}
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
2063 \newcommand*{\LWR@getmynexttoken}{%  
2064 % nothing must follow this next line  
2065 \LWR@futurenonspacelet{\LWR@mynexttoken\LWR@tabledatacolumntag}  
2066 }
```

## 49.2 Booleans

Bool **LWR@startedrow** True if should print a row tag before this column.

```
2067 \newbool{\LWR@startedrow}  
2068 \boolfalse{\LWR@startedrow}
```

Bool **LWR@doinghline** True if the next row will have an hline above it.

```
2069 \newbool{\LWR@doinghline}  
2070 \boolfalse{\LWR@doinghline}
```

Bool **LWR@doingtbrule** True if the next row will have a top/bottom rule above it.

```
2071 \newbool{\LWR@doingtbrule}  
2072 \boolfalse{\LWR@doingtbrule}
```

Bool **LWR@tableparcell** True if are handling a paragraph inside a table cell, so must close the paragraph

tag before moving on.

```
2073 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
2074 \newbool{LWR@skippingmrowcell}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
2075 \newbool{LWR@intabularmetadata}
```

```
2076 \boolfalse{LWR@intabularmetadata}
```

### 49.3 Handling & and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/>  
where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
2077 \newcommand*\{LWR@closetabledatacell}\{%
2078 \global\booltrue{LWR@intabularmetadata}%
2079 \ifbool{LWR@existingtabular}\{\}%
2080 \%\ not exiting tabular
2081 \ifbool{LWR@skippingmrowcell}\{\}%
2082 \%\ not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
2083 \unskip%
2084 \LWR@getexpparray{LWR@col afterspec}{\theLWR@tablecolspos}%
2085 \LWR@getexpparray{LWR@col atspec}{\theLWR@tablecolspos}%
2086 \LWR@getexpparray{LWR@col bangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
2087 \ifbool{LWR@tableparcell}\{\LWR@stoppars\}\{%
2088 \global\boolfalse{LWR@tableparcell}\%
```

Close the table data cell:

```
2089 \unskip\LWR@htmlltag{/td}\LWR@orignewline%
2090 \%\ not skipping mrowcell
```

```
2091 }% not exiting tabular
2092 \global\boolearn{LWR@skippingmrowcell}%
2093 }
```

LWR@tabulardepth tracks whether & is being used inside a **tabular**.

```
2094 \newcounter{LWR@tabulardepth}
2095 \setcounter{LWR@tabulardepth}{0}
2096
```

When not used inside a **tabular**, & performs its original function as recorded here ( with catcode 4 ).

```
2097 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2098 \begingroup
```

& Will behave depending on whether it is being used inside **tabular**.

& is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

The \catcode allows the & character to be redefined.

```
2099 \catcode`\&=\active
2100
2101 \gdef&{%
2102 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2103 {%
```

If not skipping a multirow cell, close the current data cell.

```
2104 \unskip%
2105 \LWR@closetabledatacell%
```

Move to the next column.

```
2106 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
2107 \LWR@getmynexttoken%
2108 }%
```

If not inside a tabular, performs the original action:

```

2109 {\LWR@origampmacro}%
2110 }%
2111 \endgroup

```

Outside the group, & is left its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded.

\LWR@lwarfStart finally makes & active at the beginning of the HTML conversion.

#### 49.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or \\\*

```
\LWR@tabularendofline
```

```

2112 \NewDocumentCommand{\LWR@tabularendofline}{s o}%
2113 {%
2114 \LWR@closetabledatacell%

```

Finish the previous row:

```

2115 \LWR@htmlltag{/tr}\LWR@orignewline
2116 \global\booltrue{LWR@intabularmetadata}

```

Not yet started a table row:

```
2117 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```

2118 \global\boolfalse{LWR@doinghline}%
2119 \global\boolfalse{LWR@doingtbrule}%
2120 \LWR@clearmidrules%

```

Start at first column:

```
2121 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```

2122 \LWR@getmynexttoken%
2123 }

```

## 49.5 Variables

```
2124 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2125 \newcommand*{\LWR@pposition}{}%
2126 \newcommand*{\LWR@pleft}{}%
2127 \newcommand*{\LWR@pright}{}%
```

`\LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalcols`.

Will contain a string such as `llrrccpc`, exactly one letter per column, without `@`, `>`, `<`, or the vertical pipe.

```
2128 \newcommand*{\LWR@tablecolspec}{}%
```

`\LWR@strresult` Holds the result of `Str` functions.

```
2129 \newcommand*{\LWR@strresult}{}%
```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
2130 \newcommand*{\LWR@origcolspec}{}%
```

Ctr `LWR@tablecolswidth` Holds the width of the table specification.

(This is not the total # columns.)

```
2131 \newcounter{LWR@tablecolswidth}
```

Ctr `LWR@tablecolspos` Where are currently looking into the table column specification.

```
2132 \newcounter{LWR@tablecolspos}
```

Ctr `LWR@tabletotalcols` Holds the final number of table columns.

```
2133 \newcounter{LWR@tabletotalcols}
```

Ctr `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
2134 \newcounter{LWR@tabletotalcolsnext}
```

`LWR@colatspec` A data array of specifications for `@` columns. The leftmost's index is `leftedge`, the others are counter values. See section 23.

`LWR@colbangspec` A data array of specifications for `!` columns. The leftmost's index is `leftedge`, the others are counter values. See section 23.

`LWR@colbeforespec` A data array of specifications for `>` columns.

`LWR@colafterspec` A data array of specifications for `<` columns.

## 49.6 Parsing @, >, <, ! columns

\LWR@parseatcolumn Handles @{text} columns.

```
2135 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2136 \LWR@traceinfo{at column}%
2137 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
2138 \LWR@traceinfo{about to read the next token}%
2139 \expandarg%
2140 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2141 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
2142 \LWR@traceinfo{have now read the next token}%
2143 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}%
2144 {%
  left edge of the table:
  \LWR@traceinfo{at the left edge}%
  \LWR@setexpparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
}%
2148 {%
  not at the left edge:
  \LWR@traceinfo{not at the left edge}%
  \LWR@setexpparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
  \LWR@traceinfo{at \theLWR@tabletotalcols: \LWR@colatspec(\theLWR@tabletotalcols)!}%
}%
2152 }%
2153 \let\LWR@strresult\relax%
2154 \booltrue{LWR@validtablecol}%
2155 }
```

\LWR@parsebangcolumn

```
2156 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
2157 \LWR@traceinfo{bang column}%
2158 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
2159 \LWR@traceinfo{about to read the next token}%
```

```

2160 \expandarg%
2161 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2162 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2163 \LWR@traceinfo{have now read the next token}%
2164 \ifthenelse{\cnttest{\value{\LWR@tabletotalcols}}=0}%
2165 {%
2166   \LWR@traceinfo{at the left edge}%
2167   \LWR@setexpparray{\LWR@colbangspec}{leftedge}{\LWR@strresult}%
2168 }%
2169 {%
2170   \LWR@traceinfo{not at the left edge}%
2171   \LWR@setexpparray{\LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2172   \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%
2173 }%
2174 \let\LWR@strresult\relax%
2175 \booltrue{\LWR@validtablecol}%
2176 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```
2177 \newcommand*\LWR@parsebeforecolumn{%
```

Move to the next token after the '>':

```
2178 \addtocounter{\LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```

2179 \expandarg%
2180 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2181 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2182 \LWR@setexpparray{\LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2183 \let\LWR@strresult\relax%
2184 \booltrue{\LWR@validtablecol}%
2185 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```
2186 \newcommand*\LWR@parseaftercolumn{%
```

Move to the next token after the '<':

---

```
2187 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
2188 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
2189 \expandarg%
2190 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2191 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
2192 \LWR@setexpparray{LWR@colaferspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2193 \let\LWR@strresult\relax%
2194 \booltrue{LWR@validtablecol}%
2195 }
```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```
2196 \newcommand*{\LWR@parseskipcolumn}{%
2197 \booltrue{LWR@validtablecol}%
2198 }
```

## 49.7 Parsing ‘l’, ‘c’, or ‘r’ columns

`\LWR@parsingnormalcolumn`  $\{\langle thiscolumn \rangle\}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
2199 \newcommand*{\LWR@parsingnormalcolumn}[1]{%
2200 \appto{\LWR@tablecolspec}{#1}%
2201 \addtocounter{LWR@tabletotalcols}{1}%
2202 \addtocounter{LWR@tabletotalcolsnext}{1}%
2203 \LWR@setexpparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%
2204 \LWR@setexpparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
2205 \LWR@setexpparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\relax}%
2206 \LWR@setexpparray{LWR@colaferspec}{\theLWR@tabletotalcolsnext}{\relax}%
2207 \booltrue{LWR@validtablecol}%
2208 }
```

## 49.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn`  $\{\langle thiscolumn \rangle\}$  The width will be ignored.

---

```
2209 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
2210 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```
2211 \addtocounter{\LWR@tablecolspos}{1} %
```

```
2212 }
```

## 49.9 Parsing ‘D’ columns

From the `dcolumn` package.

```
\LWR@parseDcolumn {<thiscolumn>} The three parameters will be ignored.
```

```
2213 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
2214 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
2215 \addtocounter{\LWR@tablecolspos}{3} %
```

```
2216 }
```

## 49.10 Parsing the column specifications



HTML CSS cannot exactly match the L<sup>A</sup>T<sub>E</sub>X concept of a baseline for a table row. Table 6 shows the L<sup>A</sup>T<sub>E</sub>X results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the `p` column specification in table 7 for details.

Table 7 describes how each kind of column is converted to HTML.

```
Bool LWR@validtablecol True if found a valid table column type.
```

```
2217 \newbool{\LWR@validtablecol}
```

```
\LWR@parsetablecols {<colspecs>}
```

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 7: Tabular HTML column conversions

**l, r, c:** Converted to table cells without paragraph tags.

Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

**p:** Converted to table cells with paragraph tags. Ref: Table 6, L<sup>A</sup>T<sub>E</sub>X places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate L<sup>A</sup>T<sub>E</sub>X baselines.

**m:** With paragraph tags, CSS `vertical-align:middle`.

**b:** With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

**P, M, B:** Horizontally-centered versions.

**S:** Converted to 'r'. From the `siunitx` package.

**D:** Converted to 'c'. From the `dcolumn` package.

**@, !, >, <:** One each, in that order.

**Unknown:** Converted to 'l'.

**\newcolumn:** Currently treated as unknown.

Scans the column specification left to right.

Builds `\LWR@tablecols` with the final specification, one column per entry. The number of final columns is stored in `\LWR@tabletotalcols`.

```
2218 \newcommand*{\LWR@parsetablecols}[1]{%
2219 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
2220 \renewcommand*{\LWR@origcols}{#1}%
```

Clear the parsed resulting column spec:

```
2221 \renewcommand*{\LWR@tablecols}{()}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
2222 \setcounter{\LWR@tabletotalcols}{0}%
2223 \setcounter{\LWR@tabletotalcolsnext}{1}%
2224 \LWR@setexpparray{\LWR@colatsspec}{leftedge}{\relax}%
2225 \LWR@setexpparray{\LWR@colatsspec}{1}{\relax}%
2226 \LWR@setexpparray{\LWR@colatsspec}{2}{\relax}%
2227 \LWR@setexpparray{\LWR@colatsspec}{3}{\relax}%
2228 \LWR@setexpparray{\LWR@colbangspec}{leftedge}{\relax}%
2229 \LWR@setexpparray{\LWR@colbangspec}{1}{\relax}%
2230 \LWR@setexpparray{\LWR@colbangspec}{2}{\relax}%
2231 \LWR@setexpparray{\LWR@colbangspec}{3}{\relax}%
2232 \LWR@setexpparray{\LWR@colbeforespec}{1}{\relax}%
2233 \LWR@setexpparray{\LWR@colbeforespec}{2}{\relax}%
2234 \LWR@setexpparray{\LWR@colbeforespec}{3}{\relax}%
2235 \LWR@setexpparray{\LWR@colafterspec}{1}{\relax}%
2236 \LWR@setexpparray{\LWR@colafterspec}{2}{\relax}%
2237 \LWR@setexpparray{\LWR@colafterspec}{3}{\relax}%

```

Starting at the first column specification:

```
2238 \setcounter{\LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
2239 \LWR@traceinfo{about to StrLen}%
2240 \noexpandarg%
2241 \StrLen{\#1}[\LWR@strresult]%
2242 \fullexpandarg%
2243 \LWR@traceinfo{finished StrLen}%
2244 \setcounter{\LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
2245 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolswidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
2246 \noexpandarg%
2247 \StrChar{\#1}{\theLWR@tablecolspos}[\LWR@strresult]%
2248 \fullexpandarg%
```

Not yet found a valid column type

```
2249 \boolfalse{LWR@validtablecol}%
```

Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
2250 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
2251 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
2252 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
2253 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
2254 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{%
2255 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{%
2256 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{%
2257 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{%
2258 \IfStrEq{\LWR@strresult}{|}{\LWR@parsingskipcolumn}{%
2259 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
2260 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
2261 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%
```

From the dcolumn package:

```
2262 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
2263 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%
```

---

Many people define centered versions “P”, “M”, and “B”:  
`\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}`

---

```
2264 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{%
2265 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{%
2266 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{%
```

If this column was an invalid column type, convert it to a `p` column:

```
2267 \ifbool{LWR@validtablecol}{}{%
2268 \LWR@parsenormalcolumn{1}%
2269 }%
2270 \addtocounter{LWR@tablecolspos}{1}%
2271 }%
2272 }%
```

## 49.11 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for `hline` or `tbrule` if necessary.

```
2273 \newcommand*{\LWR@maybenewtablerow}%
2274 {%
2275 \ifbool{LWR@startedrow}%
2276 {}% started the row
2277 {}% not started the row
```

Remember that now have started the row:

```
2278 \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
2279 \global\booltrue{LWR@intabularmetadata}%
2280 \ifbool{LWR@doinghline}%
2281 {\LWR@htmlltag{tr class="hline"{} }\LWR@orignewline}%
2282 {}% not doing hline
2283 \ifbool{LWR@doingtbrule}%
2284 {\LWR@htmlltag{tr class="tbrule"{} }\LWR@orignewline}%
2285 {\LWR@htmlltag{tr}\LWR@orignewline}%
2286 }% end of not doing hline
2287 }% end of not started the row
2288 }
```

## 49.12 Data opening tag

`\LWR@tabledatasinglecolumntag` Print a table data opening tag with style for alignment

```
2289 \newcommand*{\LWR@tabledatasinglecolumntag}%
2290 {%
2291 \LWR@maybenewtablerow%
```

If have found the end of tabular command, do not create the next data cell:

```
2292 \ifbool{LWR@exittingtabular}{}%
2293 { % not exiting tabular
```

Fetch the current column's alignment character into `\LWR@strresult`:

```
2294 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```
2295 \LWR@htmlltag{td class="td%"
```

append this column's spec:

```
2296 \LWR@strresult%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
2297 \ifthenelse{\equal{\LWR@getexpparray{LWR@midrules}}{\theLWR@tablecolspos}}{Y}}{rule}{}%
2298 " {}%
```

If this is a p, m, b, or X column, allow paragraphs:

```
2299 \ifthenelse{%
2300 \equal{\LWR@strresult}{p}\OR%
2301 \equal{\LWR@strresult}{m}\OR%
2302 \equal{\LWR@strresult}{b}\OR%
2303 \equal{\LWR@strresult}{P}\OR%
2304 \equal{\LWR@strresult}{M}\OR%
2305 \equal{\LWR@strresult}{B}\OR%
2306 \equal{\LWR@strresult}{X}%
2307 }%
2308 { % allow pars
2309 \LWR@startpars%
2310 \global\booltrue{LWR@tableparcell}%
2311 }% allow pars
2312 {}% no pars
```

Print the @ and ! contents before first column, and then the > contents:

```
2313 \ifthenelse{\cניתest{\value{LWR@tablecolspos}}=1}%
2314 {%
2315 \LWR@getexpparray{LWR@colatspec}{leftedge}%
2316 \LWR@getexpparray{LWR@colbangspec}{leftedge}%
2317 }% left edge
2318 {}% not left edge
2319 \LWR@getexpparray{LWR@colbeforespec}{\theLWR@tablecolspos}%
2320 \global\boolfalse{LWR@intabularmetadata}%
```

---

```
2321 }% not exiting tabular
2322 }%
```

### 49.13 Midrules

**LWR@midrules** LWR@midrules is a data array (section 23) of columns containing Y if a midrule should be created for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules data array.

```
2323 \newcounter{LWR@midrulecounter}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \\.

Clears all LWR@midrules markers for this line.

```
2324 \newcommand*{\LWR@clearmidrules}
2325 {%
2326 \setcounter{LWR@midrulecounter}{1}%
2327 \whiledo{%
2328 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolswidth}}}%
2329 }%
2330 {%
2331 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
2332 \addtocounter{LWR@midrulecounter}{1}%
2333 }%
2334 }
```

\LWR@subcmidrule [*width*] {[*trim*] }{*leftcolumn*} {*rightcolumn*}

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```
2335 \newcommand*{\LWR@subcmidrule}[4]{%
2336 \setcounter{LWR@midrulecounter}{#3}%
2337 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}{%
2338 {%
2339 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2340 \addtocounter{LWR@midrulecounter}{1}%
2341 }% end of the whiledo
2342 }
```

\LWR@docmidrule [*width*] {[*trim*] }{*leftcolumn-rightcolumn*}

Marks LWR@midrules array elements to be “Y” from left to right columns.

---

```
2343 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
2344 {\LWR@subcmidrule{\#1}{\#2}\#3}
```

## 49.14 Multicolumns

### 49.14.1 Parsing multicolumns

```
2345 \newcounter{LWR@tablemulticolswidth}
2346 \newcounter{LWR@tablemulticolspos}
```

`\LWR@printmccoltype {<colspec>}` Print any valid column type found. Does not print @, >, or < columns or their associated tokens.

This is printed as part of the table data tag's `class`.

```
2347 \newcommand*{\LWR@printmccoltype}[1]{%
2348 \LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
2349 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
2350 \IfStrEq{\LWR@strresult}{l}{\%}
2351 \IfStrEq{\LWR@strresult}{c}{\%}
2352 \IfStrEq{\LWR@strresult}{r}{\%}
2353 \IfStrEq{\LWR@strresult}{p}{\%}
2354 \IfStrEq{\LWR@strresult}{m}{\%}
2355 \IfStrEq{\LWR@strresult}{b}{\%}
2356 \IfStrEq{\LWR@strresult}{P}{\%}
2357 \IfStrEq{\LWR@strresult}{M}{\%}
2358 \IfStrEq{\LWR@strresult}{B}{\%}
2359 \IfStrEq{\LWR@strresult}{S}{\%}
2360 \IfStrEq{\LWR@strresult}{X}{\%}
2361 \LWR@traceinfo{lwr@printmccoltype done}%
2362 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
2363 \newcommand*{\LWR@multicolpartext}{%
2364 \LWR@startpars%
2365 \LWR@multicoltext%
2366 \LWR@stoppars%
2367 }
```

\LWR@multicolother {*colspec*} For @, >, <, print the next token without paragraph tags:

```
2368 \newcommand*{\LWR@multicolother}[1]{%
2369 \addtocounter{LWR@tablemulticolspos}{1}%
2370 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
2371 \LWR@strresult%
```

A valid column data type was found:

```
2372 \booltrue{LWR@validtablecol}%
2373 }
```

\LWR@multicolskip Nothing to print for this column type.

```
2374 \newcommand*{\LWR@multicolskip}{}%
```

A valid column data type was found:

```
2375 \booltrue{LWR@validtablecol}%
2376 }
```

\LWR@printmccoldata {*colspec*} Print the data for any valid column type found.

```
2377 \newcommand*{\LWR@printmccoldata}[1]{%
2378 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
2379 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
2380 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
2381 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
2382 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
2383 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
2384 \IfStrEq{\LWR@strresult}{D}{}%
2385 \addtocounter{LWR@tablemulticolspos}{3}%
2386 \LWR@multicoltext%
2387 }{}%
2388 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
2389 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
2390 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
```

```

2391 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}
2392 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}
2393 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}
2394 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}
2395 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}
2396 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}
2397 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}
2398 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}
2399 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}
2400 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}

```

If an invalid column type:

```
2401 \ifbool{LWR@validtablecol}{}{\LWR@multicoltext}{}
```

Tracing:

```

2402 \LWR@traceinfo{lwr@printmccoldata done}
2403 }
```

`\parsemulticolumnalignment {<1: colspec>} {<2: printresults>}`

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a p{spec} column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

2404 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2405 \setcounter{LWR@tablemulticolspos}{1}%
2406 \StrLen{\#1}[\LWR@strresult]%
2407 \setcounter{LWR@tablemulticolswidth}{\LWR@strresult}}
```

Scan across the tokens in the column spec:

```

2408 \whiledo{%
2409 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolswidth}%
2410 }%
2411 {%
```

Execute the assigned print function for each token in the column spec:

```
2412 #2{\#1}%
```

Move to the next token in the column spec:

```

2413 \addtocounter{LWR@tablemulticolspos}{1}%
2414 }%
2415 }
```

#### 49.14.2 High-level multicolumn interface

```
\LWR@domulticolumn {<1: numcols>} {<2: colspec>} {<3: text>}

2416 \newcommand{\LWR@multicoltext}{}%
2417
2418 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2419 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
2420 \renewcommand{\LWR@multicoltext}{%
2421 #3%
2422 \booltrue{LWR@validtablecol}%
2423 }%
```

Row processing:

```
2424 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
2425 \LWR@htmlltag{td colspan="#1"
2426 class="td%"}
```

Print the column type:

```
2427 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoltype}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule”.

```
2428 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
```

Close the class tag’s opening quote:

```
2429 "%
2430 }% end of the opening table data tag
2431 \global\boolearn{LWR@intabularmetadata}%
2432 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2433 }
```

#### 49.14.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption pacakge, step the counter if `longtable*`.

```

2434 \newbool{LWR@starredlongtable}
2435 \boolfalse{LWR@starredlongtable}

Per the caption package. User-redefinable float type.

2436 \providecommand*\LTcaptive{table}

\LWR@longtabledatacaptiontag * [<toc entry>] {<caption>}

2437 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2438 {%

```

Remember the latest name for \nameref:

```

2439 \IfValueTF{#2}{% optional given?
2440 \ifthenelse{\equal{#2}{}}% optional empty?
2441 {\LWR@setlatestname{#3}}% empty
2442 {\LWR@setlatestname{#2}}% given and non-empty
2443 }% optional given
2444 {\LWR@setlatestname{#3}}% no optional

```

create a multicolumn across all the columns

```

2445 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
2446 % \IfBooleanTF{#1}{ star?
2447 % {\IfValueTF{#2}{\LWR@origcaption*[#2]{#3}}{\LWR@origcaption*[#3]}}
2448 % {\IfValueTF{#2}{\LWR@origcaption[#2]{#3}}{\LWR@origcaption[#3]}}
2449 \IfBooleanTF{#1}{ star?

```

Star version, show a caption but do not make a LOT entry:

```

2450 {%
2451 \LWR@htmlblocktag{figcaption}%
2452 #3%
2453 \LWR@htmlblocktag{/figcaption}%
2454 }%
2455 {%

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

2456 \ifbool{LWR@starredlongtable}{%
2457 {%
2458 \ifthenelse{\equal{#2}{}}{ TOC entry
2459 {}%
2460 {}%
2461 \refstepcounter{\LTcaptive}%

```

```

2462 \protected@edef\currentlabel{%
2463 \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
2464 }%
2465 }{}}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

2466 \LWR@htmlblocktag{figcaption}%
2467 \csuse{fnum@\LTcaptype}\CaptionSeparator#3%
2468 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```
2469 \ifthenelse{\equal{#2}{}}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
2470 }%
```

If the optional caption was given, but might only be []:

```

2471 {%
2472 \IfNoValueTF{#2}{%
2473 {%
2474 \addcontentsline{%
2475 \csuse{ext@\LTcaptype}}{%
2476 \LTcaptype}}%
2477 }%
2478 \protect\numberline{%
2479 \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
2480 {\ignorespaces #3\protect\relax}%
2481 }%
2482 }% end of No TOC entry

```

The optional caption is []:

```

2473 {%
2474 \addcontentsline{%
2475 \csuse{ext@\LTcaptype}}{%
2476 \LTcaptype}}%
2477 }%
2478 \protect\numberline{%
2479 \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
2480 {\ignorespaces #3\protect\relax}%
2481 }%
2482 }% end of No TOC entry

```

The optional caption has text enclosed:

```

2483 {%
2484 \addcontentsline{%
2485 \csuse{ext@\LTcaptype}}{%
2486 \LTcaptype}}%
2487 }%
2488 \protect\numberline{%
2489 \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
2490 {\ignorespaces #2\protect\relax}%
2491 }%
2492 }% end of yes TOC entry

```

```

2493 }% end of TOC entry not empty
2494 }% end of no star
2495 }% end of \LWR@domulticolumn
2496
2497 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
2498 \addtocounter{LWR@tablecolspos}{-1}
2499
2500 }

```

#### 49.14.4 \tabledatamulticolumntag

```

\LWR@tabledatamulticolumntag  {\langle numcols\rangle} {\langle alignment\rangle} {\langle text\rangle}

2501 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}{%
2502 {%
2503 \LWR@domulticolumn{#1}{#2}{#3}{%
2504 \addtocounter{LWR@tablecolspos}{#1}{%
2505 \addtocounter{LWR@tablecolspos}{-1}{%
2506 }

```

## 49.15 Multirow

Pkg **multirow**

```

\LWR@tabledatamultirowtag  {\langle numrows\rangle} [\langle bigstruts\rangle] {\langle width\rangle} [\langle fixup\rangle] {\langle text\rangle}

2507 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}{%
2508 {%
2509 \LWR@maybenewtablerow{%

```

Print the start of a new table data cell:

```
2510 \LWR@htmlltag{td rowspan="#1" class="td{%
```

Append this column's spec:

```
2511 \StrChar{\LWR@tablecolspe}{\theLWR@tablecolspos}{%
```

If this column has a cmidrule, add “rule” to the end of the HTML class:

```
2512 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}}{\theLWR@tablecolspos}{Y}}{\rule}{}
2513 "[]"}
```

While printing the text, redefine \\ to generate a new line

---

```

2514 \begingroup \let\\LWR@endofline #5 \endgroup
2515 \LWR@stoppars%
2516 \global\boolfalse{LWR@intabularmetadata}%
2517 }%

```

## 49.16 Utility macros inside a table

```

2518 \newcommand*{\LWR@donothing}{}
2519 \newcommand*{\LWR@domidrule}{\booltrue{LWR@doinghline}}
2520 \newcommand*{\LWR@dotbrule}{\booltrue{LWR@doingtbrule}}

```

## 49.17 Checking for a new table cell

\LWR@tabledatacolumnntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```

2521 \newbool{LWR@exittingtabular}
2522 \newcommand*{\LWR@tabledatacolumnntag}{%
2523 {%

```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
2524 \let\mynext\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```

2525 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2526 {\booltrue{LWR@exittingtabular}}{}}%

```

`longtable` can have a caption in a cell

```

2527 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2528 {\let\mynext\LWR@donothing}{}}%

```

Look for other things which would not start a table cell:

```

2529 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2530 {\let\mynext\LWR@donothing}{}}%
2531 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2532 {\let\mynext\LWR@donothing}{}}%

```

if come to an `\mrowcell`, this is a cell to be skipped over

```

2533 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2534 {\let\mynext\LWR@donothing}{}}%
2535 %
2536 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2537 {\let\mynext\LWR@donothing}{}}%
2538 %
2539 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2540 {\let\mynext\LWR@donothing}{}}%
2541 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2542 {\let\mynext\LWR@donothing}{}}%
2543 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2544 {\let\mynext\LWR@donothing}{}}%
2545 %
2546 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2547 {\let\mynext\LWR@donothing}{}}%
2548 %
2549 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2550 {\let\mynext\LWR@donothing}{}}%
2551 %
2552 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprionly}}{%
2553 {\let\mynext\LWR@donothing}{}}%
2554 %
2555 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}{%
2556 {\let\mynext\LWR@donothing}{}}%
2557 %
2558 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}{%
2559 {\let\mynext\LWR@donothing}{}}%

```

no action for an `\end` token

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:

```

2560 \mynext%
2561 }

2562 \end{warpHTML}

```

## 49.18 \mrowcell

`\mrowcell` The user must insert `\mrowcell` into any multirow cells which must be skipped.  
⚠ This command has no action during print output.

**for HTML & PRINT:** 2563 `\begin{warpall}`  
2564 `\newcommand*{\mrowcell}{}{}`  
2565 `\end{warpall}`

## 49.19 New \tabular definition

for HTML output: 2566 \begin{warpHTML}

Env LWR@tabular [⟨vertposition⟩] {⟨colspecs⟩}

The new tabular environment will be \let in \LWR@LwarpStart, since siunitx might redefine tabular in the user's document.

```
2567 \newenvironment*{\LWR@tabular}[2] []
2568 {%
2569 \LWR@traceinfo{tabular started}%
2570 \begingroup%
2571 \addtocounter{\LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
2572 \global\boolfalse{\LWR@startedrow}%
```

Not yet doing an hline:

```
2573 \global\boolfalse{\LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2574 \global\boolfalse{\LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2575 \boolfalse{\LWR@exittingtabular}%
```

Create the `table` tag:

```
2576 \global\booltrue{\LWR@intabularmetadata}%
2577 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2578 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
2579 \LWR@stopars%
```

Track column # for setting text-align:

```
2580 \setcounter{\LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2581 \LWR@clearmidrules%
```

\\" becomes a macro to end the table row:

```
2582 \let\\LWR@tabularendofline%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

```
2583 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
2584 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
2585 \NewDocumentCommand{\cmidrule}{o d() m}%
2586 {\LWR@docmidrule[##1](##2)##3}\LWR@getmynexttoken}%
2587 \RenewDocumentCommand{\cline}{m}%
2588 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2589 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2590 \newcommand*{\bottomrule}{\LWR@dotbrule\LWR@getmynexttoken}%
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use \LWR@getmynexttoken.

```
2591 \let\multicolumn\LWR@tabledatamulticolumntag%
2592 \let\multirow\LWR@tabledatamultirowtag%
2593 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
2594 \let\caption\LWR@longtabledatacaptiontag%
```

Reset for new processing:

```
2595 \global\boolearnfalse{\LWR@tableparcell}%
2596 \global\boolearnfalse{\LWR@skippingmrowcell}%
```

Look ahead for a possible table data cell:

```
2597 \LWR@getmynexttoken%
2598 }%
```

Ending the environment:

```
2599 {%
2600 \LWR@closetabledatacell%
2601 \LWR@htmlblocktag{/tr}%
2602 \LWR@htmlblocktag{/table}%
2603 \global\boolearnfalse{\LWR@intabularmetadata}%

```

---

```

2604 \addtocounter{LWR@tabulardepth}{-1}%
2605 \endgroup%
2606 }

2607 \end{warpHTML}

```

## 49.20 Array

Pkg array

array is also automatically loaded by siunitx.

# 50 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 8 shows the data structures related to cross-referencing.

**for HTML output:** 2608 \begin{warpHTML}

## 50.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
2609 \newcommand*\{@currentlabelname}{}%
```

\LWR@stripperiod {\langle text \rangle} [⟨.⟩]

Removes a trailing period.

```
2610 \def\LWR@stripperiod#1.\ltx@empty#2@nil{#1}%
```

\LWR@setlatestname {\langle object name \rangle}

Removes \label, strips any final period, and remembers the result.

```
2611 \newcommand*\LWR@setlatestname}[1]{%
```

Table 8: Cross-referencing data structures

<b>Original L<sup>A</sup>T<sub>E</sub>X:</b>	(print and HTML)
<b>\refstepcounter:</b> Steps the counter and sets \currentlabel.	
<b>\currentlabel:</b> \p@<ctr>\the<ctr> Updated by \refstepcounter.	
<b>\label:</b> Writes to the .aux file: \newlabel{<label>}{{\currentlabel}{\thepage}}	
<b>\newlabel:</b> When the .aux file is read, sets \r@<label>.	
<b>\r@&lt;label&gt;:</b> Set to: {{\currentlabel}{\thepage}}	
<b>\ref:</b> Returns the first part of \r@<label>.	
<b>\pageref:</b> Returns the second part of \r@<label>.	
<b>Added by amsmath:</b>	(print and HTML)
<b>\label:</b> Delayed until math is completed.	
<b>Added by l warp:</b>	(HTML only)
<b>\nameref:</b> Emulated from hyperref for l warp. See section 50.4.	
<b>\ref and \nameref:</b> Adds HTML tags. See section 50.4.	
<b>Added by cleverref:</b>	(print and HTML)
<b>\refstepcounter:</b> Added: sets \cref@currentlabel.	
<b>\cref@currentlabel:</b> (<type>=<ctr> unless an alias is used): [<type>] [\\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>}	
Also see section 35.4 for use with footnotes.	
<b>\label:</b> Writes to the .aux file: \newlabel{<label>@\cref}{{\cref@currentlabel}{\thepage}}	
<b>\newlabel:</b> (Unchanged.) When the .aux file is read, sets \r@<label>@\cref.	
<b>\r@&lt;label&gt;@\cref:</b> Set to: {{\cref@currentlabel}{\thepage}}	
<b>Utility functions:</b> See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
<b>Cross-referencing names:</b> \crefname and \Crefname assign human-readable names for references to this counter type.	
<b>Additionally patched by l warp:</b>	(HTML only)
<b>\cref, etc.:</b> Modified for l warp. See section 59.	
<b>\label:</b> Adds HTML tags (section 50.3), plus \splabel data (section 50.2):	
<b>zLWR@name:</b> The section name for this label.	
<b>zLWR@htmlfilenumber:</b> The filenumber or name for this label.	
<b>zLWR@lateximagedepth:</b> The lateximagedepth for this label.	
<b>zLWR@lateximagenumber:</b> The lateximagenumber for this label.	
<b>\label inside math:</b> See section 54.4.1.	
<b>Footnotes:</b> See \noteentry in section 35.4.	

Remove \label and other commands from the name, the strip any final period.  
See zref-titleref and gettitlestring.

```
2612 \GetTitleStringExpand{#1}%
2613 \edef@\currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
2614 \edef@\currentlabelname{%
2615 \expandafter\LWR@stripperiod@\currentlabelname%
2616 \ltx@empty.\ltx@empty@nil%
2617 }%
2618 }
```

## 50.2 Zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called special:

```
2619 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2620 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2621 \zref@newprop{zLWR@htmlfilename}{%
2622 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilename}%
2623 }%
```

Additional properties for lateximages:

```
2624 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2625 \zref@newprop{zLWR@lateximageindex}{\arabic{LWR@lateximageindex}}
```

zLWR@htmlfilename property holds the file number or name

Add a LWR@htmlfilename property, and lateximage properties to special:

```
2626 \zref@addprop{special}{zLWR@name}
2627 \zref@addprop{special}{zLWR@htmlfilename}
2628 \zref@addprop{special}{zLWR@lateximagedepth}
2629 \zref@addprop{special}{zLWR@lateximageindex}
```

Returns the selected field:

---

```
2630 \newcommand*{\LWR@spref}[2]{%
2631 \zref@extractdefault{#1}{#2}{??}}
```

**\LWR@nameref** {*label*} Returns the section name for this label:

```
2632 \newcommand*{\LWR@nameref}[1]{%
2633 \LWR@spref{#1}{zLWR@name}}%
2634 }
```

**\LWR@htmlfileref** {*label*} Returns the file number for this label:

```
2635 \newcommand*{\LWR@htmlfileref}[1]{%
2636 \LWR@traceinfo{\LWR@htmlfileref A: !#1!}%
2637 \LWR@spref{#1}{zLWR@htmlfilenumber}}%
2638 \LWR@traceinfo{\LWR@htmlfileref B}%
2639 }
```

**\LWR@lateximagedepthref** {*label*} Returns the `lateximagedepth` for this label:

```
2640 \newcommand*{\LWR@lateximagedepthref}[1]{%
2641 \LWR@spref{#1}{zLWR@lateximagedepth}}%
2642 }
```

**\LWR@lateximagenumberref** {*label*} Returns the `lateximagenumber` for this label:

```
2643 \newcommand*{\LWR@lateximagenumberref}[1]{%
2644 \LWR@spref{#1}{zLWR@lateximagenumber}}%
2645 }
```

**\LWR@splabel** {*label*} Sanitize the name and then creates the label:

```
2646 \newcommand*{\LWR@splabel}[1]{%
2647 \LWR@setlatestname{@currentlabelname}}%
2648 \zref@labelbylist{#1}{special}}
```

### 50.3 Labels

**\LWR@subsublabel** {*label*} Creates an HTML id tag.

```
2649 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a `lateximage`, since it would appear in the image:

---

```

2650 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
2651 {}%
2652 {%
  % not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

2653 \ifbool{LWR@doingstartpars}{%
2654   {%
2655     \ifbool{LWR@doingapar}{%
2656       {%
2657         \LWR@htmlltag{a id="#1"}\LWR@htmlltag{/a}}%
2658       }%
2659       {%
2660         \LWR@stoppars%
2661         \LWR@htmlltag{a id="#1"}\LWR@htmlltag{/a}}%
2662       \LWR@startpars%
2663     }%
2664     {%
2665       pars allowed
2666       \LWR@htmlltag{a id="#1"}\LWR@htmlltag{/a}}%
2667     }%
2668   }%
2669 }

```

`\LWR@sublabel {<label>}`

`\label` during HTML output. Creates a traditional L<sup>A</sup>T<sub>E</sub>X label, adds a special label with additional information, and creates an HTML id tag.

Used by `\LWR@htmllabel` and `\LWR@htmlmathlabel`.

```

2670 \newcommand*{\LWR@sublabel}[1]{%

```

Create a traditional LaTeX label, as modified by `cleveref`:

```

2671 \LWR@traceinfo{\LWR@sublabel: !#1!}%
2672 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenum`:

```

2673 \LWR@traceinfo{about to splabel}%
2674 \LWR@traceinfo{filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2675 \LWR@traceinfo{LWR@thisfilename is !\LWR@thisfilename!}%
2676 \LWR@traceinfo{LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
2677 \LWR@splabel{#1}%
2678 \LWR@subsublabel{#1}%
2679 \LWR@traceinfo{finished with LWR@sublabel}%
2680 }

```

```
\LWR@newlabel {⟨label⟩}
```

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular L<sup>A</sup>T<sub>E</sub>X `\label`.

```
2681 \newcommand*{\LWR@newlabel}[1]{%
2682 \LWR@traceinfo{lwr@newlabel: !#1!}%
2683 % \@bsphack%
2684 \LWR@sublabel{#1}%
2685 % \@esphack%
2686 \LWR@traceinfo{lwr@newlabel: done}%
2687 }
```

## 50.4 References

```
\LWR@startref {⟨label⟩} (Common code for \ref and \nameref.)
```

Open an HTML tag reference to a filename, # character, and a label.

```
2688 \newcommand*{\LWR@startref}[1]%
2689 {%
2690 \edef\lWR@lidref{\LWR@lateximagedepthref{#1}}%
2691 \LWR@traceinfo{***\LWR@startref A: !#1!}%
```

Create the tag:

```
2692 \LWR@htmlltag{a href="#"%
2693 \LWR@traceinfo{***\LWR@startref B}%
2694 \LWR@htmllrefsectionfilename{#1}%
2695 \LWR@traceinfo{***\LWR@startref C}%
2696 \#}%
```

See if `\LWR@lateximagedepth` is unknown:

```
2697 \LWR@traceinfo{***\LWR@startref D: !#1!}%
2698 \ifthenelse{\equal{\lWR@lidref}{??}}{%
```

“??” if `\LWR@lateximagedepth` is unknown:

```
2699 {??}%
```

If `\LWR@lateximagedepth` is known:

```
2700 {%
2701 \LWR@traceinfo{***\LWR@startref D1}%
2702 \LWR@traceinfo{***depthref is \lWR@lidref}%
```

```

2703 \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}{%
2704 {%
2705 \LWR@traceinfo{***\LWR@startref D2}%
2706 lateximage\LWR@lateximagenumberref{#1}%
2707 }%
2708 {%
2709 \LWR@traceinfo{***\LWR@startref D3}%
2710 #1%
2711 }%
2712 }%
2713 \LWR@traceinfo{***\LWR@startref E}%

```

closing quote:

```

2714 "{}"%
2715 \LWR@traceinfo{***\LWR@startref F}%
2716 }

```

**\ref** \* {\<label>} Create an internal document reference link, or without a link if starred per hyperref.

```

2717 \NewDocumentCommand{\LWR@newref}{s m}{%
2718 \LWR@traceinfo{\LWR@newref #2}%
2719 \IfBooleanTF{#1}{%
2720 {\LWR@origref{#2}}}{%
2721 {%
2722 \LWR@startref{#2}%
2723 \LWR@origref{#2}%
2724 \LWR@htmlltag{/a}}%
2725 }%
2726 }%
2727

```

**\pageref** *PageFor* Text for starred page references.

```
2728 \newcommand*{\pageref}{\see }
```

**\pageref** \* {\<label>} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```

2729 \NewDocumentCommand{\LWR@newpageref}{s m}{%
2730 \IfBooleanTF{#1}{%
2731 {(\pageref{\LWR@origref{#2}})}{%
2732 {(\cpageref{#2})}}{%
2733 }

```

```
\nameref {⟨label⟩}

2734 \newcommand*{\nameref}[1]{%
2735 \LWR@traceinfo{nameref A}%
2736 \LWR@startref{#1}%
2737 \LWR@traceinfo{nameref B}%
2738 \LWR@nameref{#1}%
2739 \LWR@traceinfo{nameref C}%
2740 \LWR@htmltag{/a}%
2741 \LWR@traceinfo{nameref D}%
2742 }
```

**\Nameref** {⟨label⟩} In print, adds the page number. In HTML, does not.

```
2743 \let\Nameref\nameref
```

## 50.5 Hyper-references

⚠ Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`

⚠ Do not tell other packages that `hyperref` is emulated. Some packages patch various commands if `hyperref` is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.

⚠ Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
2744 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2745 % \EmulatesPackage{hyperref}[2015/08/01]%
% Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {⟨URL⟩} {⟨text⟩}
```

```
2746 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2747 \LWR@htmltag{a href="#" target="\_\_blank"\LWR@newline}\#2\LWR@htmltag{/a}%
2748 \LWR@ensuredoingapar%
2749 }
```

```
\LWR@subhyperrefclass {⟨URL⟩} {⟨text⟩} {⟨htmlclass⟩}
```

---

```

2750 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2751 \LWR@htmltag{a href="#1"%
2752 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
2753 \LWR@ensuredoingapar%
2754 }

```

**\href** [*options*] {*URL*} {*text*}

Create a link with accompanying text:

```

2755 \NewDocumentCommand{\LWR@hrefb}{O{} m +m}{%
2756 \LWR@subhyperref{#2}{#3}%
2757 \endgroup%
2758 \LWR@ensuredoingapar%
2759 }
2760
2761 \newcommand{\href}{%
2762 \LWR@ensuredoingapar%
2763 \begingroup%
2764 \catcode`\_=12
2765 \LWR@hrefb%
2766 }

```

**\nolinkurl** {*URL*}

Print the name of the link without creating the link:

```

2767 \newcommand*{\LWR@nolinkurlb}[1]{\endgroup\LWR@ensuredoingapar}
2768
2769 \newcommand{\nolinkurl}{%
2770 \LWR@ensuredoingapar%
2771 \begingroup\catcode`\_=12
2772 \LWR@nolinkurlb%
2773 }

```

**\url** {*URL*}

Create a link whose text name is the address of the link:

```

2774 \newcommand*{\LWR@urlb}[1]{%
2775 \href{#1}{#1}%
2776 \endgroup%
2777 \LWR@ensuredoingapar%
2778 }
2779
2780 \newcommand{\url}{%
2781 \LWR@ensuredoingapar%
2782 \begingroup\catcode`\_=12

```

```
2783 \LWR@urlb%
2784 }

\LWR@subinlineimage  [\langle alttag\rangle] {\langle class\rangle} {\langle filename\rangle} {\langle extension\rangle} {\langle style\rangle}

2785 \newcommand*{\LWR@subinlineimage}[5] []{%
2786 \ifthenelse{\equal{#1}{}}{%
2787 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{} }}%
2788 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{} }}%
2789 }

2790 \end{warpHTML}
```

Table 9: Float data structures

For each `<type>` of float (figure, table, etc.) there exists the following:

---

**counter <type>**: A counter called `<type>`, such as `figure`, `table`.

`\<type>name`: Name. `\figurename` prints “Figure”, etc.

`\ext@<type>`: File extension. `\ext@figure` prints “lof”, etc.

`\fps@<type>`: Placement.

`\the<type>`: Number. `\thetable` prints the number of the table, etc.

`\p@<type>`: Parent’s number. Prints the number of the [within] figure, etc.

`\fnum@<type>`: Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

`\<type>`: Starts the float environment. `\figure` or `\begin{figure}`

`\end<type>`: Ends the float environment. `\endfigure` or `\end{figure}`

`\tf@<ext>`: The L<sup>A</sup>T<sub>E</sub>X file identifier for the output file.

`LWR@have<type>`: A boolean rememebering whether a `\listof` was requested for a float of this type.

**File with extension lo<f,t,a-z>**: An output file containing the commands to build the `\listof<type><name>` “table-of-contents” structure.

**Cross-referencing names:** For `cleveref`’s `\cref` and related, `\crefname` and `\Crefname` assign human-readable names for references to this float type.

---

## 51 Floats

Floats are supported, although partially through emulation.

Table 9 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 51.1 Float captions

for HTML output: 2791 \begin{warpHTML}

```
\LWR@floatbegin {⟨type⟩} [(⟨placement⟩)]
```

Begins a \newfloat environment.

```
2792 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2793 \LWR@stopars
```

There is a new float, so increment the unique float counter:

```
2794 \addtocounter{\LWR@thisfloat}{1}%
2795 \booltrue{\LWR@freezethisfloat}%
```

```
2796 \begingroup
```

Settings while inside the environment:

```
2797 \LWR@origraggedright
```

Open an HTML `figure` tag:

```
2798 \LWR@htmntag{figure id="autofloat-\arabic{\LWR@thisfloat}" class="#1"}
```

```
2799 \renewcommand*{\@capttype}{#1}
2800 \caption@settype{#1}
2801 \LWR@startpars
2802 }
```

\@float Support packages which create floats directly.

```
\@dblfloat
2803 \let\@float\LWR@floatbegin
2804 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
2805 \newcommand*{\LWR@floatend}{%
2806 \LWR@stopars%
2807 \LWR@htmlelementend{figure}%
2808 \endgroup%
2809 \boolfalse{\LWR@freezethisfloat}%
2810 \LWR@startpars%
2811 }
```

---

```
\end@float Support packages which create floats directly.
\end@dblfloat
2812 \let\end@float\LWR@floatend
2813 \let\end@dblfloat\LWR@floatend
```

**Ctr LWR@thisfloat** A sequential counter for all floats. This is used to identify the float then reference it from the List of Figures and List of Tables.

```
2814 \newcounter{LWR@thisfloat}
```

**Bool LWR@freezethisfloat** Prevents multiple increments of \LWR@thisfloat inside a float.

```
2815 \newbool{LWR@freezethisfloat}
2816 \boolfalse{LWR@freezethisfloat}

\LWR@maybeinthisfloat
2817 \newcommand*\LWR@maybeinthisfloat{%
2818 \ifbool{LWR@freezethisfloat}{}{\addtocounter{LWR@thisfloat}{1}}%
2819 }
```

**\@capttype** Remembers which float type is in use.

```
2820 \newcommand*\@capttype{}
```

### 51.1.1 Caption inside a float environment

**\CaptionSeparator** How to separate the float number and the caption text.

```
2821 \AtBeginDocument{\providecommand*\CaptionSeparator{:-}}
```

**\@makecaption** {*name and num*} {*text*}

Prints the float type and number, the caption separator, and the caption text.

```
2822 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

### 51.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the L<sup>A</sup>T<sub>E</sub>X PDF file, the L<sup>A</sup>T<sub>E</sub>X page number at that point is stored in LWR@latestautopage, (and the associated filename is remembered by the special L<sup>A</sup>T<sub>E</sub>X labels). This page number is used to generate an **autofloat** HTML ID in the HTML output at the start of the new HTML file.

Meanwhile, there is a float counter used to generate an HTML `autofloat` IDs at the start of the float itself in the HTML file. The `autopage` and `autofloat` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
2823 \newcounter{LWR@nextautofloat}
2824 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat {⟨autopage⟩} {⟨autofloat⟩}`

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the HTML links.

```
2825 \newcommand*{\LWRsetnextfloat}[2]{%
2826 \setcounter{LWR@nextautopage}{#1}%
2827 \setcounter{LWR@nextautofloat}{#2}%
2828 }
```

Ctr `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autofloat` by the modified `\addcontentsline` just before each float's entry.

```
2829 \newcounter{LWR@latestautopage}%
2830 \setcounter{LWR@latestautopage}{1}
```

```
2831 \let\LWR@origcaption@begin\caption@begin
2832 \let\LWR@origcaption@end\caption@end
2833 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2834 \newcommand{\LWR@caption@begin}{%
2835 {
2836 \LWR@traceinfo{\LWR@caption@begin}%
}
```

Keep par and minipage changes local:

```
2837 \begingroup%
```

The `caption` code was not allowing the closing `par` tag:

```
2838 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%

```

No need for a minipage inside the caption:

```
2839 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{\begin{minipage}[t]{#1}\begin{array}{#2}\end{array}\end{minipage}}
```

Enclose the original caption code inside an HTML tag:

```
2840 \LWR@htmlblocktag{figcaption}%
2841 \LWR@origcaption@begin%
2842 }
```

\LWR@caption@end Low-level patches to create HTML tags for captions.

```
2843 \newcommand{\LWR@caption@end}{%
2844 \endgroup%
2845 \LWR@origcaption@end}
```

Subcaptions were being over-written by the closing HTML tag:

```
2846 \vspace*{\baselineskip}
```

Closing tag:

```
2847 \LWR@htmlblocktag{/figcaption}%
2848 \endgroup%
2849 % \leavevmode% avoid bad space factor (0) error
2850 \LWR@traceinfo{\LWR@caption@end: done}%
2851 }
```

\caption@begin Low-level patches to create HTML tags for captions.

```
\caption@end
2852 \let\caption@begin\LWR@caption@begin
2853 \let\caption@end\LWR@caption@end
```

\captionlistentry Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
2854 \let\LWR@origcaptionlistentry\captionlistentry
2855
2856 \renewcommand*{\captionlistentry}{%
2857 \LWR@maybeinthisfloat%
2858 \LWR@ensuredoingapar%
2859 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"{} }\LWR@htmlltag{/a}%
2860 \LWR@origcaptionlistentry%
2861 }
2862
2863 \def\LWR@LTcaptionlistentry{%
2864 \LWR@ensuredoingapar%
```

---

```

2865 \LWR@htmntag{a id="autofloat-\arabic{LWR@thisfloat}""{} }\LWR@htmntag{/a}%
2866   \bgroup
2867   @ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
2868     {\egroup\LWR@LT@captionlistentry}}%
2869 \def\LWR@LT@captionlistentry#1{%
2870   \caption@listentry@\firstoftwo[\LTcaptive]{#1}}%

```

**\addcontentsline** Patched to write the autopage and autofloat before each float's entry. No changes if writing `.toc`.

```

2871 \let\LWR@origaddcontentsline\addcontentsline
2872
2873 \renewcommand*\addcontentsline[3]{%
2874 \ifthenelse{\equal{#1}{toc}}{}{%
2875 \addtocontents{\@nameuse{ext@#2}}{%
2876 \protect\LWRsetnextfloat%
2877 {\arabic{LWR@latestautopage}}%
2878 {\arabic{LWR@thisfloat}}}}%
2879 }%
2880 }%
2881 \LWR@origaddcontentsline{#1}{#2}{#3}%
2882 }

```

**\captionof** Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

2883 \AtBeginDocument{
2884 \let\LWR@origcaptionof\captionof
2885
2886 \renewcommand*\captionof{%
2887 \LWR@maybeincthisfloat%
2888 \LWR@stopars
2889 \LWR@htmntag{a id="autofloat-\arabic{LWR@thisfloat}""{} }\LWR@htmntag{/a}%
2890 \LWR@origcaptionof%
2891 }
2892 }

2893 \end{warpHTML}

```

## 52 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a div of class `.toc`, `.lof`, or `.lot`.

A “sidetoc” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular L<sup>A</sup>T<sub>E</sub>X infrastructure is used for TOC, along with some patches to generate HTML output.

**for HTML output:** 2894 \begin{warpHTML}

## 52.1 Reading and printing the TOC

\LWR@myshorttoc {\langle toc/lof/lot\rangle}

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular L<sup>A</sup>T<sub>E</sub>X, the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideTOC.

2895 \newcommand\*\LWR@myshorttoc[1]{  
2896 \LWR@ensuredoingapar

Only if the file exists:

2897 \IfFileExists{\jobname.\#1}{

⚠ Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.

⚠ disabled For pdflatex, also change to latin1 encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it utf8.

2898 \begingroup  
2899 % \ifxetexorluatex%  
2900 % \else  
2901 % \inputencoding{latin1}% currently disabled  
2902 % \fi  
2903 \makeatletter

Read in the TOC file:

2904 \@input{\jobname.\#1}  
2905 % \makeatother  
2906 \endgroup  
2907 }%  
2908 {}%

2909 }

\LWR@subtableofcontents {⟨toc/lof/lot⟩} {⟨sectionstarname⟩}

Places a TOC/LOF/LOT at the current position.

2910 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

Closes previous levels:

2911 \@ifundefined{chapter}  
2912 {\LWR@closeprevious{\LWR@depthsection}}  
2913 {\LWR@closeprevious{\LWR@depthchapter}}

Prints any pending footnotes so that they appear above the potentially large TOC:

2914 \LWR@printpendingnotes

Place the list into its own chapter (if defined) or section:

2915 \@ifundefined{chapter}{\section\*{#2}}{\chapter\*{#2}}

Create a new HTML nav containing the TOC/LOF/LOT:

2916 \LWR@htmlelementclass{nav}{#1}

Create the actual list:

2917 \LWR@myshorttoc{#1}

Close the nav:

2918 \LWR@htmlelementclassend{nav}{#1}  
2919 }

Patch \@starttoc to encapsulate the TOC inside HTML tags:

2920 \let\LWR@orig@starttoc\@starttoc  
2921  
2922 \renewcommand{\@starttoc}[1]{  
2923 \LWR@htmlelementclass{nav}{#1}  
2924 \LWR@orig@starttoc{#1}  
2925 \LWR@htmlelementclassend{nav}{#1}  
2926 }

Patch \tableofcontents, etc. to print footnotes first. newfloat uses \listoffigures for all future float types.

---

```
2927 \let\LWR@origtableofcontents\tableofcontents
2928 \let\LWR@origlistoffigures\listoffigures
2929 \let\LWR@origlistoftables\listoftables
2930 \renewcommand*\{tableofcontents}{%
```

Copy the `.toc` file to `.sidetoc` for printing the sidetoc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
2931 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
2932 \LWR@printpendingnotes%
2933 \LWR@origtableofcontents%
2934 }
2935 \renewcommand*\{listoffigures}{\LWR@printpendingnotes\LWR@origlistoffigures}
2936 \renewcommand*\{listoftables}{\LWR@printpendingnotes\LWR@origlistoftables}
```

## 52.2 High-level TOC commands

`\listof {<type>} {<title>}`

Emulate the `\listof` command from the `float` package (section 86). Used to create lists of custom float types. Also used to redefine the standard L<sup>A</sup>T<sub>E</sub>X `\listoffigures` and `\listoftables` commands.

```
2937 \NewDocumentCommand{\listof}{m +m}{%
2938 \LWR@subtableofcontents{\@nameuse{ext@\#1}}{\#2}
2939 \expandafter\newwrite\csname tf@\csname ext@\#1\endcsname\endcsname
2940 \immediate\openout \csname tf@\csname ext@\#1\endcsname\endcsname
2941     \jobname.\csname ext@\#1\relax
2942 }
```

## 52.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

css may be used to format the sideTOC:

CSS related to sideTOC:

---

`nav.sidetoc`: The entire sideTOC.  
`div.sidetoctitle`: The title.  
`div.sidetoccontents`: The table of contents.

---

```
2943 \end{warpHTML}
```

**for HTML & PRINT:** 2944 \begin{warpall}

Ctr SideTOCDepth Controls how deep the side-TOC gets. Use a standard L<sup>A</sup>T<sub>E</sub>X section level similar to tocdepth.

```
2945 \newcounter{SideTOCDepth}
2946 \setcounter{SideTOCDepth}{1}
```

\sidetocname Holds the default name for the sidetoc.

```
2947 \newcommand{\sidetocname}{Contents}
2948 \end{warpall}
```

**for HTML output:** 2949 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
2950 \newcommand*{\LWR@sidetoc}{
2951 \LWR@stoppars
2952
```

The entire sideTOC is placed into a nav of class sidetoc.

```
2953 \LWR@htmlelementclass{nav}{sidetoc}
2954
2955 \setcounter{tocdepth}{\value{SideTOCDepth}}
2956
```

The title is placed into a div of class sidetoctitle, and may contain paragraphs.

```
2957 \begin{BlockClass}{sidetoctitle}
2958 \sidetocname
2959 \end{BlockClass}
```

The table of contents is placed into a div of class sidetoccontents.

```
2960 \begin{BlockClass}{sidetoccontents}
2961 \LinkHome
2962
2963 \LWR@myshorttoc{sidetoc}
2964 \end{BlockClass}
2965 \LWR@htmlelementclassend{nav}{sidetoc}
2966 }
```

## 52.4 Low-level TOC line formatting

\numberline {⟨number⟩}

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
2967 \renewcommand*\numberline[1]{%
2968 \LWR@sectionnumber{#1}%
2969 }
```

\hypertoc {⟨1: depth⟩} {⟨2: class⟩} {⟨3: name⟩} {⟨4: page⟩}

Called by \l@section, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

#1 is depth

#2 HTML CSS class is tocsection, tocsubsection, etc.

#3 the text of the caption

#4 page number

```
2970 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
2971 \ifthenelse{\cnttest{#1}<=\{\value{tocdepth}\}}{%
2972 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
2973 \LWR@subhyperrefclass{%
2974 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{#2}%
2975 \LWR@stoppars%
2976 }%
2977 {}%
2978 }
```

\hypertocfloat {⟨1: depth⟩} {⟨2: class⟩} {⟨3: caption⟩} {⟨4: page⟩}

#1 is depth

#2 HTML CSS class is tocfigure, toctable

#3 the text of the caption

#4 page number

```
2979 \newcommand{\hypertocfloat}[4]{%
2980 \LWR@startpars
```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```
2981 \LWR@subhyperrefclass{%
2982 \LWR@htmrefsectionfilename{autopage-\arabic{\LWR@nextautopage}}%
2983 \#autofloat-\arabic{\LWR@nextautofloat}}%
2984 {#3}{#2}%
2985 \LWR@stoppars%
2986 }
```

Automatically called by \contentsline:

```
2987 \renewcommand{\l@part}[2]{\hypertoc{-1}{tocpart}{#1}{#2}}
2988 \DeclareDocumentCommand{\l@chapter}{m m}
2989 {\hypertoc{0}{tocchapter}{#1}{#2}}
2990 \renewcommand{\l@section}[2]{\hypertoc{1}{tocsection}{#1}{#2}}
2991 \renewcommand{\l@subsection}[2]{\hypertoc{2}{tocsubsection}{#1}{#2}}
2992 \renewcommand{\l@subsubsection}[2]
2993 {\hypertoc{3}{tocsubsubsection}{#1}{#2}}
2994 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{tocparagraph}{#1}{#2}}
2995 \renewcommand{\l@ subparagraph}[2]{\hypertoc{5}{toc subparagraph}{#1}{#2}}
2996 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{tocfigure}{#1}{#2}}
2997 \renewcommand{\l@table}[2]{\hypertocfloat{1}{toctable}{#1}{#2}}
```

```
2998 \end{warpHTML}
```

## 53 Index

See:

[http://tex.stackexchange.com/questions/187038/  
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

for HTML output: 2999 \begin{warpHTML}

\LWR@indexsection Controls whether the index will be in a section or a chapter, depending on the documentclass.

```
3000 \@ifundefined{chapter}
3001 {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
3002 {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}

\printindex

3003 \let\LWR@origprintindex\printindex
3004
3005 \renewcommand*{\printindex}
3006 {
3007 \LWR@indexsection
3008 \LWR@startpars
3009 \LWR@origprintindex
3010 }

Env   theindex

3011 \renewenvironment*{\theindex}{%
3012 \let\item\LWR@indexitem%
3013 \let\subitem\LWR@indexsubitem%
3014 \let\subsubitem\LWR@indexsubsubitem%
3015 }{}

\LWR@indexitem

3016 \newcommand{\LWR@indexitem}{%
3017
3018 \InlineClass{indexitem}{}%
3019 }

\LWR@indexitem

3020 \newcommand{\LWR@indexsubitem}{%
3021
3022 \InlineClass{indexsubitem}{}%
3023 }

\LWR@indexitem

3024 \newcommand{\LWR@indexsubsubitem}{%
3025
3026 \InlineClass{indexsubsubitem}{}%
3027 }

\hyperindexref  {\langle autosecnumber\rangle}
```

---

\hyperindexref{web address} is inserted into \*.ind by the xindy style file lwarf\_html.xdy

```
3028 \newcommand*\hyperindexref[1]{  
3029 \LWR@htmltag{a href="%
```

Create an HTML reference to the autosection:

```
3030 \LWR@htmlrefsectionfilename{autopage-#1}\#autosec-#1"{}%  
3031 \LWR@origref{autopage-#1}\enskip\LWR@nameref{autopage-#1}%  
3032 \LWR@htmltag{/a}%  
3033 }
```

```
3034 \end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case hyperref was not used:

```
3035 \begin{warpprint}  
3036 \newcommand{\hyperindexref}[1]{#1}  
3037 \end{warpprint}
```

## 54 Math

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

**SVG math option** For SVG math, math is rendered as usual by L<sup>A</sup>T<sub>E</sub>X into the initial PDF file using the current font<sup>7</sup>, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by L<sup>A</sup>T<sub>E</sub>X with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the L<sup>A</sup>T<sub>E</sub>X code which generated the math, allowing copy/paste of the L<sup>A</sup>T<sub>E</sub>X math expression into other documents.

**SVG image font size** The size of the math and text used in the SVG image may be adjusted by setting \LateximageFontSizeName to a font size name — *without the backslash*, for ex:  
\renewcommand{\LateximageFontSizeName}{large}

**SVG files** As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files

---

<sup>7</sup>See section 136 regarding fonts and fractions.

with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.

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

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

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

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

Prog MathJax

When MathJax is enabled, math is rendered twice:

1. As regular L<sup>A</sup>T<sub>E</sub>X PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of L<sup>A</sup>T<sub>E</sub>X, and
2. As detokenized printed L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X values, and the MathJax cross-referencing system is ignored in favor of the L<sup>A</sup>T<sub>E</sub>X internal system, seamlessly integrating with the rest of the L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X. \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 L<sup>A</sup>T<sub>E</sub>X 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/mathjax/MathJax-third-party-extensions/  
tree/master/siunitx](https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx)
- `LATEX macros` • MathJax does not automatically support custom L<sup>A</sup>T<sub>E</sub>X macros, but they may be set up by the user.

**custom MathJax macros** As an example of using custom L<sup>A</sup>T<sub>E</sub>X macros with MathJax, place the following at the start of the document, after `\begin{document}`:

---

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
  % New macros for MathJax are placed inside a math expression:
  \newcommand{\expval}[1]{\langle#1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
}
\end{warpHTML}
```

---

**for HTML output:** 3038 `\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.

```
3039 \let\LWR@origtextdollar\
3040
3041 \renewcommand*{\$}{%
3042 \LWR@htmlltagc{span}%
3043 \LWR@origtextdollar%
3044 \LWR@htmlltagc{/span}%
3045 }
```

**Ctr LWR@externalfilecnt** Counter for the external files which are generated and then referenced from the HTML:

```
3046 \newcounter{LWR@externalfilecnt}
```

## 54.1 Inline and display math

```
3047 \let\LWR@origdollar=$
3048 \let\secondorigdollar=$% balance for editor syntax highlighting

3049 \let\LWR@origopenparen\
3050 \let\LWR@origcloseparen\
```

**\$** Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:  
**\$\$**

```
3051 \begingroup
3052 \catcode`$=\active%
3053 \protected\gdef${\ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
3054 \gdef\LWR@doubledollar##1$${
3055 \ifbool{mathjax}{}
```

For MathJax, print the math between `\[` and `\]`:

```
3056 {\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the L<sup>A</sup>T<sub>E</sub>X code:

```
3057 {%
3058   \begin{lateximage}
3059     \begin{math}
3060       [\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]{}]
3061       \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
3062     \end{math}
3063   \end{lateximage}
3064 }
3065 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
3066 \gdef\LWR@singledollar#1${%
3067 \ifbool{mathjax}{}
```

For MathJax, print the math between `\(` and `\)`:

```
3068 {\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the L<sup>A</sup>T<sub>E</sub>X code:

```
3069 {%
3070   \begin{lateximage}
3071     \begin{math}
3072       [\textbackslash[LWR@HTMLsanitize{\#1}\textbackslash]{}]
3073     \end{math}
3074   \end{lateximage}
3075 }%
```

---

```

3072 \LWR@origdollar#1\LWR@origdollar%
3073 \end{lateximage}%
3074 }%
3075 }%

\(\ Redefine to the above dollar macros.
\)
3076 \gdef\(#1\){$#1$}
3077 \gdef\[#1\]{$$#1$$}
3078
3079 \endgroup

```

Remove the old `math` and `displaymath` environments:

```

3080 \let\math\relax
3081 \let\endmath\relax
3082 \let\displaymath\relax
3083 \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`.

```
3084 \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`.

```
3085 \NewEnviron{displaymath}{\expandafter\[\BODY\]\ignoretrue}
```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```
3086 \AtBeginDocument{\catcode`\$=\active}
```

## 54.2 MathJax support

`Ctr LWR@nextequation` Used to add one to compute the next equation number.

```
3087 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “`\(`” and “`\)`” characters. They are printed to HTML output, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

---

```
3088 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
3089 \ifcsdef{thechapter}{
3090   \BlockClassSingle{hidden}{
3091     \textbackslash(
3092     \textbackslash) \seteq \section{\thechapter}
3093     \textbackslash)
3094   }
3095 }
3096 {}% not using chapters
3097
```

MathJax doesn't allow setting the equation number to 1:

```
3098 \ifthenelse{\cnttest{\value{equation}}>0}{}
3099 {
```

Tell MathJax that the next set of equations begins with the current L<sup>A</sup>T<sub>E</sub>X equation number, plus one.

```
3100 \setcounter{LWR@nextequation}{\value{equation}}
3101 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\(” and “\)” characters, to be printed to HTML, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

```
3102 \BlockClassSingle{hidden}{
3103   \textbackslash(
3104   \textbackslash) \seteq \arabic{LWR@nextequation}
3105   \textbackslash)
3106 }
3107 }{}% not eq > 1
3108 }
```

`\LWR@restoremathlatexformatting` While producing math, use regular L<sup>A</sup>T<sub>E</sub>X formatting instead of HTML tags.

```
3109 \newcommand*{\LWR@restoremathlatexformatting}{%
3110   \let\hspace{\LWR@origspace}%
3111   \let\rule{\LWR@origrule}%
3112   \let\,,\LWR@origcomma% disable HTML short unbreakable space
3113   \let\textit{\LWR@origtextit}%
3114   \let\textbf{\LWR@origtextbf}%
3115   \let\texttt{\LWR@origtexttt}%
3116   \let\textsc{\LWR@origtextsc}%
3117   \let\textsf{\LWR@origtextsf}%
3118 }
```

---

```

3118 \let\textrm{\LWR@origtextrm%
3119 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3120 \let\textsuperscript{\LWR@origtextsuperscript%
3121 \let\textsubscript{\LWR@origtextsubscript%
3122 \let~\LWR@origtilde%
3123 \let\enskip\LWR@origenskip%
3124 \let\quad\LWR@origquad%
3125 \let\quad\LWR@origquad%
3126 }

```

`\LWR@hidelatexequation {⟨environment⟩} {⟨contents⟩}`

Creates the L<sup>A</sup>T<sub>E</sub>X version of the equation inside an HTML comment.

```

3127 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%

```

Stop HTML paragraph handling and open an HTML comment:

```

3128 \LWR@stopars
3129 \LWR@htmlopencomment
3130

```

Start the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```

3131 \begingroup
3132 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their L<sup>A</sup>T<sub>E</sub>X meanings.

```

3133 \LWR@restoremathlatexformatting

```

When a label is encountered, the HTML comment is closed, the label is generated in an HTML-output context, then the HTML comment is opened again. See `\LWR@htmlmathlabel` in section 54.4.1.

Print the contents of the equation:

```

3134 #2

```

End the L<sup>A</sup>T<sub>E</sub>X math environment inside the HTML comment:

```

3135 \csuse{\LWR@origend#1}
3136 \endgroup
3137

```

Close the HTML comment and resume HTML paragraph handling:

```

3138 \LWR@htmclosecomment

```

---

```
3139 \LWR@startpars
3140 }
```

\LWR@addmathjax {⟨environment⟩} {⟨contents⟩}

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by L<sup>A</sup>T<sub>E</sub>X.

```
3141 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3142 }
```

Enclose the MathJax environment inside printed “\⟨” and “\⟩” characters.

```
3143 \textbackslash{}begin{#1\`}
```

Print the contents, sanitizing for HTML special characters.

```
3144 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
3145 \textbackslash{}end{#1\`}
3146
3147 }
```

### 54.3 Equation environment

Remember existing `equation` environment:

```
3148 \let\LWR@origequation\equation
3149 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
3150 \let\equation\relax
3151 \let\endequation\relax
```

**Env `equation`** The new `equation` environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the L<sup>A</sup>T<sub>E</sub>X source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
3152 \NewEnviron{equation}{%
3153
3154 \ifbool{mathjax}
```

MathJax output:

```
3155 {
```

Print commands to synchronize MathJax's equation number and format to the current L<sup>A</sup>T<sub>E</sub>X chapter/section and equation number:

```
3156 \LWR@syncmathjax
```

Print the L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment:

```
3157 \LWR@hidelatexequation{equation}{\BODY}
3158 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the L<sup>A</sup>T<sub>E</sub>X equation environment commands, and the contents of the environment's `\BODY`.

```
3159 {%
  not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3160 \begin{lateximage}[(\theequation) \textbackslash\begin\{equation\}\}%
3161 \LWR@HTMLsanitizeexpand{\detokenize\expandafter\{\BODY\}\}%
3162 \textbackslash\end\{equation\}\}]% alt tag
```

Create the actual L<sup>A</sup>T<sub>E</sub>X-formatted equation inside the `lateximage` using the contents of the environment.

```
3163 \LWR@origequation
3164 \BODY% contents collected by NewEnviron
3165 \LWR@origendequation
3166 \end{lateximage}%
3167 }
3168
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3169 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

## 54.4 AMS Math environments

### 54.4.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if it is used in an `amsmath` environment which is not a `multiline` environment and not an `equation`.

```
3170 \newbool{LWR@amsmultiline}
3171 \boolfalse{LWR@amsmultiline}
```

```
\LWR@htmlmathlabel {<label>}
```

`\label` when called inside a `LATEX` AMS math environment's math mode which is encased inside an HTML comment while creating a HTML document.

```
3172
3173 \newcommand*{\LWR@htmlmathlabel}[1]{%
3174 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3175 \ifbool{mathjax}{%
```

The combined `LATEX` & HTML label is printed in a `\text` field:

```
3176 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3177 \ifbool{LWR@amsmultiline}{}{\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.)

```
3178 \LWR@htmclosecomment\LWR@sublabel{\@firstofone#1}\LWR@htmopencomment}%
3179 }%
3180 {\LWR@sublabel{\@firstofone#1}}%
3181 }
```

```
\LWR@beginhideamsmath Starts hiding LATEX math inside an HTML comment.
```

```
3182 \newcommand*{\LWR@beginhideamsmath}{%
3183 \LWR@stoppars
3184
3185 \LWR@htmopencomment}
```

---

```

3186
3187 \begingroup
3188 \LWR@restoremathlatexformatting
3189 }
```

\LWR@endhideamsmath Ends hiding L<sup>A</sup>T<sub>E</sub>X math inside an HTML comment.

```

3190 \newcommand*{\LWR@endhideamsmath}{
3191 \endgroup
3192
3193 \LWR@htmlclosecomment
3194
3195 \LWR@startpars
3196 }
```

#### 54.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWR@AMS@BODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the  $\mathcal{M}\mathcal{S}$  environment.

For MathJax: Each environment is syched with L<sup>A</sup>T<sub>E</sub>X's equation numbers, typeset with L<sup>A</sup>T<sub>E</sub>X inside an HTML comment, then printed to HTML output for MathJax to process.

```

Env  multiline

3197 \BeforeBeginEnvironment{multiline}{
3198 \ifbool{mathjax}
3199 {
3200 \LWR@syncmathjax
3201 \booltrue{\LWR@amsmultiline}
3202 \LWR@beginhideamsmath
3203 }
3204 {
3205 \latexitimage[LWR@AMS@BODY]
3206 }
3207 }
3208
3209 \AfterEndEnvironment{multiline}{
3210 \ifbool{mathjax}
3211 {
```

```
3213 \LWR@endhideamsmath
3214 \boolfalse{LWR@amsmultline}
3215 \LWR@addmathjax{multiline}{\the\@envbody}
3216 }
3217 {\endlateximage}
3218
3219 }

Env  multiline*
3220 \BeforeBeginEnvironment{multiline*}{

3221 \ifbool{mathjax}
3222 {
3223 \LWR@syncmathjax
3224 \booltrue{LWR@amsmultline}
3225 \LWR@beginhideamsmath
3226 }
3227 {
3228 \latexitimage[LWRAMSMATHBODY]
3229 }
3230 }
3231
3232 \AfterEndEnvironment{multiline*}{

3233
3234 \ifbool{mathjax}
3235 {
3236 \LWR@endhideamsmath
3237 \boolfalse{LWR@amsmultline}
3238 \LWR@addmathjax{multiline*}{\the\@envbody}
3239 }
3240 {\endlateximage}
3241
3242 }
3243

Env  gather
3244 \BeforeBeginEnvironment{gather}{

3245 \ifbool{mathjax}
3246 {
3247 \LWR@syncmathjax
3248 \boolfalse{LWR@amsmultline}
3249 \LWR@beginhideamsmath
3250 }
3251 {
3252 \latexitimage[LWRAMSMATHBODY]
3253 }
3254 }
3255 }
```

```
3256 \AfterEndEnvironment{gather}{

3257
3258 \ifbool{mathjax}
3259 {
3260 \LWR@endhideamsmath
3261 \LWR@addmathjax{gather}{\the\@envbody}
3262 }
3263 {\endlateximage}
3264
3265 }

Env gather*

3266 \BeforeBeginEnvironment{gather*}{

3267 \ifbool{mathjax}
3268 {
3269 \LWR@syncmathjax
3270 \boolfalse{LWR@amsmultiline}
3271 \LWR@beginhideamsmath
3272 }
3273 {
3274 \latexitimage[LWRAMSMATHBODY]
3275 }
3276 }
3277
3278 \AfterEndEnvironment{gather*}{

3279 \ifbool{mathjax}
3280 {
3281 \LWR@endhideamsmath
3282 \LWR@addmathjax{gather*}{\the\@envbody}
3283 }
3284
3285 {\endlateximage}
3286
3287 }

Env align

3288 \BeforeBeginEnvironment{align}{

3289 \ifbool{mathjax}
3290 {
3291 \LWR@syncmathjax
3292 \boolfalse{LWR@amsmultiline}
3293 \LWR@beginhideamsmath
3294 }
3295 {
3296 \latexitimage[LWRAMSMATHBODY]
3297 }
3298 }
```

```
3299
3300 \AfterEndEnvironment{align}{
3301
3302 \ifbool{mathjax}
3303 {
3304 \LWR@endhideamsmath
3305 \LWR@addmathjax{align}{\the\@envbody}
3306 }
3307 {\endlateximage}
3308
3309 }

Env align*
3310 \BeforeBeginEnvironment{align*}{

3311 \ifbool{mathjax}
3312 {
3313 \LWR@syncmathjax
3314 \boolfalse{LWR@amsmultiline}
3315 \LWR@beginhideamsmath
3316 }
3317 {
3318 \lateximage[LWRAMSMATHBODY]
3319 }
3320 }
3321
3322 \AfterEndEnvironment{align*}{

3323 \ifbool{mathjax}
3324 {
3325 \LWR@endhideamsmath
3326 \LWR@addmathjax{align*}{\the\@envbody}
3327 }
3328
3329 {\endlateximage}
3330
3331 }

Env flalign
3332 \BeforeBeginEnvironment{flalign}{

3333 \ifbool{mathjax}
3334 {
3335 \LWR@syncmathjax
3336 \boolfalse{LWR@amsmultiline}
3337 \LWR@beginhideamsmath
3338 }
3339 {
3340 \lateximage[LWRAMSMATHBODY]
3341 }
```

```

3342 }
3343
3344 \AfterEndEnvironment{flalign}{
3345
3346 \ifbool{mathjax}
3347 {
3348 \LWR@endhideamsmath
3349 \LWR@addmathjax{flalign}{\the\@envbody}
3350 }
3351 {\endlateximage}
3352
3353 }

Env flalign*
3354 \BeforeBeginEnvironment{flalign*}{

3355 \ifbool{mathjax}
3356 {
3357 \LWR@syncmathjax
3358 \boolfalse{LWR@amsmultiline}
3359 \LWR@beginhideamsmath
3360 }
3361 {
3362 \lateximage[LWRAMSMATHBODY]
3363 }
3364
3365
3366 \AfterEndEnvironment{flalign*}{

3367
3368 \ifbool{mathjax}
3369 {
3370 \LWR@endhideamsmath
3371 \LWR@addmathjax{flalign*}{\the\@envbody}
3372 }
3373 {\endlateximage}
3374
3375 }

3376 \end{warpHTML}

```

## 55 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment starting on the preceding page and ending on following page, and instructions are written to `lateximage.txt` for `lwarfpmk` 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.

Since the `\lateximage` is placed onto its own PDF page, which `pdftotext` will read and try to convert to text, an HTML open comment is placed onto the bottom of the current page, but it must appear below any footnotes on the current page, so a special page style (`LWR@commentfooterstyle`) is used.

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.

**SVG image font size** The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:  
`\renewcommand{\LateximageFontSizeName}{large}`

**for HTML output:** 3377 `\begin{warpHTML}`

**LWR@commentfooterstyle**

```
3378 \newpagestyle{LWR@commentfooterstyle}{
3379 \sethead{}{}{}
3380 \setfoot{\LWR@htmlopencomment}{}{}
3381 }
```

Ctr `LWR@lateximagenumber` Sequence the images.

```
3382 \newcounter{LWR@lateximagenumber}
3383 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@latexitagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
3384 \newcounter{LWR@latexitagedepth}
3385 \setcounter{LWR@latexitagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `latexitimes.txt`:

```
3386 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3387 \edef\LWR@hashmark{\string#} % for use in \write
3388 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@nextpage` Used to count ahead to the next page:

```
3389 \newcounter{LWR@nextpage}
```

```
3390 \end{warpHTML}
```

**for HTML & PRINT:** 3391 \begin{warpall}

\LateximageFontSizeName Declares how large to write text in the \latexitimage. The .svg file text size should blend well with the surrounding HTML text size.

*Do not include the leading backslash in the name.*

3392 \newcommand\*\LateximageFontSizeName{large}

3393 \end{warpall}

**for HTML output:** 3394 \begin{warpHTML}

\LWR@HTMLsanitize {\text{}}

Math expressions are converted to `latexitimages`, 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...

```
3395 \newcommand{\LWR@HTMLsanitize}[1]{%
3396 \protect\StrSubstitute{\detokenize{\#1}}{%
3397 {\detokenize{&}}}{%
3398 {\detokenize{&#}}}{[\LWR@strresult]}{%
3399 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3400 {\detokenize{<}}}{%
3401 {\detokenize{&lt;}}}{[\LWR@strresult]}{%
3402 {\LWR@strresult}}{%
3403 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3404 {\detokenize{>}}}{%
3405 {\detokenize{&gt;}}}{[\LWR@strresult]}{%
3406 {\LWR@strresult}}{%
3407 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3408 {\detokenize{##}}}{%
3409 {\#}}}{%
3410 [\LWR@strresult]}{%
3411 \LWR@strresult}{%
3412 }}
```

\LWR@HTMLsanitizeexpand {\text{}}

This version expands the argument before sanitizing it.

```
3413 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3414 \protect\StrSubstitute{\detokenize\expandafter{\#1}}{%
3415 {\detokenize{&}}}{%
```

---

```

3416 {\detokenize{&}}%
3417 [\\LWR@strresult]%
3418 \protect\StrSubstitute{\detokenize\expandafter{\\LWR@strresult}}%
3419 {\detokenize{<}}%
3420 {\detokenize{<}}%
3421 [\\LWR@strresult]%
3422 \protect\StrSubstitute{\detokenize\expandafter{\\LWR@strresult}}%
3423 {\detokenize{>}}%
3424 {\detokenize{>}}%
3425 [\\LWR@strresult]%
3426 \\LWR@strresult%
3427 }

```

Env `lateximage` [ $\langle alttag \rangle$ ]

```

3428 \NewDocumentEnvironment{lateximage}{O{image}}{%
3429 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

3430 {%
3431 \addtocounter{LWR@lateximagedepth}{1}%
3432 }%

```

Otherwise, this is the outer-most `lateximage`:

```
3433 { % start of outer-most lateximage
```

Starting a new `lateximage`:

```
3434 \addtocounter{LWR@lateximagenumber}{1}%

```

While inside a `lateximage`, do not use `mathjax`:

```
3435 \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
3436 \\LWR@ensuredoingapar%
```

Next file:

```
3437 \addtocounter{LWR@externalfilecnt}{1}%

```

Figure out what the next page number will be:

```

3438 \setcounter{LWR@nextpage}{\\thepage}%
3439 \addtocounter{LWR@nextpage}{1}%

```

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:

```
3440 \LWR@htmlltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
3441 class="lateximagesource"}{} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
3442 \immediate\write\LWR@file{|\theLWR@nextpage|\theLWR@externalfilecnt|}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```
3443 \LWR@origthispagestyle{LWR@commentfooterstyle}%
3444 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
3445 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
3446 \LWR@origminipage{6in}%
3447 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
3448 \let\hspace\LWR@orighspace%
3449 \let\rule\LWR@origrule%
3450 \let\,,\LWR@origcomma% disable HTML short unbreakable space
3451 \let\textit\LWR@origtextit%
3452 \let\textbf\LWR@origtextbf%
3453 \let\texttt\LWR@origtexttt%
3454 \let\textsc\LWR@origtextsc%
3455 \let\textsf\LWR@origtextsf%
3456 \let\textrm\LWR@origtextrm%
3457 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3458 \let\textsuperscript\LWR@origtextsupsript%
3459 \let\textsubscript\LWR@origtextsubscript%
3460 \let~\LWR@origtilde%
3461 \let\enskip\LWR@origenskip%
3462 \let\quad\LWR@origquad%
3463 \let\qquad\LWR@origqquad%
3464 \let\tabular\LWR@origtabular%
3465 \let\endtabular\LWR@origendtabular%
3466 \let\newline\LWR@orignewline%
3467 }% end of outer-most lateximage
```

---

```
3468 }% end of \begin{lateximage}
3469 {% start of \end{lateximage}
3470 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3471 {%
3472 \addtocounter{LWR@lateximagedepth}{-1}%
3473 }%
```

if this is the outer-most lateximage:

```
3474 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
3475 \LWR@origendminipage%
3476 \LWR@orignewline%
3477 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
3478 \LWR@htmclosecomment{} \LWR@orignewline%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```
3479 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
3480 {%
3481 \LWR@subinlineimage[%
3482 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
3483 ]%
3484 {lateximage}%
3485 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
3486 {svg}%
3487 {}%
3488 }%
3489 {%
3490 \LWR@subinlineimage[#1]{lateximage}%
3491 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
3492 }%
3493 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
3494 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
3495 \LWR@htmlltag{/span}%
3496 \LWR@htmlcomment{End of lateximage}%
3497 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
3498 \addtocounter{LWR@lateximagedepth}{-1}%
3499 }% end of outer-most lateximage
3500 }%
3501 \end{warpHTML}
```

**for PRINT output:** 3502 \begin{warpprint}
3503 \newenvironment{lateximage}[1] [] {\begingroup}{\endgroup}
3504 \end{warpprint}

## 56 center, flushleft, flushright

**for HTML output:** 3505 \begin{warpHTML}

Env center Replace center functionality with CSS tags:

```
3506 \renewenvironment*{center}%
3507 {\BlockClass{center}}%
3508 {\endBlockClass}
```

Env flushright

```
3509 \renewenvironment*{flushright}%
3510 {\BlockClass{flushright}}%
3511 {\endBlockClass}
```

Env flushleft

```
3512 \renewenvironment*{flushleft}%
3513 {\BlockClass{flushleft}}%
3514 {\endBlockClass}
```

```
3515 \end{warpHTML}
```

## 57 Siunitx

Pkg **siunitx**

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

for HTML output: 3516 `\begin{warpHTML}`

Options for siunitx:

```
3517 \PassOptionsToPackage{
3518 detect-mode=true,
3519 per-mode=symbol,% fraction is not seen by pdftotext
3520 text-celsius = {\HTMLentity{deg}C},
3521 text-degree = {\HTMLentity{deg}},
3522 }{siunitx}

3523 \end{warpHTML}
```

## 58 Graphics

Pkg **graphics**

Pkg **graphicx**

**\graphicspath** `\graphicspath` only works for a single directory; all graphics must be in this directory.

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

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

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

**△ image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for HTML output, and

will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.

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

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

**for HTML output:** 3524 `\begin{warpHTML}`

## 58.1 `\graphicspath`

```
\graphicspath {<path>}

3525 \newcommand*{\thisgraphicspath}{{}}
3526 \renewcommand*{\graphicspath}[1]{\renewcommand*{\thisgraphicspath}{#1}}
```

`\DeclareGraphicsExtensions` {<list>}

```
\DeclareGraphicsRule {} {} {} {}

3527 \renewcommand*{\DeclareGraphicsExtensions}[1]{}
3528 \renewcommand*{\DeclareGraphicsRule}[4]{}
```

## 58.2 Length conversions and graphics options

 **⚠ whitespace** A scaled image in L<sup>A</sup>T<sub>E</sub>X by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

3529 `\renewcommand*{\unitspace}{}`

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyz` option.

```
3530 \newlength{\LWR@igwidth}
3531 \newlength{\LWR@igheight}
3532 \newcommand*{\LWR@igwidthstyle}{{}}
3533 \newcommand*{\LWR@igheightstyle}{{}}
3534 \newcommand*{\LWR@igorigin}{{}}
```

---

```

3535 \newcommand*{\LWR@igangle}{}
3536 \newcommand*{\LWR@igxscale}{1}
3537 \newcommand*{\LWR@igyscale}{1}
3538 \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:

```

3539 \define@key{igraph}{width}{%
3540 \setlength{\LWR@igwidth}{#1}%
3541 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3542 {}}
```

Default to use the converted fixed length given:

```

3543 \uselengthunit{PT}%
3544 \renewcommand*{\LWR@igwidthstyle}[width:\rndprintlength{\LWR@igwidth}]%
```

If ex or em dimensions were given, use those instead:

```

3545 \IfEndWith{#1}{ex}%
3546 {\renewcommand*{\LWR@igwidthstyle}[width:#1]}% yes ex
3547 {}% not ex
3548 \IfEndWith{#1}{em}%
3549 {\renewcommand*{\LWR@igwidthstyle}[width:#1]}% yes em
3550 {}% not em
3551 \IfEndWith{#1}{\%}%
3552 {\renewcommand*{\LWR@igwidthstyle}[width:#1]}% yes percent
3553 {}% not percent
3554 \IfEndWith{#1}{px}%
3555 {\renewcommand*{\LWR@igwidthstyle}[width:#1]}% yes px
3556 {}% not px
3557 }{}% end of length > 0pt
3558 }
```

If an optional height was given, set an HTML style:

```

3559 \define@key{igraph}{height}{%
3560 \setlength{\LWR@igheight}{#1}%
3561 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
3562 {}}
```

Default to use the converted fixed length given:

```

3563 \uselengthunit{PT}%
3564 \renewcommand*{\LWR@igheightstyle}[%
3565 height:\rndprintlength{\LWR@igheight} ]%
```

```
3566 }%
```

If ex or em dimensions were given, use those instead:

```
3567 \IfEndWith{#1}{ex}%
3568 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
3569 {}% not ex
3570 \IfEndWith{#1}{em}%
3571 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
3572 {}% not em
3573 \IfEndWith{#1}{\%}%
3574 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
3575 {}% not percent
3576 \IfEndWith{#1}{px}%
3577 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
3578 {}% not px
3579 }{}% end of length > Opt
3580 }
```

Handle origin key:

```
3581 \define@key{igraph}{origin}{%
3582 \renewcommand*{\LWR@igorigin}{#1}%
3583 }
```

Handle angle key:

```
3584 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
3585 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
3586
```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
3587 \define@key{igraph}{scale}{%
3588 \renewcommand*{\LWR@igxscale}{#1}%
3589 \renewcommand*{\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
3590 \define@key{igraph}{bb}(){}
3591 \define@key{igraph}{bbllx}){}
3592 \define@key{igraph}{bbly}){}
3593 \define@key{igraph}{bburx}){}
3594 \define@key{igraph}{bbury}){}
3595 \define@key{igraph}{natwidth}{}{}
```

```

3596 \define@key{igraph}{natheight}{}%
3597 \define@key{igraph}{hiresbb}{}%
3598 \define@key{igraph}{viewport}{}%
3599 \define@key{igraph}{trim}{}%
3600 \define@key{igraph}{totalheight}{}%
3601 \define@key{igraph}{keepaspectratio}{}%
3602 \define@key{igraph}{clip}{}%
3603 \define@key{igraph}{draft}{}%
3604 \define@key{igraph}{type}{}%
3605 \define@key{igraph}{ext}{}%
3606 \define@key{igraph}{read}{}%
3607 \define@key{igraph}{command}{}%

```

**\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.

```

3608 \newcommand*{\LWR@rotstyle}[2]{%
3609   #1transform:rotate(-#2deg);%
3610 }

```

**\LWR@scalestyle** {*prefix*} {*xscale*} {*yscale*}

Prints the scale style with the given prefix.

*prefix* is **-ms-** or **-webkit-** or nothing, and is used to generate three versions of the **transform:scale** style.

```

3611 \newcommand*{\LWR@scalestyle}[3]{%
3612   #1transform:scale(#2,#3);%
3613 }

```

### 58.3 \includegraphics

**Bool LWR@infloatrow** Used to compute \linewidth.

```

3614 \newbool{LWR@infloatrow}%
3615 \boolfalse{LWR@infloatrow}%

3616 \newcommand*{\LWR@imageextension}{}%
3617 \newcommand*{\LWR@expgraphicsfilename}{}%

```

```
\LWR@includegraphicsb  [<options>] {<filename>}

3618
3619 \newcommand*\{\LWR@includegraphicsb}{2} []
3620 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
3621 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\ linewidth`, `\ textwidth`, `\ textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
3622 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
3623 \ifbool{\LWR@infloatrow}{%
3624 {}%
3625 {%
3626 \setlength{\linewidth}{6in}%
3627 \setlength{\textwidth}{6in}%
3628 \setlength{\textheight}{9in}%
3629 }%
3630 }{}}
```

See if can find the image by adding an extension:

Preference is `svgz`, then `svg`, `gif`, `png`, and `jpg`.

`\detokenize\expandafter` allows underscore characters in filenames.

```
3631 \edef\LWR@expgraphicsfilename{#2}
3632 \renewcommand*\{\LWR@imageextension}{()}%
3633 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.jpg}}{%
3634 \renewcommand*\{\LWR@imageextension}{.jpg}{()}%
3635 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.JPG}}{%
3636 \renewcommand*\{\LWR@imageextension}{.JPG}{()}%
3637 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.png}}{%
3638 \renewcommand*\{\LWR@imageextension}{.png}{()}%
3639 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.PNG}}{%
3640 \renewcommand*\{\LWR@imageextension}{.PNG}{()}%
3641 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.gif}}{%
3642 \renewcommand*\{\LWR@imageextension}{.gif}{()}%
3643 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.GIF}}{%
3644 \renewcommand*\{\LWR@imageextension}{.GIF}{()}%
3645 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.svg}}{%
3646 \renewcommand*\{\LWR@imageextension}{.svg}{()}%
3647 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.SVG}}{%
3648 \renewcommand*\{\LWR@imageextension}{.SVG}{()}%
3649 \IfFileExists{\detokenize\expandafter\thisgraphicspath\{\LWR@expgraphicsfilename.svgz}}{%
3650 \renewcommand*\{\LWR@imageextension}{.svgz}{()}}
```

---

```
3651 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.SVGZ}{%
3652 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}}
```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also `\detokenize .pdf` and `.svg` comparison strings.

```
3653 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}{%
3654 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]}%
3655 %
3656 \StrSubstitute{\LWR@strresult}{%
3657 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]}
```

For correct em sizing during the width and height conversions:

```
3658 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
3659 \setlength{\LWR@igwidth}{0pt}%
3660 \setlength{\LWR@igheight}{0pt}%
3661 \renewcommand*{\LWR@igwidthstyle}{}%
3662 \renewcommand*{\LWR@igheightstyle}{}%
3663 \renewcommand*{\LWR@igorigin}{}%
3664 \renewcommand*{\LWR@igangle}{}%
3665 \renewcommand*{\LWR@igxscale}{1}%
3666 \renewcommand*{\LWR@igyscale}{1}%
3667 \renewcommand*{\LWR@igclass}{inlineimage}%
```

Options are in #1

```
3668 \setkeys{igraph}{#1}%
```

Create the HTML reference with the graphicspath, filename, extension, alt tag, style, and class.

The `\LWR@origtilde` adds space between tags in case this is being done inside a `\savebox` where `\newline` has no effect.

```
3669 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}{%
3670 % start of href
3671 \LWR@htmltag% start of image tags
3672 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3673 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline}
```

Only include a style tag if a width, height, angle, or scale was given:

```
3674 \ifthenelse{
```

```

3675 \NOT\equal{\LWR@igwidthstyle}{} \OR
3676 \NOT\equal{\LWR@igheightstyle}{} \OR
3677 \NOT\equal{\LWR@igorigin}{} \OR
3678 \NOT\equal{\LWR@igangle}{} \OR
3679 \NOT\equal{\LWR@igxscale}{1} \OR
3680 \NOT\equal{\LWR@igyscale}{1}
3681 }%
3682 {\LWR@origtilde{} style="%"
3683 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{} }{%
3684 {\LWR@igwidthstyle;}}{%
3685 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{} }{%
3686 {\LWR@igheightstyle;}}{%
3687 \ifthenelse{\NOT\equal{\LWR@igorigin}{} }{%
3688 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{} }{%
3689 \ifthenelse{\NOT\equal{\LWR@igangle}{} }{%
3690 {%
3691 \LWR@rotstyle{-ms-}{\LWR@igangle}%
3692 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
3693 \LWR@rotstyle{}{\LWR@igangle}%
3694 }}{%
3695 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR{%
3696 \NOT\equal{\LWR@igyscale}{1}}}{%
3697 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}}{%
3698 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}}{%
3699 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}}{%
3700 " \LWR@orignewline}{} }{%

```

Set the class:

```

3701 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
3702 }% end of image tags
3703 }% end of href
3704 \endgroup

```

Return to small-sized output:

```

3705 \LWR@origscriptsize
3706 }

```

\includegraphics [⟨key=val⟩] {⟨filename⟩}

Handles width and height, converted to fixed width and heights.

Converts any .pdf references to .svg for HTML

The user should always refer to .pdf in the document source.

```

3707 \renewcommand*\includegraphics{%
3708 }

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

3709 \LWR@ensuredoingapar%
3710 \begingroup%
3711 \LWR@includegraphicsb%
3712 }

3713 \end{warpHTML}

```

**for PRINT output:** For print output, accept and then discard the new `class` key:

```

3714 \begin{warpprint}
3715 \define@key{Gin}{class}{}%
3716 \end{warpprint}

```

## 58.4 \rotatebox, \scalebox, \reflectbox

**for HTML output:** 3717 `\begin{warpHTML}`

`\LWR@rotboxorigin` Holds the origin key letters.

```
3718 \newcommand*{\LWR@rotboxorigin}{}%
```

`\LWR@originname`  $\{\langle letter \rangle\}$

Given one L<sup>A</sup>T<sub>E</sub>X origin key value, translate into an HTML origin word:

```

3719 \newcommand*{\LWR@originname}[1]{%
3720 \ifthenelse{\equal{#1}{t}}{top}{}%
3721 \ifthenelse{\equal{#1}{b}}{bottom}{}%
3722 \ifthenelse{\equal{#1}{c}}{center}{}%
3723 \ifthenelse{\equal{#1}{l}}{left}{}%
3724 \ifthenelse{\equal{#1}{r}}{right}{}%
3725 }

```

`\LWR@originnames`  $\{\langle letters \rangle\}$

Given one- or two-letter L<sup>A</sup>T<sub>E</sub>X origin key values, translate into HTML origin words:

```

3726 \newcommand*{\LWR@originnames}[1]{%
3727 \StrChar{#1}{1}[\LWR@strresult]%
3728 \LWR@originname{\LWR@strresult}%
3729 \StrChar{#1}{2}[\LWR@strresult]%
3730 \LWR@originname{\LWR@strresult}%
3731 }

```

Handle the origin key for `\rotatebox`:

```
3732 \define@key{krotbox}{origin}{%
3733 \renewcommand*\{\LWR@rotboxorigin\}{#1}%
3734 }
```

These keys are ignored:

```
3735 \define@key{krotbox}{x}{}
3736 \define@key{krotbox}{y}{}
3737 \define@key{krotbox}{units}{}
```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
3738 \NewDocumentCommand{\LWR@rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
3739 \renewcommand*\{\LWR@rotboxorigin\}{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
3740 \setkeys{krotbox}{#1}{}
```

Select `inline-block` so that HTML will transform this span:

```
3741 \LWR@htmllagc{span style="display: inline-block;}
```

If an origin was given, translate and print the origin information:

```
3742 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
3743 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}
```

Print the rotation information:

```
3744 \LWR@rotstyle{-ms-}{#2} %
3745 \LWR@rotstyle{-webkit-}{#2} %
3746 \LWR@rotstyle{}{#2} %
3747 "{}"\LWR@orignewline%
```

Print the text to be rotated:

```
3748 \begin{LWR@nestspan}%
3749 #3%
```

Close the span:

```
3750 \LWR@htmntagc{/span}%
3751 \end{LWR@nestspan}%
3752 }
```

```
\scalebox {<h-scale>} [<v-scale>] {<text>}
```

Will \let\scalebox\LWR@scalebox at \LWR@LwarpStart, in case \scalebox was over-written by a later package load.

```
3753 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
3754 \LWR@htmntagc{span style="display: inline-block; %
```

Print the scaling information:

```
3755 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3756 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3757 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3758 "{}" %
```

Print the text to be scaled:

```
3759 \begin{LWR@nestspan}%
3760 #3%
```

Close the span:

```
3761 \LWR@htmntagc{/span}%
3762 \end{LWR@nestspan}%
3763 }
```

```
\reflectbox {<text>}
```

Will \let\reflectbox\LWR@reflectbox at \LWR@LwarpStart, in case \reflectbox was over-written by a later package load.

```
3764 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
```

```
3765 \end{warpHTML}
```

## 58.5 Null functions

These functions are not supported by lwarf's HTML conversion.

**for HTML output:** 3766 \begin{warpHTML}

```
\resizebox {\langle h-length\rangle} {\langle v-length\rangle} {\langle text\rangle}
```

Simply prints its text argument.

3767 \renewcommand{\resizebox}[3]{#3}

3768 \end{warpHTML}

## 59 Cleverref

Pkg **cleveref** cleveref package is used as-is with minor patches.

**loading order** cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via \AtEndPreamble and \AfterEndPreamble. This is done because the HTML conversion requires cleveref. The user's document may not require cleveref, thus the user may never explicitly load it, so during HTML output l warp loads it last. If the user's document preamble uses cleveref options, or functions such as \crefname, then cleveref may be loaded in the user's preamble near the end, and l warp's additional loading of cleveref will have no effect.

Table 8 on 189 shows the data structure of the label/reference system as revised by l warp and cleveref.

A few patches allow cleveref to work as-is:

**for HTML output:** 3769 \begin{warpHTML}

\AtEndPreamble forces cleveref to be loaded last:

```
3770 \AtEndPreamble{  
3771 \RequirePackage{cleveref}  
3772 \crefname{pagenote}{note}{notes}  
3773 }
```

The following patches are applied after cleveref has loaded, and after \AtBeginDocument:

3774 \AfterEndPreamble{

```
\@setcref {\langle kindofref\rangle} {\langle label\rangle}
```

3775 \renewcommand\*\{@setcref}[2]{#1{\ref{#2}}{}{}}

---

```
\@@setcrefrange {<text>} {<label>} {<label>}

3776 \renewcommand{\@@setcrefrange}[3]{%
3777 #1{\ref{#2}}{\ref{#3}}{}{}{}{}}
```

**\cpagerefFor** Redefinable word between “page(s)” and the page numbers.

```
3778 \newcommand*{\cpagerefFor}{for}
```

**\@@setcpageref** {<typeofref>} {<label>}, where typeofref is “page” or “pages”

```
3779 \renewcommand*{\@@setcpageref}[2]{%
3780 #1{\cpagerefFor\ \cref{#2}}{}{}%}
3781 }

3782 \renewcommand{\@@setcpagerefrange}[3]{%
3783 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}}
```

Remember and patch some label-related defintions.

```
3784 \let\LWR@origlabel\label
3785 \let\label\LWR@newlabel
3786 \let\LWR@origref\ref
3787 \let\ref\LWR@newref% \end{ syntax highlighting
3788 \let\LWR@origpageref\pageref
3789 \let\pageref\LWR@newpageref
3790
3791
3792 }% AfterEndPreamble
3793
3794 \end{warpHTML}
```

## 60 Picture

**Env picture** The **picture** environment is enclosed inside a **\lateximage**.

**for HTML output:** 3795 \begin{warpHTML}

```
Env picture

3796 \BeforeBeginEnvironment{picture}%
3797
3798 \lateximage%
3799
```

---

```

3800 \AfterEndEnvironment{picture}{\endlateximage}
3801
3802 }

3803 \end{warpHTML}
```

## 61 Tikzpicture

Pkg **tikz**

Env **tikzpicture** May be used as-is, and its contents will be converted to an image.

⚠ **preamble** For recent versions of **Tikz** (v3.0.0 — 2013/12/20 and later), include in the document preamble:

```
\usetikzlibrary{babel} % v3.0.0 and up
```

For older versions of **Tikz** without the **Tikz babel** library, **lwarp** automatically handles the catcode change for the \$ math shift.

**catcodes** (**lwarp** changes the catcode of \$ for its own use. The **Tikz babel** library temporarily changes catcodes back to normal for **Tikz**'s use, restoring them to their **lwarp** purpose while inside each **Tikz** node.)

**for HTML output:** 3804 \begin{warpHTML}

Bool **LWR@tikzbabel** **tikz** v3.0.0 introduced the **babel** library which handles catcode changes. For older versions, **lwarp** must change \$'s catcode itself.

```

3805 \newbool{LWR@tikzbabel}
3806 \boolfalse{LWR@tikzbabel}
3807
3808 \AtBeginDocument{
3809 \@ifpackageloaded{tikz}{
3810 \@ifpackagelater{tikz}{2013/12/20}%
3811 {\booltrue{LWR@tikzbabel}}
3812 {\boolfalse{LWR@tikzbabel}}
3813 }{}}
3814 }
```

Env **tikzpicture** **tikzpicture** environment is enclosed inside a **\lateximage**.

```

3815 \BeforeBeginEnvironment{tikzpicture}{%
3816
3817 \lateximage
3818 \ifbool{LWR@tikzbabel}{}
3819 {}}
```

---

```

3820 {\catcode`\$=3} % dollar sign is math shift
3821 }
3822
3823 \AfterEndEnvironment{tikzpicture}{%
3824 \endlateximage
3825
3826 \ifbool{LWR@tikzbabel}{}
3827 {}
3828 {\catcode`\$=\active}
3829 }

3830 \end{warpHTML}

```

## 62 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

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

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

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

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

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

**for HTML output:** 3831 `\begin{warpHTML}`

### 62.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

---

```
3832 \newcounter{LWR@minipagedepth}
3833 \setcounter{LWR@minipagedepth}{0}
```

Len \WR@minipagewidth Used to convert the width into printable units.

```
3834 \newlength{\LWR@minipagewidth}
```

Len \WR@minipageheight Used to convert the height into printable units.

```
3835 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
3836 \let\LWR@origminipage\minipage
3837 \let\LWR@origendminipage\endminipage
```

## 62.2 Footnote handling

Also see section 35 for other forms of footnotes.

```
\LWR@nullrule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Used to cancel the footnotesep rule during HTML output. This rule was causing occasional end paragraph tags to be inserted for some (so far) undetermined reason, as well as adding purposeless code to the HTML output.

```
3838 \NewDocumentCommand{\LWR@nullrule}{o m m}{}%
```

```
\LWR@makefntext {⟨text⟩}
```

Adds a paragraph closing tag to the end of each footnote.

```
3839 \newcommand{\LWR@makefntext}[1]{%
3840 \begingroup%
3841 \let\rule\LWR@nullrule%
3842 \textsuperscript{\@thefnmark} #1%
3843 \endgroup%
3844 \LWR@htmltagc{/LWR@tagregularparagraph}%
3845 }
```

```
\LWR@makefnmark Creates an HTML superscript footnote mark:
```

```
3846 \newcommand*{\LWR@makefnmark}{\textsuperscript{\@thefnmark}}
```

### 62.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L<sup>A</sup>T<sub>E</sub>X definition and modified to create a `mpfootnotes` div class:

```

3847 \def\endminipage{%
3848   \par
3849   \unskip
3850   \ifvoid\@mpfootins\else
3851     \vskip\skip\@mpfootins
3852     \normalcolor
3853 \LWR@htmldivclass{mpfootnotes}
3854   \unvbox\@mpfootins
3855 \LWR@htmldivclassend{mpfootnotes}
3856   \fi
3857   \minipagetrue
3858 \color@endgroup
3859 \egroup
3860 \expandafter\iiiparbox\mpargs{\unvbox\tempboxa}}
```

`\LWR@subminipage` Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```

3861 \newcommand*\subminipage{%
3862 \LWR@stopars
3863 \LWR@origminipage{6in}}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3864 \LWR@origraggedright%
```

Minipage footnotes appear the bottom of the minipage.

```

3865 \let\footnote\LWR@origfootnote%
3866 \let\makefntext\LWR@makefntext%
3867 \let\makefnmark\LWR@makefnmark%
```

Resume paragraph tag handling for the contents of the minipage:

```

3868 \LWR@startpars%
3869 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
3870 \newcommand*\endsubminipage{%
```

---

```

3871 \LWR@stoppars%
3872 \LWR@endminipage% The following empty line is required:
3873
3874 }
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```

3875 \newbool{LWR@minipagefullwidth}
3876 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

**for HTML output:** 3877 \newcommand\*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
 3878 \end{warpHTML}

**for PRINT output:** 3879 \begin{warpprint}
 3880 \newcommand\*{\minipagefullwidth}{}%
 3881 \end{warpprint}

**for HTML output:** 3882 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```

3883 \newbool{LWR@minipagethispar}
3884 \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  $6 \times 9$  inch text area.

```

3885 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}
3886 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```

3887 \LWR@traceinfo{starting minipage}%
3888 \uselengthunit{in}%
3889 \setlength{\LWR@minipagewidth}{#4}%

```

```

3890 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3891 \addtolength{\LWR@minipagewidth}{3em}%
3892 \setlength{\linewidth}{6in}%
3893 \setlength{\textwidth}{6in}%
3894 \setlength{\textheight}{9in}%
3895 }{}%
3896 \addtocounter{LWR@minipagedepth}{1}%
3897 \setlength{\LWR@minipageheight}{\textheight}%
3898 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%

```

$\text{\LaTeX}$  wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
3899 \LWR@stopars%
```

Create the `<div>` tag with optional alignment style:

```

3900 \LWR@traceinfo{minipage: creating div class}%
3901 \LWR@htmntag{div class="minipage" style="%"
3902 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
3903 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
3904 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
3905 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
3906 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
3907 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
3908 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%

```

Print the width and optional height styles:

```

3909 \LWR@traceinfo{minipage: about to print the width}%
3910 \uselengthunit{PT}%
3911 \ifbool{LWR@minipagefullwidth}{}%
3912 {\boolfalse{LWR@minipagefullwidth}}%
3913 {}%
3914 \ifthenelse{\lengthtest{#4}=\linewidth}{%
3915 {}%
3916 {width:\rndprintlength{\LWR@minipagewidth} ; }%
3917 }%
3918 \LWR@traceinfo{minipage: about to print the height}%
3919 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
3920 "{}"%

```

Finish with an empty line to start  $\text{\LaTeX}$  minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

3921
3922 \LWR@origminipage{6in}%
The preceding empty line is required.

```

Set the user-accessible minipage and text width and height values inside the minipage. These do not affect the actual size of the large minipage created by `\LWR@origminipage` above, but are used by any reference to `\linewidth`, etc. inside the PDF minipage being created here.

```
3923 \setlength{\linewidth}{#4}%
3924 \setlength{\textwidth}{6in}%
3925 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3926 \LWR@origraggedright%
3927 \let\footnote\LWR@origfootnote%
3928 \let\@makefntext\LWR@makefntext%
3929 \let\@makefnmark\LWR@makefnmark%
```

Resume paragraph tag handling for the contents of the minipage:

```
3930 \LWR@startpars%
3931 \LWR@traceinfo{minipage: finished starting the minipage}%
3932 }
```

End the environment with L<sup>A</sup>T<sub>E</sub>X processing and closing tag:

```
3933 {%
3934 \LWR@stoppars%
3935 \LWR@endminipage% The following empty line is required:
3936
3937 \LWR@htmldivclassend{minipage}%
3938 \vspace{1\baselineskip}% required for subcaption
3939 \addtocounter{\LWR@minipagedepth}{-1}%
3940 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
3941 \global\booltrue{\LWR@minipagethispar}%
3942 }
```

## 62.4 Parbox, makebox, framebox, fbox, raisebox

```
\parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}
```

A parbox uses the minipage code:

```
3943 \RenewDocumentCommand{\parbox}{O{t} O{t} m m}
```

---

```

3944 {
3945 \begin{minipage}[#1][#2][#3]{#4}
3946 #5
3947 \end{minipage}
3948 }
```

`\makebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}`

Width and position are ignored.

```

3949 \RenewDocumentCommand{\makebox}{o o m}{%
3950 \mbox{#3}%
3951 }
```

`\framebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}`

Width and position are ignored.

```

3952 \RenewDocumentCommand{\framebox}{o o m}{%
3953 \fbox{#3}%
3954 }
```

`\fbox {⟨text⟩}`

```

3955 \let\LWR@orig\fbox\fbox
3956 %
3957 \renewcommand*{\fbox}[1]{%
3958 \InlineClass{framebox}{#1}%
3959 }
```

`\raisebox {⟨raiselen⟩} [⟨height⟩] [⟨depth⟩] {⟨text⟩}`

```

3960 \RenewDocumentCommand{\raisebox}{m o o m}{%
3961 #4%
3962 }
```

```
3963 \end{warpHTML}
```

## 63 Direct formatting

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

For high-level block and inline custom CSS classes, see section [29.7](#).

**for HTML output:** 3964 `\begin{warpHTML}`

```
\emph  {\langle text\rangle}

3965 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}


\textmd  {\langle text\rangle}

3966 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}


\textbf  {\langle text\rangle}

3967 \renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}


\textrm  {\langle text\rangle}

3968 \renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}


\textsf  {\langle text\rangle}

3969 \renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}


\texttt  {\langle text\rangle}

3970 \renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}


\textup  {\langle text\rangle}

3971 \renewcommand{\textup}[1]{\LWR@htmlspan{textup}{#1}}


\textit  {\langle text\rangle}

3972 \renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}


\textsc  {\langle text\rangle}

3973 \renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}


\textnormal  {\langle text\rangle}

3974 \renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}} 


\mdseries

3975 \renewcommand*{\mdseries}{} 
```

\bfseries  
3976 \renewcommand\*\{\bfseries\}{}

\rmfamily  
3977 \renewcommand\*\{\rmfamily\}{}

\sffamily  
3978 \renewcommand\*\{\sffamily\}{}

\ttfamily  
3979 \renewcommand\*\{\ttfamily\}{}

\upshape  
3980 \renewcommand\*\{\upshape\}{}

\itshape  
3981 \renewcommand\*\{\itshape\}{}

\scshape  
3982 \renewcommand\*\{\scshape\}{}

\normalfont  
3983 \renewcommand\*\{\normalfont\}{}

\sp {*text*}  
For siunitx. Must work in math mode  
3984 \renewcommand{\sp}[1]{\text{<}}

\sb {*text*}  
For siunitx. Must work in math mode  
3985 \renewcommand{\sb}[1]{\text{<}}

```
\textsuperscript {⟨text⟩}
```

```
3986 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

```
\textsubscript {⟨text⟩}
```

```
3987 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
```

**\up** {⟨text⟩} Prints superscript.

This is \let at the beginning of the document in case some other package has changed the definition.

```
3988 \AtBeginDocument{\let\up\textsuperscript}
```

**\fup** {⟨text⟩} Prints superscript.

Supports fmtcount package.

This is \let at the beginning of the document in case some other package has changed the definition.

```
3989 \AtBeginDocument{\let\fup\textsuperscript}
```

**\hfill**

```
3990 \renewcommand*{\hfill}{\qquad}
```

**\rulefill**

```
3991 \renewcommand*{\rulefill}{\rule{1in}{1pt}}
```

**\dotfill**

```
3992 \renewcommand*{\dotfill}{\dots}
```

```
3993 \end{warpHTML}
```

## 64 Skips, spaces, font sizes

**for HTML output:** 3994 \begin{warpHTML}

\, must be redefined after \RequirePackage{printlen}

```
3995 \let\LWR@origcomma\,
3996 \let\LWR@origtilde\
3997 \let\LWR@origskip\enskip
3998 \let\LWR@origquad\quad
3999 \let\LWR@origqquad\qquad
4000 \let\LWR@orighspace\hspace
4001 \let\LWR@origrule\rule
```

Direct-formatting space commands become HTML entities:

```
4002 \renewcommand*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
4003 \renewcommand*{~}{\HTMLentity{nnbsp}}
4004 \renewcommand*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
4005 \let\LWR@orignormalsize\normalsize
4006 \let\LWR@origsmall\small
4007 \let\LWR@origfootnotesize\footnotesize
4008 \let\LWR@origscriptsize\scriptsize
4009 \let\LWR@origtiny\tiny
4010 \let\LWR@origlarge\large
4011 \let\LWR@origLarge\Large
4012 \let\LWR@origLARGE\LARGE
4013 \let\LWR@orighuge\huge
4014 \let\LWR@origHuge\Huge
4015 \renewcommand*{\normalsize}={}
4016 \renewcommand*{\small}={}
4017 \renewcommand*{\footnotesize}={}
4018 \renewcommand*{\scriptsize}={}
4019 \renewcommand*{\tiny}={}
4020 \renewcommand*{\large}={}
4021 \renewcommand*{\Large}={}
4022 \renewcommand*{\LARGE}={}
4023 \renewcommand*{\huge}={}
4024 \renewcommand*{\Huge}={}
4025
4026 \renewcommand*{\onecolumn}={}
4027
4028 \renewcommand{\twocolumn}[1][]{
4029
4030 #1
4031
4032 }
```

\newline Uses HTML <br /> tag

```
4033 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
4034 \let\newline\LWR@newlinebr
```

\\" Redefined to \LWR@endofline or \LWR@tabularendofline.

\LWR@endofline \* [<len>]

\\" is assigned to \LWR@endofline at \LWR@LwarpStart.

Inside tabular, \\ is temporarily changed to \LWR@tabularendofline.

```
4035 \NewDocumentCommand{\LWR@endofline}{s o}%
4036 {%
4037 \newline%
4038 }
```

\LWR@minipagestartpars Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a div to be inside a p, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, lwarp correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. lwarp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around \hspace, \enskip, \quad, and \qquad until the end of the paragraph, when the closing p tag is created.

When a minipage is seen, the boolean LWR@minipagethispar is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. LWR@minipagethispar is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before \hspace, \quad, or \qquad's HTML output.

```
4039 \newcommand*{\LWR@minipagestartpars}{%
4040 \ifbool{LWR@minipagethispar}{%
4041 {%
4042 \LWR@startpars{%
4043 }{}}%
4044 }}
```

\LWR@minipagestoppars Placed just after \hspace, \quad, or \qquad's HTML output.

```

4045 \newcommand*{\LWR@minipagestoppars}{%
4046 \ifbool{LWR@minipagethispar}{%
4047 {%
4048 \LWR@stoppars%
4049 }{}}%
4050 }

```

**\quad** Handles special minipage & horizontal space interactions.

```

4051 \renewcommand*{\quad}{%
4052 \LWR@minipagestoppars%
4053 \HTMLunicode{2001}%
4054 \LWR@minipagestartpars%
4055 }

```

**\quad** Handles special minipage & horizontal space interactions.

```
4056 \renewcommand*{\quad}{\quad\quad}
```

**\enskip** Handles special minipage & horizontal space interactions.

```

4057 \renewcommand*{\enskip}{%
4058 \LWR@minipagestoppars%
4059 \HTMLunicode{2000}%
4060 \LWR@minipagestartpars%
4061 }

```

**Len \WR@tempwidth** Used to compute span width, height, raise for **\hspace** and **\rule**:

```

Len \WR@tempheight 4062 \newlength{\LWR@tempwidth}
Len \WR@tempraise 4063 \newlength{\LWR@tempheight}
4064 \newlength{\LWR@tempraise}

```

**\LWR@hspace \* {<length>}**

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

**\fill** is zero-width, so **\hspace{\fill}** is ignored.

```

4065
4066 \NewDocumentCommand{\LWR@hspace}{s m}{%
4067 \setlength{\LWR@tempwidth}{#2}}

```

Only if the width is not zero:

```
4068 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4069 \LWR@minipagestopars%
```

Support the HTML thin wrappable space:

```
4070 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}%
4071 {%
4072 \HTMLunicode{2009} thin breakable space
4073 }%
```

Print the span with the converted width. Not rounded.

```
4074 {%
4075 \uselengthunit{PT}%
4076 \LWR@htmltagc{%
4077 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
4078 }%
4079 \LWR@htmltagc{/span}%
4080 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4081 \LWR@minipagestartpars%
4082 }%
4083 }
```

\hspace \* {*length*}

Handles special minipage & horizontal space interactions.

```
4084 \let\hspace\LWR@hspace
```

\linebreak [*num*]      Inserts an HTML br tag.

```
4085 \renewcommand*\linebreak[1][]{\newline}
```

\nolinebreak [*num*]

```
4086 \renewcommand*\nolinebreak[1][]{}
```

\pagebreak [*num*]      Starts a new paragraph.

```
4087 \renewcommand*\pagebreak[1][]{%
4088 %
4089 }
```

---

```
\nopagebreak  [<num>]
4090 \renewcommand*{\nopagebreak}[1] [] {}

\enlargethispage * {<len>}
4091 \RenewDocumentCommand{\enlargethispage}{s m} {}

\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.

4092 \NewDocumentCommand{\LWR@rule}{o m m}{%
The width is copied into a temporary LATEX length, from which comparisons and
conversions may be made:

4093 \setlength{\LWR@tempwidth}{#2}%
If it's zero-width then skip the entire rule:

4094 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
4095 {}% zero- width
4096 {}% non-zero width

If it's non-zero width, set a minimal thickness so that it more reliably shows in the
browser:

4097 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
4098 \lengthtest{\LWR@tempwidth<1pt}}%
4099 {\setlength{\LWR@tempwidth}{1pt}}{}%

Likewise with height:

4100 \setlength{\LWR@tempheight}{#3}%
4101 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
4102 \lengthtest{\LWR@tempheight<1pt}}%
4103 {\setlength{\LWR@tempheight}{1pt}}{}%

If had a minipage this paragraph, try to inline the rule without generating paragraph
tags:

4104 \LWR@minipagestopars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in L<sup>A</sup>T<sub>E</sub>X code.

```
4105 \uselengthunit{PT}%
4106 \LWR@htmltagc{%
4107 span
4108 style=" %
```

The background color is used to draw the filled rule. The color may be changed by \textcolor.

```
4109 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
4110 width:\printlength{\LWR@tempwidth}; %
4111 height:\printlength{\LWR@tempheight}; %
```

The raise height is converted to a CSS transform. The \*2 raise multiplier is to approximately match HTML output's X height. Conversion to a L<sup>A</sup>T<sub>E</sub>X length allows a typical L<sup>A</sup>T<sub>E</sub>X expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L<sup>A</sup>T<sub>E</sub>X length limits the allowable syntax. To do: A superior method would compute a ratio of L<sup>A</sup>T<sub>E</sub>X ex height, then print that to HTML with an ex unit.

```
4112 \IfValueTF{#1}%
4113 {%
4114 \setlength{\LWR@tempraise}{\printlength{\LWR@tempwidth}}%
4115 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4116 \LWR@orignewline%
4117 -ms-transform: translate(\printlength{\LWR@tempraise}); %
4118 \LWR@orignewline%
4119 -webkit-transform: translate(\printlength{\LWR@tempraise}); %
4120 \LWR@orignewline%
4121 transform: translate(\printlength{\LWR@tempraise}); %
4122 \LWR@orignewline%
4123 }{}}%
```

Display inline-block to place the span inline with the text:

```
4124 display:inline-block;"%
4125 }%
4126 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4127 \LWR@minipagestartpars%
```

```

4128 }% non-zero width
4129 }

\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}

Handles special minipage & horizontal space interactions.

4130 \let\rule\LWR@rule

4131 \end{warpHTML}

```

## 65 \phantomsection

**for HTML output:** 4132 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```

4133 \newcommand*\phantomsection{\section*{}}

4134 \end{warpHTML}

```

## 66 \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>

### 66.1 HTML logos

**for HTML output:** 4135 \begin{warpHTML}

\TeX \TeX

`latexlogo` is a CSS class used to properly typeset the E and A in L<sup>A</sup>T<sub>E</sub>X and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in L<sup>A</sup>T<sub>E</sub>X, L<sup>A</sup>u<sub>T</sub>E<sub>X</sub>, Con<sub>T</sub>E<sub>X</sub>t, etc.

```
4136 \newcommand*{\LWR@TeX}{%
4137 {\InlineClass{latexlogofont}{%
4138 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}}
```

\LaTeX L<sup>A</sup>T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

```
\LaTeXe
4139 \newcommand*{\LWR@LaTeX}{%
4140 {\InlineClass{latexlogofont}{%
4141 {\InlineClass{latexlogo}{%
4142 {L\textsuperscript{a}T\textsubscript{e}X}}}}}
4143
4144 \renewcommand*{\LaTeXe}{%
4145 {\LaTeX\InlineClass{latexlogofont}{%
4146 {\textit{\text{,2\textsubscript{\textit{\text{HTMLunicode{3B5}}}}}}}}}
```

\LuaTeX LuaL<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X

```
\LuaLaTeX
4147 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
4148 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX X<sub>F</sub>T<sub>E</sub>X, X<sub>F</sub>L<sup>A</sup>T<sub>E</sub>X

\XeLaTeX xetexlogo is a CSS class which aligns the backwards E in X<sub>F</sub>T<sub>E</sub>X and spaces T<sub>E</sub>X appropriately.

xelatexlogo is a CSS class which aligns the backwards E in X<sub>F</sub>L<sup>A</sup>T<sub>E</sub>X and spaces L<sup>A</sup>T<sub>E</sub>X appropriately.

```
4149 \newcommand*{\Xe}{%
4150 {\textit{\text{X\textsubscript{F}\text{TeX}}}{\textit{\text{HTMLunicode{18e}}}}}}
4151 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
4152 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt Con<sub>T</sub>E<sub>X</sub>t

```
4153 \newcommand*{\LWR@ConTeXt}{%
4154 {\InlineClass{latexlogofont}{Con}\TeX{}}}
4155 \InlineClass{latexlogofont}{t}
```

\BibTeX BIBT<sub>E</sub>X, *MakeIndex*

```
\MakeIndex
4156 \providecommand*{\BibTeX}{%
4157 {\InlineClass{latexlogofont}{B\textit{\text{texsc{ib}}}}}\TeX}
```

```
4158
4159 \newcommand*{\MakeIndex}
4160 {\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.

```
4161 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
4162 {\InlineClass{amslogo}{\textit{A}\textsubscript{M}\textsubscript{S}}}}
```

### \MiKTeX MiKTeX

```
4163 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\text{\TeX{}}}
```

### \LyX LyX

`lyxlogo` is a CSS class used for the LyX logo.

```
4164 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
4165 \end{warpHTML}
```

## 66.2 Print logos

**for PRINT output:**

```
4166 \begin{warpprint}
4167 \newcommand*{\XeTeXrevE}
4168 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
4169 \providecommand*{\XeTeX}{\mbox{X}\XeTeXrevE\TeX{}}
4170 \providecommand*{\XeLaTeX}{\mbox{X}\XeTeXrevE\LaTeX{}}
4171 \providecommand*{\AmS}{%
4172 \leavevmode\hbox{$\mathcal{A}$}\kern-.2em\lower.376ex%
4173 \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}{}}
4174 \newcommand*{\LyX}{\textsf{LyX}}
4175 \providecommand*{\LuaTeX}{\mbox{Lua\TeX{}}}
4176 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX{}}}
4177 \providecommand*{\BibTeX}{\mbox{B\textit{textsc{ib}}\TeX{}}}
4178 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
4179 \providecommand*{\ConTeXt}{\mbox{Con\TeXt{}}}
4180 \providecommand*{\MiKTeX}{\mbox{MiK\TeX{}}}
4181 \end{warpprint}
```

## 67 \AtBeginDocument, \AtEndDocument

**for HTML output:** 4182 \begin{warpHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the  
\LWR@LwarpEnd document.

4183 \AfterEndPreamble{\LWR@LwarpStart}  
4184 \AtEndDocument{\LWR@LwarpEnd}

4185 \end{warpHTML}

## 68 lwarpmk.conf

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

An example `lwarpmk.conf`:

---

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

---

Defaults unless already over-ridden by the user:

```
4186 \providecommand*\{\HomeHTMLFileName\}{\BaseJobname}
4187 \providecommand*\{\HTMLFileName\}{}
4188 \providecommand*\{\UseLatexmk\}{false}
```

**for PRINT output:**

```
4189 \begin{warpprint}
4190 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
4191 \immediate\openout\LWR@file=lwarpmk.conf
4192 \ifwindows
4193 \immediate\write{\LWR@file}{opsystem = "Windows"}
4194 \else
4195 \immediate\write{\LWR@file}{opsystem = "Unix"}
4196 \fi
4197 \ifPDFTeX
4198 \immediate\write{\LWR@file}{latexname = "pdflatex"}
4199 \fi
4200 \ifXeTeX
4201 \immediate\write{\LWR@file}{latexname = "xelatex"}
4202 \fi
4203 \ifLuaTeX
4204 \immediate\write{\LWR@file}{latexname = "lualatex"}
4205 \fi
4206 \immediate\write{\LWR@file}{sourcename = "\jobname"}
4207 \immediate\write{\LWR@file}{%
4208 homehtmlfilename = "\HomeHTMLFileName"%
4209 }
4210 \immediate\write{\LWR@file}{htmlfilename = "\HTMLFileName"}
4211 \immediate\write{\LWR@file}{uselatexmk = "\UseLatexmk"}
4212 \immediate\closeout{\LWR@file}
4213 \end{warpprint}
```

## 69 Trademarks

- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

## Package 2

# **lwarf-newproject.sty**

## 70 lwarf-newproject

Pkg **lwarf-newproject** lwarf-newproject is used to create lwarf-specific system files in a new project.

Opt **lwarf-newproject warpprint** Configuration files are only created if the **warpprint** option was used. When this is the case, **lwarf-newproject** knows that **\jobname** is the source code's filename (as opposed to the **<project>.html.tex** filename).

Opt **lwarf-newproject warpHTML** Does not create configuration files. **warpprint** or **warpHTML** are passed to both **lwarf** and **lwarf-newproject** by **lwarpmk**, depending on whether **lwarpmk print** or **lwarpmk html** was used.

Opt **lwarf-newproject BaseJobName** **BaseJobname=somename** sets **\BaseJobname** to **somename**, which is used to set **\HomeHTMLFileName**, which is then written to the **lwarpmk.conf** and **<project>.lwarpmkconf** configuration files.

**\BaseJobname** is also written to **<project>.html.tex** as an option given for the **lwarf** and **lwarf-newproject** packages during the creation of HTML output.

**\BaseJobname** is the **\jobname** of the printed version, even while compiling the HTML version, where **\jobname** has **\_html** appended.

Opt **lwarf-newproject lwarpmk** Tells **lwarf-newproject** 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**.

In the document source:

---

```
\documentclass{article} % or book, report
...
(font selection, input encoding)
...
\newcommand*{\HomeHTMLFileName}{index}, or {projectname}
\newcommand*{\HTMLFileName}{}
\usepackage{lwarf-newproject}
\usepackage{lwarf}
...
(load other packages, the rest of the preamble)
...
\NewCSS{project.css}
\NewHTMLdescription{Summarize this webpage.}
```

---

```
...
\begin{document}
...
\end{document}
```

---

Place `\usepackage{l warp-newproject}` just before `\usepackage{l warp}` in the document source. The operating system and TeX engine will be auto-detected, and a customized `l warpmk.conf` file will be generated, along with the other support files (`*.css`, `l warp_html.xdy`, `l warp_mathjax.txt`). Rename `sample_project.css` to your own custom `projectname.css` and edit it if desired.

After the first compile, `\usepackage{l warp-newproject}` may be commented out unless the configuration changes. For example, it may be re-enabled to switch between `pdflatex`, `xelatex`, and `lualatex`, in which case a new manual compile (without using `l warpmk`) will auto-configure and regenerate the files. It may also be left enabled at all times, in which case the configuration files will be regenerated each time. Note that your own `projectname.css` file will not be over-written by the newly regenerated `sample_project.css`.

Discard all options for `l warp-newproject`:

```
1 \ProvidesPackage{l warp-newproject}
2
3 \RequirePackage{etoolbox}
4 \RequirePackage{comment}
5 \RequirePackage{fancyvrb}
6 \RequirePackage{ifplatform}%
7 \RequirePackage{iftex}%
8 \RequirePackage{kvoptions}
9 \SetupKeyvalOptions{family=LWRNP,prefix=LWRNP@}
```

Generate config files?

```
10 \newbool{LWRNP@genconf}
11 \boolfalse{LWRNP@genconf}
```

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

```
12 \excludecomment{LWR@createlwarpmk}
```

Process options:

```
13 \DeclareVoidOption{warpprint}{\booltrue{LWRNP@genconf}}
14 \DeclareVoidOption{warpHTML}{\boolfalse{LWRNP@genconf}}
15 \DeclareVoidOption{l warpmk}{\includecomment{LWR@createlwarpmk}}
16 \DeclareStringOption[\jobname]{BaseJobname}
17
```

```

18 \booltrue{LWRNP@genconf}%
19 \ProcessKeyvalOptions*\relax
20

```

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

```
21 \providecommand*{\BaseJobname}{\LWRNP@BaseJobname}
```

Defaults unless already over-ridden by the user:

```

22 % \ifbool{LWRNP@genconf}%
23 \providecommand*{\HomeHTMLFileName}{\BaseJobname}%
24 % }{}%
25 \providecommand*{\HTMLFileName}{}%
26 \providecommand*{\UseLatexmk}{false}

```

## 70.1 project\_html.tex

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

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

```

27 \ifbool{LWRNP@genconf}%
28 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}%
29 \immediate\openout\LWR@file=\jobname_html.tex
30 \immediate\write{\LWR@file}%
31 \detokenize{\PassOptionsToPackage}%
32 {warpHTML,BaseJobname=\jobname}{lwarf}%
33 }%
34 \immediate\write{\LWR@file}%
35 \detokenize{\PassOptionsToPackage}%
36 {warpHTML,BaseJobname=\jobname}{lwarf-newproject}%
37 }%
38 \immediate\write{\LWR@file}%
39 \detokenize{\input}\string{\jobname.tex}\string }%
40 }%
41 \immediate\closeout{\LWR@file}
42 }{}%

```

## 70.2 project.lwarpmkconf

File `project.lwarpmkconf` The configuration file for `lwarpmk`.

```

43 \ifbool{LWRNP@genconf}%
44 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}%

```

```
45 \immediate\openout\LWR@file=\jobname.lwarpmkconf
46 \ifwindows
47 \immediate\write\LWR@file{opsystem = "Windows"}
48 \else
49 \immediate\write\LWR@file{opsystem = "Unix"}
50 \fi
51 \ifPDFTeX
52 \immediate\write\LWR@file{latexname = "pdflatex"}
53 \fi
54 \ifXeTeX
55 \immediate\write\LWR@file{latexname = "xelatex"}
56 \fi
57 \ifLuaTeX
58 \immediate\write\LWR@file{latexname = "lualatex"}
59 \fi
60 \immediate\write\LWR@file{sourcename = "\jobname"}
61 \immediate\write\LWR@file{%
62 homehtmlfilename = "\HomeHTMLFileName"%
63 }
64 \immediate\write\LWR@file{htmlfilename = "\HTMLFileName"}
65 \immediate\write\LWR@file{uselatexmk = "\UseLatexmk"}
66 \immediate\closeout\LWR@file
67 }{}
```

### 70.3 l warp.css

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

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

```
68 \begin{VerbatimOut}{l warp.css}
69 /*
70   CSS stylesheet for the LaTeX l warp package
71   Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
72 */
73
74
75 /* a fix for older browsers: */
76 header, section, footer, aside, nav, main,
77     article, figure { display: block; }
78
79
80 A:link {color:#000080 ; text-decoration: none ; }
81 A:visited {color:#800000 ; }
82 A:hover {color:#000080 ; text-decoration: underline ;}
83 A:active {color:#800000 ; }
```

```
84
85 a.tocpart {display: inline-block ; margin-left: 0em ;
86     font-weight: bold ;}
87 a.tocchapter {display: inline-block ; margin-left: 0em ;
88     font-weight: bold ;}
89 a.tocsection {display: inline-block ; margin-left: 1em ;
90     text-indent: -.5em ; font-weight: bold ;}
91 a.tocsubsection {display: inline-block ; margin-left: 2em ;
92     text-indent: -.5em ;}
93 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
94     text-indent: -.5em ;}
95 a.tocparagraph {display: inline-block ; margin-left: 4em ;
96     text-indent: -.5em ;}
97 a.toc subparagraph {display: inline-block ; margin-left: 5em ;
98     text-indent: -.5em ;}
99 a.tocfigure {margin-left: 0em}
100 a.tocsubfigure {margin-left: 2em}
101 a.toctable {margin-left: 0em}
102 a.tocstable {margin-left: 2em}
103 a.toclstlisting {margin-left: 0em}
104
105
106 body {
107     font-family: "DejaVu Serif", "Bitstream Vera Serif",
108         "Lucida Bright", Georgia, serif;
109     background: #FAF7F4 ;
110     color: black ;
111     margin:0em ;
112     padding:0em ;
113     font-size: 100% ;
114     line-height: 1.2 ;
115 }
116
117 p {margin: 1.5ex 0em 1.5ex 0em ;}
118
119 /* Holds a section number to add space between it and the name */
120 span.sectionnumber { margin-right: .6em }
121
122 /* Inserted in front of index lines */
123 span.indexitem {margin-left: 0em}
124 span.indexsubitem {margin-left: 2em}
125 span.indexsubsubitem {margin-left: 4em}
126
127 div.hidden { display: none ; }
128
129 kbd {
130     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
131         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
132         "Courier New", monospace;
133     font-size: 100% ;
```

```
134 }
135
136 span.strong { font-weight: bold; }
137
138 span.textmd { font-weight: normal; }
139
140 span.textsc { font-variant: small-caps; }
141
142 span.textup { font-variant: normal; }
143
144 span.textrm {
145     font-family: "DejaVu Serif", "Bitstream Vera Serif",
146     "Lucida Bright", Georgia, serif;
147 }
148
149 span.textsf {
150     font-family: "DejaVu Sans", "Bitstream Vera Sans",
151     Geneva, Verdana, sans-serif ;
152 }
153
154 span.attribution {
155     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
156 }
157
158 span.citetitle {
159     margin-left: 1em ; font-size: 80% ; font-style: oblique;
160 }
161
162 span.poemtitle {
163     font-size: 120% ; font-weight: bold;
164 }
165
166 blockquote {
167 margin-left: 0px ;
168 margin-right: 0px ;
169 }
170
171 blockquote p {
172     line-height: 1.5;
173     text-align: left ;
174     font-size: .85em ;
175     margin-left: 3em ;
176 margin-right: 3em ;
177 }
178
179 blockquotation {
180 margin-left: 0px ;
181 margin-right: 0px ;
182 }
183
```

```
184 blockquotation p {
185   line-height: 1.5;
186   text-align: left ;
187   font-size: .85em ;
188   margin-left: 3em ;
189 margin-right: 3em ;
190 }
191
192 div.epigraph {
193   line-height: 1.2;
194   text-align: left ;
195   padding: 3ex 1em 0ex 1em ;
196 /*    margin: 3ex auto 3ex auto ; /* Epigraph centered */
197   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
198 /*    margin: 3ex 1em 3ex 1em ; /* Epigraph to the left */
199   font-size: .85em ;
200   max-width: 27em ;
201 }
202
203
204
205 div.epigraphsource{
206   text-align:right ;
207   margin-left:auto ;
208 /*    max-width: 50% ; */
209   border-top: 1px solid #AOAOAO ;
210   padding-bottom: 3ex ;
211   line-height: 1.2;
212 }
213
214 div.epigraph p { padding: .5ex ; margin: 0ex ;}
215 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
216
217
218
219
220 html body {
221   margin: 0 ;
222   line-height: 1.2;
223 }
224
225
226 body div {
227   margin: 0ex;
228 }
229
230
231 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
232 {
233   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
```

```
234     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
235     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
236     "Times New Roman", serif;
237     font-style: normal ;
238     font-weight: bold ;
239     text-align: left ;
240 }
241
242 h1 { /* title of the entire website, used on each page */
243     text-align: center ;
244     font-size: 2.5em ;
245     padding: .4ex 0em 0ex 0em ;
246 }
247 h2 { font-size: 2.25em }
248 h3 { font-size: 2em }
249 h4 { font-size: 1.75em }
250 h5 { font-size: 1.5em }
251 h6 { font-size: 1.25em }
252 span.paragraph {font-size: 1em ; font-variant: normal ;
253     margin-right: 1em ; }
254 span subparagraph {font-size: 1em ; font-variant: normal ;
255     margin-right: 1em ; }
256
257
258
259 /* Title of the file */
260 h1 {
261     margin: 0ex 0em 0ex 0em ;
262     line-height: 1.3;
263     text-align: center ;
264 }
265
266 /* Part */
267 h2 {
268     margin: 1ex 0em 1ex 0em ;
269     line-height: 1.3;
270     text-align: center ;
271 }
272
273 /* Chapter */
274 h3 {
275     margin: 3ex 0em 1ex 0em ;
276     line-height: 1.3;
277 }
278
279 /* Section */
280 h4 {
281     margin: 3ex 0em 1ex 0em ;
282     line-height: 1.3;
283 }
```

```
284
285 /* Sub-Section */
286 h5 {
287   margin: 3ex 0em 1ex 0em ;
288   line-height: 1.3;
289 }
290
291 /* Sub-Sub-Section */
292 h6 {
293   margin: 3ex 0em 1ex 0em ;
294   line-height: 1.3;
295 }
296
297
298 div.titlepage {
299   text-align: center ;
300 }
301
302 .footnotes {
303   font-size: .85em ;
304   margin: 3ex 1em 0ex 1em ;
305   padding-bottom: 1ex ;
306 }
307
308 .marginpar {
309 width:20%; float:right;
310 text-align:left;
311 margin: 1ex 0.5em 1ex 1em ;
312 padding: 1ex 0.5em 1ex 0.5em ;
313 font-size: 85% ;
314 border-top: 1px solid silver ;
315 border-bottom: 1px solid silver ;
316 overflow-x: auto;
317 }
318
319 .marginpar br { margin-bottom: 2ex ; }
320
321 div.marginblock {
322 width:50%; float:right;
323 text-align:left;
324 margin: 1ex 0.5em 1ex 1em ;
325 padding: 1ex 0.5em 1ex 0.5em ;
326 overflow-x: auto;
327 }
328
329 div.marginblock div.minipage p { font-size: 85%}
330
331 div.marginblock br { margin-bottom: 2ex ; }
332
333
```

```
334 section.textbody div.footnotes{
335     margin: 3ex 0em 0ex 0em ;
336     border-bottom: 2px solid silver ;
337 }
338
339 .footnoteheader {
340     border-top: 2px solid silver ;
341     margin-top: 3ex ;
342     padding-top: 1ex ;
343     font-weight: bold ;
344 }
345
346 .mpfootnotes {
347     text-align: left ;
348     font-size: .85em ;
349     margin: 3ex 0em 3ex 0em ;
350     border-top: 1px solid silver ;
351     border-bottom: 1px solid silver ;
352 /*     padding-top: 1ex ; */
353 }
354
355 /* Remove footnote top border in the title page. */
356 div.titlepage div.mpfootnotes {
357     border-top: none ;
358 }
359
360
361
362 ol {
363     margin: 1ex 1em 1ex 0em;
364     line-height: 1.2;
365 }
366
367 ul, body dir, body menu {
368     margin: 1ex 1em 1ex 0em;
369     line-height: 1.2;
370 }
371
372 li { margin: 0ex 0em 1ex 0em; }
373
374 html {
375     margin: 0;
376     padding: 0;
377 }
378
379 .programlisting {
380     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
381             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
382             "Courier New", monospace;
383     margin: 1ex 0ex 1ex 0ex ;
```

```
384   padding: .5ex 0pt .5ex 0pt ;
385   overflow-x: auto;
386 }
387
388 section.textbody>pre.programlisting {
389 border-top: 1px solid silver ;
390 border-bottom: 1px solid silver ;
391 }
392
393
394 .inlineprogramlisting {
395   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
396               "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
397               "Courier New", monospace;
398   overflow-x: auto;
399 }
400
401
402 div.abstract {
403   margin: 2em 5% 2em 5% ;
404   padding: 1ex 1em 1ex 1em ;
405 /*   font-weight: bold ; */
406   font-size: 90% ;
407 }
408
409 div.abstract dl {line-height:1.5;}
410 div.abstract dt {color:#304070;}
411
412 div.abstracttitle{
413   font-family: "URW Classico", Optima, "Linux Biolinum O",
414               "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
415               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
416   font-weight:bold;
417   font-size:1.25em;
418   text-align: center ;
419 }
420
421 span.abstractrunintitle{
422   font-family: "URW Classico", Optima, "Linux Biolinum O",
423               "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
424               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
425   font-weight:bold;
426 }
427
428
429 .verbatim {
430   overflow-x: auto ;
431 }
432
433 .alltt {
```

```
434     overflow-x: auto ;
435 }
436
437
438 .bverbatim {
439     margin: 1ex 0pt 1ex 0pt ;
440     padding: .5ex 0pt .5ex 0pt ;
441     overflow-x: auto ;
442 }
443
444 .lverbatim {
445     margin: 1ex 0pt 1ex 0pt ;
446     padding: .5ex 0pt .5ex 0pt ;
447     overflow-x: auto ;
448 }
449
450 .fancyvrb {
451     font-size:.85em ;
452     margin: 3ex 0pt 3ex 0pt
453 }
454
455 .fancyvrblabel {
456     font-weight:bold;
457     text-align: center ;
458 }
459
460
461 .verse {
462     font-family: "Linux Libertine Mono O", "Lucida Console",
463             "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
464             "Liberation Mono", "FreeMono", "Andale Mono",
465             "Nimbus Mono L", "Courier New", monospace;
466     margin-left: 1em ;
467 }
468
469
470 div.singlespace { line-height: 1.2 ; }
471 div.onehalfspace { line-height: 1.5 ; }
472 div.doublespace { line-height: 2 ; }
473
474
475 .theorem {
476     background: #e0e0e0;
477     padding: 1ex ;
478     margin: 1em ;
479     border: 1px solid silver ;
480 }
481
482 .theoremlabel { font-weight: bold ; }
483
```

```
484 .theoremcontents { font-style: italic; margin-left: 1em ; }
485
486
487 /* Minipage environments, vertically aligned to top, center, bottom: */
488 .minipage {
489     /* display: inline-block ; */
490     /* Mini pages which follow each other will be tiled. */
491     margin: .25em .25em .25em .25em;
492     padding: .25em .25em .25em .25em;
493     display: inline-flex;
494     flex-direction: column ;
495     overflow: auto;
496 }
497
498 /* Paragraphs in the flexbox did not collapse their margins. */
499 /* Have not yet researched this. */
500 .minipage p {margin: .75ex 0em .75ex 0em ;}
501
502
503
504 .framebox {
505     margin: 0ex ;
506     padding: 0ex ;
507     border: 1px solid black;
508     border-radius: 0px ;
509     padding: .3ex .2em 0ex .2em ;
510     margin: .1ex ;
511     display: inline-block ;
512 }
513
514
515 .mdframed {
516     /* padding: 0ex ; */
517     /* border: 1px solid blaefck; */
518     /* border-radius: 0px ; */
519     padding: 0ex ;
520     margin: 3ex 5% 3ex 5% ;
521     /* display: inline-block ; */
522 }
523
524 .mdframed p { padding: 0ex .5em 0ex .5em ; }
525
526 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
527
528 .mdframedtitle {
529     padding: .5em ;
530     display: block ;
531     font-size: 130%
532 }
533
```

```
534 .mdframedsubtitle {
535     padding: 0ex .5em 0ex .5em ;
536     display: block ;
537     font-size: 115% ;
538 }
539
540 .mdframedsubsubtitle {
541     padding: 0ex .5em 0ex .5em ;
542     display: block ;
543 }
544
545 .mdtheorem {
546     padding: 0ex .5em 0ex .5em ;
547     margin: 3ex 5% 3ex 5% ;
548 /*   display: inline-block ; */
549 }
550
551
552
553
554 dl {
555     margin: 1ex 2em 1ex 0em;
556     line-height: 1.3;
557 }
558
559 dl dt {
560     margin-top: 1ex;
561     font-weight: bold;
562 }
563
564 dl dd p { margin-top: 0em; }
565
566
567 nav.toc, nav.lof, nav.lot, nav.lol {
568     font-family: "URW Classico", Optima, "Linux Biolinum 0",
569                 "DejaVu Sans", "Bitstream Vera Sans",
570                 Geneva, Verdana, sans-serif ;
571     margin-bottom: 4ex ;
572 }
573
574 nav.toc p, nav.lof p, nav.lot p, nav.lol p {
575     line-height: 1.2 ;
576     margin-top:.5ex ;
577     margin-bottom:.5ex;
578     font-size: .9em ;
579 }
580
581
582
583 img, img.hyperimage, img.borderimage {
```

```
584     max-width: 600px;
585     border: 1px solid silver;
586     box-shadow: 3px 3px 3px #808080 ;
587     padding: .5% ;
588     margin: .5% ;
589     background: none ;
590 }
591
592 img.inlineimage{
593     padding: 0px ;
594     box-shadow: none ;
595     border: none ;
596     background: none ;
597     margin: 0px ;
598     display: inline-block ;
599     border-radius: 0px ;
600 }
601
602 img.logoimage{
603     max-width: 300px ;
604     box-shadow: 3px 3px 3px #808080 ;
605     border: 1px solid black ;
606     background:none ;
607     padding:0 ;
608     margin:.5ex ;
609     border-radius: 10px ;
610 }
611
612
613 .section {
614 /*
615     To have each section float relative to each other:
616 */
617 /*
618     display: block ;
619     float: left ;
620     position: relative ;
621     background: white ;
622     border: 1px solid silver ;
623     padding: .5em ;
624 */
625     margin: 0ex .5em 0ex .5em ;
626     padding: 0 ;
627 }
628
629
630 figure {
631     margin: 3ex auto 3ex auto ;
632     padding: 1ex 1em 1ex 1em ;
633     overflow-x: auto ;
```

```
634 }
635
636
637 /* To automatically center images in figures: */
638 /*
639 figure img.inlineimage {
640     margin: 0ex auto 0ex auto ;
641     display: block ;
642 }
643 */
644
645 /* To automatically center minipages in figures: */
646 /*
647 figure div.minipage, figure div.minipage div.minipage {
648     margin: 1ex auto 1ex auto ;
649     display: block ;
650 }
651 */
652
653 figure div.minipage p { font-size: 85% ; }
654
655 figure.subfigure, figure.subtable {
656     display: inline-block ; margin: 3ex 1em 3ex 1em ;
657 }
658
659 div.floatrow { text-align: center; }
660
661 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
662
663 div.floatfoot { font-size: .85em ;
664     border-top: 1px solid silver ; line-height: 1.2 ; }
665
666 figcaption , .lstlistingtitle {
667     font-size: .85em ;
668     text-align: center ;
669     font-weight: bold ;
670 margin-top: 1ex ;
671 margin-bottom: 1ex ;
672 }
673
674 figure.subfigure figcaption, figure.subtable figcaption {
675     border-bottom: none ; background: none ;
676 }
677
678 div.nonfloatcaption {
679     margin: 1ex auto 1ex auto ;
680     font-size: .85em ;
681     text-align: center ;
682     font-weight: bold ;
683 }
```

```
684
685 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
686 figure div.floatrow div.minipage figcaption {
687 border: none ;
688 background: none ;
689 }
690
691
692 table {
693     margin: 1ex auto 1ex auto ;
694     border-collapse: collapse ;
695     border-spacing: 0px ;
696     line-height: 1.3 ;
697 }
698
699 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
700     margin-bottom: 0ex ; } /* for \hline */
701
702 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
703     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
704
705 td {padding: 1ex .5em 1ex .5em ;}
706
707 table td.tdl { text-align: left ; vertical-align: middle ; }
708 table td.tdc { text-align: center ; vertical-align: middle ; }
709 table td.tdr { text-align: right ; vertical-align: middle ; }
710 table td.tdp { text-align: left ; vertical-align: bottom ; }
711 table td.tdm { text-align: left ; vertical-align: middle ; }
712 table td.tdb { text-align: left ; vertical-align: top ; }
713 table td.tdP { text-align: center ; vertical-align: bottom ; }
714 table td.tdM { text-align: center ; vertical-align: middle ; }
715 table td.tdB { text-align: center ; vertical-align: top ; }
716 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
717     vertical-align: middle ; } /* for cmidrule */
718 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
719     vertical-align: middle ; }
720 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
721     vertical-align: middle ; }
722 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
723     vertical-align: bottom ; }
724 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
725     vertical-align: middle ; }
726 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
727     vertical-align: top ; }
728 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
729     vertical-align: bottom ; }
730 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
731     vertical-align: middle ; }
732 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
733     vertical-align: top ; }
```

```
734
735 /* Margins of paragraphs inside table cells: */
736 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
737     padding-bottom: 1ex ; margin: 0ex ; }
738 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
739     padding-bottom: 1ex ; margin: 0ex ; }
740 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
741     padding-bottom: 1ex ; margin: 0ex ; }
742
743 td.tdp , td.tdprule , td.tdP , td.tdPrule
744     { padding: 0ex .5em 0ex .5em ; }
745 td.tdm , td.tdmrule , td.tdM , td.tdMrule
746     { padding: 0ex .5em 0ex .5em ; }
747 td.tdb , td.tdbrule , td.tdB , td.tdBrule
748     { padding: 0ex .5em 0ex .5em ; }
749
750
751 /* table notes: */
752 .tnotes {
753     margin: 0ex 5% 1ex 5% ;
754     padding: 0.5ex 1em 0.5ex 1em;
755     font-size:.85em;
756     text-align: left ;
757 }
758
759 .tnotes dl dt p {margin-bottom:0px;}
760
761 .tnoteitemheader {margin-right: 1em;}
762
763
764
765 /* center, flushleft, flushright environments */
766 div.center{text-align:center;}
767 div.center table {margin-left:auto;margin-right:auto;}
768 div.flushleft{text-align:left;}
769 div.flushleft table {margin-left:0em ; margin-right:auto;}
770 div.flushright{text-align:right;}
771 div.flushright table {margin-left:auto ; margin-right: 0em ;}
772
773
774
775
776 /* program listing callouts: */
777 span.callout {
778     font-family: "DejaVu Sans", "Bitstream Vera Sans",
779         Geneva, Verdana, sans-serif ;
780     border-radius: .5em;
781     background-color:black;
782     color:white;
783     padding:0px .25em 0px .25em;
```

```
784 margin: 0 ;
785     font-weight: bold;
786     font-size:.72em ;
787 }
788
789 div.programlisting pre.verbatim span.callout{
790 font-size: .85em ;
791 }
792
793
794
795
796
797 div.published
798 {
799     text-align: center ;
800     font-variant: normal ;
801     font-style: italic ;
802     font-size: 1em ;
803     margin: 3ex 0em 3ex 0em ;
804 }
805
806 div.subtitle
807 {
808     text-align: center ;
809     font-variant: normal ;
810     font-style: italic ;
811     font-size: 1.25em ;
812     margin: 3ex 0em 3ex 0em ;
813 }
814
815 div.subtitle p { margin: 1ex ; }
816
817 div.author
818 {
819     font-variant: normal ;
820     font-style: normal ;
821     font-size: 1em ;
822     margin: 3ex 0em 3ex 0em ;
823 }
824
825 div.author table {
826     margin: 3ex auto 0ex auto ;
827     background: none ;
828 }
829
830 div.author table tbody tr td { padding: .25ex ; }
831
832 span.affiliation {font-size: .85em ; font-variant: small-caps; }
833
```

```
834 div.titledate {
835     text-align: center ;
836     font-size: .85em ;
837     font-style: italic;
838     margin: 6ex 0em 6ex 0em ;
839 }
840
841
842 nav.topnavigation{
843     text-align: left ;
844     padding: 0.5ex 1em 0.5ex 1em ;
845 /*      margin: 2ex 0em 3ex 0em ; */
846     margin: 0 ;
847     border-bottom: 1px solid silver ;
848     border-top: 1px solid silver ;
849     clear:right ;
850 }
851
852 nav.botnavigation{
853     text-align: left ;
854     padding: 0.5ex 1em 0.5ex 1em ;
855 /*      margin: 3ex 0em 2ex 0em ; */
856     margin: 0 ;
857     border-top: 1px solid silver ;
858     border-bottom: 1px solid silver ;
859     clear:right ;
860 }
861
862
863 header{
864     line-height: 1.2 ;
865     font-size: 1em ;
866 /*      border-bottom: 2px solid silver ; */
867     margin: 0px ;
868     padding: 0ex 1em 0ex 1em ;
869     text-align:center ;
870 }
871
872 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
873
874
875 footer{
876     font-size: .85em ;
877     line-height: 1.2 ;
878     margin-top: 1ex ;
879     border-top: 2px solid silver ;
880     padding: 2ex 1em 2ex 1em ;
881     clear:right ;
882     text-align:left ;
883 }
```

```
884
885
886 a.linkhome { font-weight:bold ; font-size: 1em ;}
887
888
889 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
890
891 img.lateximage{
892     padding: 0px 0px 0px 0px ;
893     box-shadow: none ;
894     border: none ;
895     background: none ;
896     margin: 0px 0px -.15ex 0px ;
897     /* pdfcrop leaves a slight margin, adjust to baseline */
898     max-width: 100% ;
899     border-radius: 0ex ;
900     border: none ;
901 }
902
903
904
905 nav.sidetoc {
906     font-family: "DejaVu Serif", "Bitstream Vera Serif",
907         "Lucida Bright", Georgia, serif;
908     float:right ;
909     width: 20%;
910     border-left: 1px solid silver;
911     border-top: 1px solid silver;
912     border-bottom: 1px solid silver;
913 /*     border-top: 2px solid #808080 ; */
914     background: #FAF7F4 ;
915     padding: 2ex 0em 2ex 1em ;
916     margin: 0ex 0em 2ex 1em ;
917     font-size:.9em ;
918     border-radius: 20px 0px 0px 20px ;
919 }
920
921 div.sidetoccontents {
922 /*     border-top: 1px solid silver ; */
923     overflow-y: auto ;
924     width: 100% ;
925     text-align: left ;
926 }
927
928 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
929     text-indent: 0 ; }
930 nav.sidetoc p a {color:black ; font-size: .7em ;}
931 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
932     border-bottom: 1px solid silver ;     }
933 nav.sidetoc a:hover {text-decoration: underline ; }
```

```
934
935
936
937 section.textbody { margin: 0ex 1em 0ex 1em ;}
938
939
940 div.multicolsheading { -webkit-column-span: all;
941     -moz-column-span: all; column-span: all; }
942 div.multicols { -webkit-columns: 3 380px ;
943     -moz-columns: 3 380px ; columns: 3 380px ; }
944 div.multicols p {margin-top: 0ex}
945
946 /* Used to support algorithmicx: */
947 span.floatright { float: right ; }
948
949
950
951 /*
952 For CSS LaTeX and related logos:
953 Based on:
954 http://edward.oconnor.cx/2007/08/tex-poshlet
955 http://nitens.org/taraborelli/texlogo
956 */
957
958 .latexlogofont {
959     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
960         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
961     font-variant: normal ;
962 }
963
964 .latexlogo {
965     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
966         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
967     letter-spacing: .03em ;
968     font-size: 1.1em;
969 }
970
971 .latexlogo sup {
972     text-transform: uppercase;
973     letter-spacing: .03em ;
974     font-size: 0.85em;
975     vertical-align: 0.15em;
976     margin-left: -0.36em;
977     margin-right: -0.15em;
978 }
979
980 .latexlogo sub {
981     text-transform: uppercase;
982     vertical-align: -0.5ex;
983     margin-left: -0.1667em;
```

```
984   margin-right: -0.125em;
985   font-size: 1em;
986 }
987
988 .xetexlogo {
989   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
990   "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
991   letter-spacing: .03em ;
992   font-size: 1.1em;
993 }
994
995 /* A smaller gap between Xe and Tex v.s. LaTeX: */
996 .xetexlogo sub {
997   text-transform: uppercase;
998   vertical-align: -0.5ex;
999   margin-left: -0.0667em;
1000   margin-right: -0.2em;
1001   font-size: 1em;
1002   letter-spacing: .03em ;
1003 }
1004
1005 /* A large gap between Xe and LaTeX v.s. TeX: */
1006 .xelatexlogo sub {
1007   text-transform: uppercase;
1008   vertical-align: -0.5ex;
1009   margin-left: -0.0667em;
1010   margin-right: -.05em;
1011   font-size: 1em;
1012   letter-spacing: .03em ;
1013 }
1014
1015 .amslogo {
1016   font-family: "TeXGyreChorus","URW Chancery L",
1017   "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1018   "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1019   "Hoefler Text", Times, "Times New Roman", serif;
1020   font-style: italic;
1021 }
1022
1023 .lyxlogo {
1024   font-family: "URW Classico", Optima, "Linux Biolinum O",
1025   "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1026 Verdana, sans-serif ;
1027 }
1028
1029
1030
1031
1032 /* Only display top and bottom navigation if a small screen: */
1033 /* Hide the sidetoc if a small screen: */
```

```
1034 nav.topnavigation { display:none; }
1035 nav.botnavigation { display:none; }
1036
1037 @media screen and (max-width: 45em) {
1038 /*      nav.sidetoc {display:none;} */
1039     nav.sidetoc {
1040         float: none ;
1041         width: 100% ;
1042         margin: 5ex 0px 5ex 0px ;
1043         padding: 0 ;
1044         border-radius: 0 ;
1045         border-bottom: 1px solid black ;
1046         border-top: 1px solid black ;
1047         box-shadow: none ;
1048     }
1049 /*      nav.topnavigation { display:block } */
1050     nav.botnavigation { display:block }
1051 }
1052
1053 @media print {
1054     body {
1055         font-family: "Linux Libertine O",
1056             "DejaVu Serif", "Bitstream Vera Serif",
1057             "Liberation Serif", "Nimbus Roman No 9 L",
1058             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1059     }
1060     nav.sidetoc { display:none; }
1061     nav.topnavigation { display: none; }
1062     nav.botnavigation { display: none; }
1063 }
1064
1065 @media handheld {
1066     nav.sidetoc { display:none; }
1067     nav.topnavigation { display:block }
1068     nav.botnavigation { display:block }
1069 }
1070
1071 @media projection {
1072     nav.sidetoc { display:none; }
1073     nav.topnavigation { display:block }
1074     nav.botnavigation { display:block }
1075 }
1076 \end{VerbatimOut}
1077 % \end{Verbatim}% for syntax highlighting
```

## 70.4 lwarp\_sagebrush.css

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

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

```
1078 \begin{VerbatimOut}{lwarp_sagebrush.css}
1079 @import url("lwarp.css") ;
1080
1081
1082 A:link {color:#105030 ; text-decoration: none ; }
1083 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1084 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1085 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1086
1087
1088
1089 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1090 {
1091     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1092             "Linux Libertine 0", "Liberation Serif",
1093             "Nimbus Roman No 9 L", "FreeSerif",
1094             "Hoefler Text", Times, "Times New Roman", serif;
1095     font-variant: small-caps ;
1096 font-weight: normal ;
1097     color: #304070 ;
1098     text-shadow: 2px 2px 3px #808080;
1099 }
1100
1101 h1 { /* title of the entire website, used on each page */
1102     font-variant: small-caps ;
1103     color: #304070 ;
1104     text-shadow: 2px 2px 3px #808080;
1105     background-color: #F7F7F0 ;
1106     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1107 }
1108
1109 h1 {
1110     border-bottom: 1px solid #304070;
1111     border-top: 2px solid #304070;
1112 }
1113
1114 h2 {
1115     border-bottom: 1px solid #304070;
1116     border-top: 2px solid #304070;
1117     background-color: #F7F7F0 ;
1118     background-image: linear-gradient(to bottom, #F7F7F0, #DADOC0);
1119 }
1120
1121
1122
1123 div.abstract {
```

```
1124     background: #f5f5eb ;
1125     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1126
1127     border: 1px solid silver;
1128     border-radius: 1em ;
1129 }
1130
1131 div.abstract dl {line-height:1.5;}
1132 div.abstract dt {color:#304070;}
1133
1134 div.abstracttitle{
1135     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1136         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1137         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1138     font-weight:bold;
1139     font-variant: small-caps ;
1140     font-size:1.5em;
1141     border-bottom: 1px solid silver ;
1142     color: #304070 ;
1143     text-align: center ;
1144     text-shadow: 1px 1px 2px #808080;
1145 }
1146
1147 span.abstractrunintitle{
1148     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1149         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1150         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1151     font-weight:bold;
1152 }
1153
1154
1155 div.epigraph {
1156     background: #f5f5eb ;
1157     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1158
1159     border: 1px solid silver ;
1160     border-radius: 1ex ;
1161     box-shadow: 3px 3px 3px #808080 ;
1162 }
1163
1164
1165 .example {
1166     background-color: #f5f5eb ;
1167     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1168
1169 }
1170
1171 div.exampletitle{
1172     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1173         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
```

```
1174     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1175     font-weight:bold;
1176     font-variant: small-caps ;
1177     border-bottom: 1px solid silver ;
1178     color: #304070 ;
1179     text-align: center ;
1180     text-shadow: 1px 1px 2px #808080;
1181 }
1182
1183
1184 .sidebar {
1185     background-color: #f5f5eb ;
1186     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1187
1188 }
1189
1190 div.sidebarTitle{
1191     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1192         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1193         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1194     font-weight:bold;
1195     font-variant: small-caps ;
1196     border-bottom: 1px solid silver ;
1197     color: #304070 ;
1198     text-align: center ;
1199     text-shadow: 1px 1px 2px #808080;
1200 }
1201
1202
1203 .fancyvrblabel {
1204     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1205         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
1206         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1207     font-weight:bold;
1208     font-variant: small-caps ;
1209 font-size: 1.5em ;
1210     color: #304070 ;
1211     text-align: center ;
1212     text-shadow: 1px 1px 2px #808080;
1213 }
1214
1215
1216
1217 .minipage {
1218     background-color: #eeeeee7 ;
1219     border: 1px solid silver ;
1220 border-radius: 1ex ;
1221 }
1222
1223 figure.figure .minipage { border: none; }
```

```
1224
1225 div.marginblock div.minipage { border: none; }
1226
1227 figure , div.marginblock {
1228     background-color: #eeeeee7 ;
1229     border: 1px solid silver ;
1230     border-radius: 1ex ;
1231     box-shadow: 3px 3px 3px #808080 ;
1232 }
1233
1234 figure figure {
1235     border: 1px solid silver ;
1236     margin: 0em ;
1237 box-shadow: none ;
1238 }
1239
1240 /*
1241 figcaption {
1242     border-top: 1px solid silver ;
1243     border-bottom: 1px solid silver ;
1244     background-color: #e8e8e8 ;
1245 }
1246 */
1247
1248
1249 div.table {
1250     box-shadow: 3px 3px 3px #808080 ;
1251 }
1252
1253 /*
1254 .tnotes {
1255     background: #e8e8e8;
1256     border: 1px solid silver;
1257 }
1258 */
1259
1260
1261 nav.topnavigation{
1262     background-color: #b0b8b0 ;
1263     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1264 }
1265
1266 nav.botnavigation{
1267     background-color: #b0b8b0 ;
1268     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1269 }
1270
1271
1272
1273 header{
```

```

1274     background-color: #F7F7F0 ;
1275     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1276 }
1277
1278 footer{
1279     background-color: #F7F7F0 ;
1280     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1281 }
1282
1283
1284
1285 nav.sidetoc {
1286     background-color: #F7F7F0 ;
1287     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1288     box-shadow: 3px 3px 3px #808080 ;
1289     border-radius: 0px 0px 0px 20px ;
1290 }
1291
1292 div.sidetotitle {color: #304070 ; }
1293 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1294
1295
1296 @media screen and (max-width: 45em) {
1297     nav.sidetoc { border-radius: 0 ; }
1298 }
1299
1300
1301 \end{VerbatimOut}
1302 % \end{Verbatim} for syntax highlighting

```

## 70.5 lwarp\_formal.css

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

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

```

1303 \begin{VerbatimOut}{lwarp_formal.css}
1304 @import url("lwarp.css") ;
1305
1306
1307
1308 A:link {color:#802020 ; text-decoration:none; }
1309 A:visited {color:#802020 ; text-decoration:none ;}
1310 A:hover {color:#400000 ; text-decoration:none ;}
1311 A:active {color:#C00000 ; text-decoration:none ;}
1312

```

```
1313
1314 body {
1315     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1316         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1317         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1318         "Times New Roman", serif;
1319     background: #fffcf5;
1320 }
1321
1322 span.textrm {
1323     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1324         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1325         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1326         "Times New Roman", serif;
1327 }
1328
1329 span.textsf {
1330     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1331         Geneva, Verdana, sans-serif ;
1332 }
1333
1334
1335
1336 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1337 {
1338     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1339         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1340         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1341         "Times New Roman", serif;
1342     color: #800000 ;
1343     text-shadow: none ;
1344 }
1345
1346 h1, h2 {
1347     background-color: #fffcf5 ;
1348     background-image: none ;
1349     border-bottom: 1px solid #808080;
1350     border-top: 2px solid #808080;
1351 }
1352
1353 div.abstracttitle {
1354     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1355         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1356         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1357         "Times New Roman", serif;
1358     color: black ;
1359     text-shadow: none ;
1360 }
1361
1362 span.abstractrunintitle {
```

```
1363     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1364         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1365         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1366         "Times New Roman", serif;
1367     color: black ;
1368     text-shadow: none ;
1369 }
1370
1371 div.abstract { font-size: 100% }
1372
1373 .sidebar {
1374     background: #fffcf5;
1375     background-image: none ;
1376     margin: 2em 5% 2em 5%;
1377     padding: 0.5em 1em;
1378     border: none ;
1379     border-top : 1px solid silver;
1380     border-bottom : 1px solid silver;
1381     font-size: 90% ;
1382 }
1383
1384 div.sidebartitle{
1385     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1386         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1387         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1388         "Times New Roman", serif;
1389     color: #800000 ;
1390     text-shadow: none ;
1391     border: none ;
1392 }
1393
1394 .example {
1395     background: #fffcf5;
1396     background-image: none ;
1397     margin: 2em 5% 2em 5%;
1398     padding: 0.5em 1em;
1399     border: none ;
1400     border-top : 1px solid silver;
1401     border-bottom : 1px solid silver;
1402 }
1403
1404 div.examplettitle{
1405     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1406         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1407         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1408         "Times New Roman", serif;
1409     color: #800000 ;
1410     text-shadow: none ;
1411     border: none ;
1412 }
```

```
1413
1414 div.fancyvrblabel{
1415     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1416         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1417         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1418         "Times New Roman", serif;
1419     color: #800000 ;
1420     text-shadow: none ;
1421     border: none ;
1422 }
1423
1424
1425
1426 .verse {
1427     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1428         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1429         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1430         "Times New Roman", serif;
1431 }
1432
1433
1434 figure {
1435     margin: 3ex 5% 3ex 5% ;
1436     padding: 1ex 1em 1ex 1em ;
1437     background-color: #ffffcf5 ;
1438     overflow-x: auto ;
1439     border: none ;
1440 /*     border-top: 1px solid silver; */
1441 /*     border-bottom: 1px solid silver; */
1442 }
1443
1444
1445 figcaption , .lstlisting {
1446     border: none ;
1447 /*     border-top: 1px solid silver ; */
1448 /*     border-bottom: 1px solid silver ; */
1449     background-color: #ffffcf5 ;
1450 }
1451
1452 .tnotes {
1453     background: #ffffcf5 ;
1454 }
1455
1456 .theorem {
1457     background: none ;
1458 }
1459
1460 .minipage {
1461     background-color: #ffffcf5 ;
1462     border: none ;
```

```
1463     margin: 1em ;
1464 }
1465
1466 div.floatrow figure { border: none ; }
1467
1468 figure figure { border: none ; }
1469
1470
1471 nav.toc, nav.lof, nav.lot, nav.lol {
1472     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1473         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1474         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1475         "Times New Roman", serif;
1476 }
1477
1478 nav.sidetoc {
1479     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1480         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1481         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1482         "Times New Roman", serif;
1483     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
1484     border-radius: 0px 0px 0px 20px ;
1485 }
1486
1487 div.sidetoctitle{
1488     color: #800000 ;
1489 }
1490
1491 header{
1492     background-color: #e0e0e0 ;
1493     background-image: linear-gradient(to top, #ffffcf5, #b0b0b0);
1494     text-align:center ;
1495 }
1496
1497 footer{
1498     background-color: #e0e0e0 ;
1499     background-image: linear-gradient(to bottom, #ffffcf5, #b0b0b0);
1500     padding: 2ex 1em 2ex 1em ;
1501     clear:right ;
1502     text-align:left ;
1503 }
1504
1505 nav.botnavigation {
1506     background: #dedcd5 ;
1507     border-top: 1px solid black ;
1508 }
1509 \end{VerbatimOut}
1510 % \end{Verbatim}% for syntax highlighting
```

## 70.6 sample\_project.css

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

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

```
1511 \begin{VerbatimOut}{sample_project.css}
1512 /* ( --- Start of project.css --- ) */
1513 /* A sample project-specific CSS file for lwarf --- ) */
1514
1515 /* Load default lwarf settings: */
1516 @import url("lwarf.css") ;
1517 /* or lwarf_formal.css, lwarf_sagebrush.css */
1518
1519 /* Project-specific CSS setting follow here. */
1520 /* . . . */
1521
1522 /* ( --- End of project.css --- ) */
1523 \end{VerbatimOut}
1524 % \end{Verbatim}% for syntax highlighting
```

## 70.7 lwarf\_html.xdy

File `lwarf_html.xdy` Used to modify the index for lwarf.

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

```
1525 \begin{VerbatimOut}{lwarf_html.xdy}
1526 (require "tex/inputenc/latin.xdy")
1527 (merge-rule "\PS *" "Postscript")
1528 (require "texindy.xdy")
1529 (require "page-ranges.xdy")
1530 (require "book-order.xdy")
1531 (markup-locref :open "\hyperindexref{" :close "}")
1532 \end{VerbatimOut}
1533 % \end{Verbatim}% for syntax highlighting
```

## 70.8 lwarf\_mathjax.txt

File `lwarf_mathjax.txt` Used by lwarf when using MathJax.

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

```
1534 \begin{VerbatimOut}{lwarp_mathjax.txt}
1535 <!-- https://groups.google.com/forum/#!topic/
1536           mathjax-users/jUtewUcE2bY -->
1537 <script type="text/x-mathjax-config">
1538 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1539     var seteqsectionDefault = {name: "", num: 0};
1540     var seteqsections = {}, seteqsection = seteqsectionDefault;
1541     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1542     var AMS = MathJax.Extension["TeX/AMSmath"];
1543     TEX.Definitions.Add({
1544         macros: {
1545             seteqsection: "mySection",
1546             seteqnumber: "mySetEqNumber"
1547         }
1548     });
1549
1550     PARSE.Augment({
1551         mySection: function (name) {
1552             seteqsection.num = AMS.number;
1553             var n = this.GetArgument(name);
1554             if (n === "") {
1555                 seteqsection = seteqsectionDefault;
1556             } else {
1557                 if (!seteqsections["_"+n])
1558                     seteqsections["_"+n] = {name:n, num:0};
1559                 seteqsection = seteqsections["_"+n];
1560             }
1561             AMS.number = seteqsection.num;
1562         },
1563         mySetEqNumber: function (name) {
1564             var n = this.GetArgument(name);
1565             if (!n || !n.match(/^\*[0-9]+\*$/))
1566                 n = ""; else n = parseInt(n)-1;
1567             <!-- $ syntax highlighting -->
1568             if (n === "" || n < 1)
1569                 TEX.Error
1570                 ("Argument to "+name+" should be a positive integer");
1571             AMS.number = n;
1572         }
1573     });
1574     MathJax.Hub.Config({
1575         TeX: {
1576             equationNumbers: {
1577                 formatTag: function (n)
1578                     {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"};
1579                 formatID: function (n) {
```

```

1580         n = (seteqsection.name+'.'+n).replace
1581             (/[:]'/>&]/g,"").replace(/^\./,"");
1582             return 'mjax-eqn-' + n;
1583     }
1584   }
1585 }
1586 });
1587 });
1588 </script>
1589
1590 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1591 <script type="text/x-mathjax-config">
1592   MathJax.Ajax.config.path["Contrib"] =
1593     "https://cdn.mathjax.org/mathjax/contrib";
1594 </script>
1595
1596 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1597                 tree/master/siunitx -->
1598 <script type="text/x-mathjax-config">
1599   MathJax.Hub.Config({
1600     extensions: ["tex2jax.js","[Contrib]/siunitx/siunitx.js"],
1601     jax: ["input/TeX","output/HTML-CSS"],
1602     tex2jax: {inlineMath: [[[$","$"],["\\(","\\)"]]]},
1603     TeX: {extensions: ["AMSmath.js","AMSSymbols.js", "sinuitx.js"]}
1604   });
1605 </script>
1606
1607 <script type="text/x-mathjax-config">
1608 MathJax.Hub.Config({
1609   TeX: {
1610     equationNumbers: {
1611       autoNumber: "AMS"
1612     }
1613   }
1614 });
1615 </script>
1616
1617 <script
1618   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1619 </script>
1620 \end{VerbatimOut}
1621 % \end{Verbatim}% for syntax highlighting

```

## 70.9 Lwarpmk option

The following is only generated if the `lwarpmk` option was given to `lwarf-newproject`.

```
1622 \begin{LWR@createlwarpmk}
```

File **lwarpmk** Creates a local copy of lwarpmk:

```
1623 \begin{VerbatimOut}{lwarpmk.lua}
1624 #!/usr/bin/env texlua
1625
1626 -- Copyright 2016-2017 Brian Dunn
1627
1628 -- Print the usage of the lwarpmk command:
1629
1630 printversion = "v0.23"
1631
1632 function printhelp ()
1633 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1634 end
1635
1636 function printusage ()
1637 print ( [[
1638
1639 lwarpmk print [project]: Compile a print version.
1640 lwarpmk printindex [project]: Process the index for the print version.
1641 lwarpmk html [project]: Compile an HTML version.
1642 lwarpmk htmlindex [project]: Process the index for the html version.
1643 lwarpmk again [project]: Touch the source code to trigger recompiles.
1644 lwarpmk limages [project]: Process the "lateximages" created by larp.sty.
1645 lwarpmk pdftohtml [project]:
1646     For use with latexmk or a Makefile:
1647     Convert project_html.pdf to project_html.html and
1648     individual HTML files.
1649 lwarpmk clean [project]: Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
1650 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1651 lwarpmk -h: Print this help message.
1652 lwarpmk --help: Print this help message.
1653
1654 ]] )
1655 printconf ()
1656 end
1657
1658 -- Print the format of the configuration file lwarpmk.conf:
1659
1660 function printconf ()
1661 print ( [[
1662 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1663 --
1664 opsystem = "Unix"    (or "Windows")
1665 latexname = "pdflatex"  (or "lualatex", or "xelatex")
1666 sourcename = "projectname"  (the source-code filename w/o .tex)
1667 homehtmlfilename = "index"   (or perhaps the project name)
1668 htmlfilename = ""    (or "projectname" - filename prefix)
1669 uselatexmk = "false"   (or "true" to use latexmk to build PDFs)
1670 --
```

```
1671 Filenames must contain only letters, numbers, underscore, or dash.
1672 Values must be in "quotes".
1673
1674 ]] ) ;
1675 end
1676
1677
1678 -- Split one large sourcefile into a number of files,
1679 -- starting with destfile.
1680 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1681
1682 function splitfile (destfile,sourcefile)
1683 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1684 io.input(sourcefile)
1685 io.output(destfile)
1686 for line in io.lines() do
1687 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
1688 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1689 io.output(newfilename) ;
1690 else -- not a splitpoint
1691 io.write (line .. "\n") ;
1692 end
1693 end -- do
1694 end -- function
1695
1696 -- Incorrect value, so print an error and exit.
1697
1698 function cvalueerror ( line, linenum , cvalue )
1699     print ( linenum .. " : " .. line ) ;
1700     print ("lwarpmk: incorrect variable value \" .. cvalue .. \" in lwarpmk.conf.\n" ) ;
1701     printconf () ;
1702     os.exit(1) ;
1703 end
1704
1705 -- Load settings from the project's "lwarpmk.conf" file:
1706
1707 function loadconf ()
1708 -- Default configuration filename:
1709 local conffile = "lwarpmk.conf"
1710 -- Optional configuration filename:
1711 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1712 -- Verify the file exists:
1713 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1714 print("lwarpmk: " .. conffile .. " does not exist.")
1715 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1716 printhelp () ;
1717 os.exit(1) -- exit the entire lwarpmk script
1718 else -- file exists
1719 -- Read the file:
1720 print ("lwarpmk: Reading " .. conffile ..".")
```

```
1721 io.input(conffile) ;
1722 -- Scan each line:
1723 local linenum = 0
1724 for line in io.lines() do -- scan lines
1725 linenum = linenum + 1
1726 i,j,cvarname,cvalue = string.find (line,"([%w_]*)%s*=%s*\"([%w_]*)\\\"") ;
1727 -- Error if incorrect enclosing characters:
1728 if ( i == nil ) then
1729 print ( linenum .. " : " .. line ) ;
1730 print ( "lwarfpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
1731 printconf () ;
1732 os.exit(1) ;
1733 end
1734 if ( cvarname == "opsystem" ) then
1735     -- Verify choice of opsystem:
1736     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1737         opsystem = cvalue
1738     else
1739         cvalueerror ( line, linenum , cvalue )
1740     end
1741 elseif ( cvarname == "latexname" ) then
1742     -- Verify choice of LaTeX compiler:
1743     if (
1744         (cvalue == "pdflatex") or
1745         (cvalue == "xelatex") or
1746         (cvalue == "lualatex")
1747     ) then
1748         latexname = cvalue
1749     else
1750         cvalueerror ( line, linenum , cvalue )
1751     end
1752 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1753 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1754 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1755 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1756 else
1757 print ( linenum .. " : " .. line ) ;
1758 print ("lwarfpmk: Incorrect variable name \" .. cvarname .. \" in " .. conffile .. ".\n" ) ;
1759 printconf () ;
1760 os.exit(1) ;
1761 end
1762 end -- do scan lines
1763 end -- file exists
1764 -- Select some operating-system commands:
1765 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1766 rmname = "rm"
1767 mvname = "mv"
1768 touchnamepre = "touch"
1769 touchnamepost = ""
1770 dirslash = "/"
```

```
1771 opquote= "\'"
1772 elseif opsystem=="Windows" then -- For Windows
1773 rmname = "DEL"
1774 mvname = "MOVE"
1775 touchnamepre = "COPY /b"
1776 touchnamepost = "+,,"
1777 dirslash = "\\\""
1778 opquote= "\""
1779 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1780 end --- for Windows
1781 end -- loadconf
1782
1783
1784 function refreshdate ()
1785 os.execute(touchnamepre .. " " .. sourcename .. ".tex" .. touchnamepost)
1786 end
1787
1788
1789 -- Scan the LaTeX log file for the phrase "Rerun to get",
1790 -- indicating that the file should be compiled again.
1791 -- Return true if found.
1792
1793 function reruntoget (filesource)
1794 io.input(filesource)
1795 for line in io.lines() do
1796 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1797 end
1798 return false
1799 end
1800
1801 -- Compile one time, return true if should compile again.
1802 -- fsuffix is "" for print, "_html" for HTML output.
1803
1804 function onetime (fsuffix)
1805 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
1806 err = os.execute(
1807 --     "echo " ..
1808     latexname .. " " .. sourcename..fsuffix )
1809 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1810 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
1811 end
1812
1813 -- Compile up to five times.
1814 -- fsuffix is "" for print, "_html" for HTML output
1815
1816 function manytimes (fsuffix)
1817 if onetime(fsuffix) == true then
1818 if onetime(fsuffix) == true then
1819 if onetime(fsuffix) == true then
1820 if onetime(fsuffix) == true then
```

```
1821 if onetime(fsuffix) == true then
1822 end end end end
1823 end
1824
1825 -- Exit if the given file does not exist.
1826
1827 function verifyfileexists (filename)
1828 if (lfs.attributes ( filename , "modification" ) == nil ) then
1829 print ( "lwarpmk: " .. filename .. " not found." ) ;
1830 os.exit (1) ;
1831 end
1832 end
1833
1834 -- Convert <project>_html.pdf into HTML files:
1835
1836 function pdftohtml ()
1837     -- Convert to text:
1838     print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html")
1839     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf" .. sourcename .. ".html")
1840     -- Split the result into individual HTML files:
1841     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
1842 end
1843
1844 -- Remove auxiliary files:
1845
1846 function removeaux ()
1847     os.execute ( rname .. " " ..
1848                 sourcename .. ".aux" .. sourcename .. "_html.aux" ..
1849                 sourcename .. ".toc" .. sourcename .. "_html.toc" ..
1850                 sourcename .. ".lof" .. sourcename .. "_html.lof" ..
1851                 sourcename .. ".lot" .. sourcename .. "_html.lot" ..
1852                 sourcename .. ".idx" .. sourcename .. "_html.idx" ..
1853                 sourcename .. ".ind" .. sourcename .. "_html.ind" ..
1854                 sourcename .. ".log" .. sourcename .. "_html.log"
1855             )
1856 end
1857
1858
1859
1860 -- Create lateximages based on lateximages.txt:
1861 function createlateximages ()
1862 print ("lwarpmk: Creating lateximages.")
1863 io.input("lateximages.txt")
1864 -- Create the lateximages directory, ignore error if alreadt exists
1865 err = os.execute("mkdir lateximages")
1866 -- Scan lateximages.txt
1867 for line in io.lines() do
1868     -- lwimgpage is the page number in the PDF which has the image
1869     -- lwimgnum is the sequential lateximage number to assign for the image
1870 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
```

```
1871 -- For each entry:
1872 if ( (i~=nil) ) then
1873 -- Separate out the image into its own single-page pdf:
1874 err = os.execute(
1875 "pdfseparate -f " .. lwimgpage .. " -l " ..
1876 lwimgpage .. " " .. sourcename .."_html.pdf lateximagetemp-%d.pdf")
1877 -- Crop the image:
1878 err = os.execute(
1879 "pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")
1880 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1881 -- Convert the image to svg:
1882 err = os.execute(
1883 "pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")
1884 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1885 -- Move the result into lateximages/:
1886 err = os.execute(
1887 mvname .. " lateximage-" .. lwimgnum ..".svg lateximages" .. dirslash )
1888 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1889 -- Remove the temporary files:
1890 err = os.execute(
1891 rmname .. " lateximage-" .. lwimgnum ..".pdf lateximagetemp-" .. lwimgpage ..".pdf")
1892 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1893 end
1894 end -- do
1895 end -- function
1896
1897
1898
1899
1900 -- lwarpmk --version :
1901
1902 if (arg[1] == "--version") then
1903 print ( "lwarpmk: " .. printversion )
1904
1905 else -- not -- version
1906
1907 -- print intro:
1908
1909 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarf package.")
1910
1911 -- lwarpmk print:
1912
1913 if arg[1] == "print" then
1914 loadconf ()
1915 if ( uselatexmk == "true" ) then
1916     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\\" .. latexname .. "%0 %S\" " .. sourcename )
1917     print ("lwarpmk: Done.")
1918 else -- not latexmk
1919     verifyfileexists (sourcename .. ".tex") ;
1920     -- See if up to date:
```

```
1921      if (
1922          ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
1923          (
1924              lfs.attributes ( sourcename .. ".tex" , "modification" ) >
1925              lfs.attributes ( sourcename .. ".pdf" , "modification" )
1926          )
1927      ) then
1928          -- Recompile if not yet up to date:
1929          manytimes("")
1930          print ("lwarpmk: Done.") ;
1931      else
1932          print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
1933      end
1934 end -- not latexmk
1935
1936 -- lwarf printindex:
1937 -- Compile the index then touch the source
1938 -- to trigger a recompile of the document:
1939
1940 elseif arg[1] == "printindex" then
1941 loadconf ()
1942 print ("lwarpmk: Processing the index.")
1943 os.execute("texindy -M lwarf_html.xdy" .. sourcename .. ".idx")
1944 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
1945 refreshdate ()
1946 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1947 print ("lwarpmk: Done.")
1948
1949 -- lwarpmk html:
1950
1951 elseif arg[1] == "html" then
1952 loadconf ()
1953 if ( uselatexmk == "true" ) then
1954     -- The recorder option is required to detect changes in <project>.tex
1955     -- while we are loading <project>.html.tex.
1956     err=os.execute ( "latexmk -pdf -dvi -ps -recorder "
1957         .. "-e "
1958         .. opquote .. "$makeindex = q/texindy -M lwarf_html.xdy/" .. opquote
1959         .. " -pdflatex="" .. latexname .. "%0 %S\" "
1960         .. sourcename .. "_html.tex" ) ;
1961     if ( err ~= 0 ) then print ("lwarpmk: Compile error.") ; os.exit(1) ; end
1962     pdftohtml ()
1963     print ("lwarpmk: Done.")
1964 else -- not latexmk
1965     verifyfileexists ( sourcename .. ".tex" ) ;
1966     -- See if exists and is up to date:
1967     if (
1968         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
1969         (
1970             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
```

```
1971         lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
1972     )
1973   ) then
1974     -- Recompile if not yet up to date:
1975     manytimes("_html")
1976     pdftohtml ()
1977     print ("lwarpmk: Done.")
1978   else
1979     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
1980   end
1981 end -- not latexmk
1982
1983 elseif arg[1] == "pdftohtml" then
1984   loadconf ()
1985   pdftohtml ()
1986
1987 -- lwarpmk htmlindex:
1988 -- Compile the index then touch the source
1989 -- to trigger a recompile of the document:
1990
1991 elseif arg[1] == "htmlindex" then
1992 loadconf ()
1993 print ("lwarpmk: Processing the index.")
1994 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. "_html.idx")
1995 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
1996 refreshdate ()
1997 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1998 print ("lwarpmk: Done.")
1999
2000 -- lwarpmk limages:
2001 -- Scan the lateximages.txt file to create lateximages,
2002 -- then touch the source to trigger a recompile.
2003
2004 elseif arg[1] == "limages" then
2005 loadconf ()
2006 print ("lwarpmk: Processing images.")
2007 createlateximages ()
2008 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2009 refreshdate ()
2010 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2011 print ("lwarpmk: Done.")
2012
2013 -- lwarpmk again:
2014 -- Touch the source to trigger a recompile.
2015
2016 elseif arg[1] == "again" then
2017 loadconf ()
2018 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2019 refreshdate ()
2020 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
```

```
2021 print ("lwarpmk: Done.")  
2022  
2023 -- lwarpmk clean:  
2024 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log  
2025  
2026 elseif arg[1] == "clean" then  
2027 loadconf ()  
2028 removeaux ()  
2029 print ("lwarpmk: Done.")  
2030  
2031 -- lwarpmk cleanall  
2032 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log  
2033 --     and also project.pdf, *.html  
2034  
2035 elseif arg[1] == "cleanall" then  
2036 loadconf ()  
2037 removeaux ()  
2038 os.execute ( rmname .. " " ..  
2039     sourcename .. ".pdf" .. sourcename .. "_html.pdf" ..  
2040     "*.html"  
2041     )  
2042 print ("lwarpmk: Done.")  
2043  
2044 -- lwarpmk with no argument :  
2045  
2046 elseif (arg[1] == nil) then  
2047 printhelp ()  
2048  
2049 -- lwarpmk -h or lwarpmk --help :  
2050  
2051 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then  
2052 printusage ()  
2053  
2054 else  
2055 print ("lwarpmk: Unknown command \"\"..arg[1].."\".\n")  
2056 printhelp ()  
2057 end  
2058  
2059 end -- not --version  
2060 \end{VerbatimOut}  
2061 % \end{Verbatim}% for syntax highlighting  
  
2062 \end{LWR@createlwarpmk}
```

## Package 3

# **lwarf-abstract.sty**

## 71 Abstract

Pkg **abstract** abstract is supported and patched by lwarf.

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

**for HTML output:** Accept all options for lwarf-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{\BlockClass{abstract}}
4 \AfterEndEnvironment{abstract}{\endBlockClass}
5 }
6
7 \renewcommand{\@bsrunintitle}{%
8 \hspace*{\abstitleskip}%
9 {\abstractnamefont}%
10 \InlineClass{abstractrunintitle}{\abstractname}%
11 \@bslabeldelim}%
12 }
13
14 \if@titlepage
15   \renewenvironment{abstract}{%
16 %     \titlepage
17     \null\vfil
18     \begin{par penalty}\lowpenalty
19     \if@bsrunin
20     \else
21       \if@bstitlestyle
22         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}%
23       \else
24         \ifnumber@bs
25           \num@bs
26         \else
27           \begin{\absnamepos}%
28           \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}%
29           \end{par penalty}\OM
30           \end{\absnamepos}%

```

```
31 %%          \vspace{\abstitleskip}%
32      \fi
33      \fi
34      \vspace{\abstitleskip}%
35  \fi
36  \put@bsintoc%
37  \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
38  {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
39 }
40 \else
41  \renewenvironment{abstract}{%
42      \if@bsrunin
43  \else
44      \if@bsstyle
45          \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}%
46      \else
47          \ifnumber@bs
48              \num@bs
49          \else
50  \begin{\absnamepos}%
51 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
52 \end\absnamepos%
53 %%          \vspace{\abstitleskip}%
54      \fi
55      \fi
56      \vspace{\abstitleskip}%
57  \fi
58  \put@bsintoc%
59  \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
60  {\par\end{@bstr@ctlist}}
61 \fi
62
```

## Package 4

### **l warp-afterpage.sty**

#### **72 Afterpage**

Pkg **afterpage** Not used.

**for HTML output:** Discard all options for l warp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
2 \newcommand{\afterpage}[1]{#1}
```

## Package 5

### **lwarf-algorithmicx.sty**

#### 73 Algorithmicx

Pkg **algorithmicx** algorithmicx is supported with minor adjustments.

**for HTML output:** 1 \LWR@ProvidesPackagePass{algorithmicx}

Inside the `algorithmic` environment, level indenting is converted to a `span` of the required length, and comments are placed inside a `span` which is floated right.

**⚠ package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [131.1](#).

**for HTML output:** 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*\ALG@doentity{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}{}}%
19 }

20 \end{warpHTML}

```

## Package 6

### l warp-alltt.sty

#### 74 Alltt

Pkg alltt alltt is patched for use by l warp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \AtBeginEnvironment{alltt}{\LWR@atbeginverbatim{alltt}\unskip\vspace*{-\baselineskip}}
4 \AfterEndEnvironment{alltt}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
5 }
```

## Package 7

### l warp-bookmark.sty

#### 75 Bookmark

Pkg bookmark bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for l warp-bookmark:

```
1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*\{\bookmarksetup}[1]{}
3 \newcommand*\{\bookmarksetupnext}[1]{}
4 \newcommand*\{\bookmark}[2][]{}
5 \newcommand*\{\bookmarkdefinestyle}[2]{}
6 \newcommand*\{\bookmarkget}[1]{}
7 \newcommand{\BookmarkAtEnd}[1]{}
```

## Package 8

### l warp-booktabs.sty

#### 76 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and the booktabs package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{booktabs}
```

Booktabs emulation is spread among the tabular code.

Emulated for source compatibility.

```
2 \newcommand*{\addlinespace}[1]{}
3 \newcommand*{\morecmidrules}{}
4 \newcommand*{\specialrule}[3]{}
```

## Package 9

### l warp-changepage.sty

#### 77 Changepage

Pkg changepage changepage is ignored.

for HTML output:

Discard all options for l warp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpage true}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

## Package 10

### l warp-dcolumn.sty

#### 78 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

## Package 11

### l warp-draftwatermark.sty

#### 79 Draftwatermark

Pkg draftwatermark draftwatermark is emulated during HTML output, and the draftwatermark package is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{draftwatermark}

```
2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}
```

## Package 12

### **lwarf-enumerate.sty**

#### 80    Enumerate

Pkg **enumerate** `enumerate` is ignored. `enumitem` is then modified per the `shortlabels` option.

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

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

**for HTML output:** Discard all options for `lwarf-enumerate`:

```
1 \LWR@ProvidesPackageDrop{enumerate}

2 % \DeclareOption{shortlabels}
3 %
4 \def\enit@shl#1{%
5   \ifnum\enit@type=\tw@
6     \enit@toks{\#1}%
7   \else
8     \def\enit@c{\#1}%
9     \enit@first#1,\@nil\@nil % Returns \enit@toks
10   \fi}
11 }
```

## Package 13

### **lwarf-epigraph.sty**

#### 81 Epigraph

Pkg **epigraph** `epigraph` is emulated during HTML output, and the `epigraph` package is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}
```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{\flushepinormal}{}{}
24 \newcommand{\textflush}[1]{\flushepinormal}
25 \newcommand{\epigraphflush}[1]{\flushright}
26 \newcommand{\sourceflush}[1]{\flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
```

```
32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1][]{}
```

## Package 14

### **lwarf-eso-pic.sty**

#### 82 Eso-pic

Pkg **eso-pic** *eso-pic* is emulated during HTML output, and the *eso-pic* package is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \let\AddToShipoutPicture\AddToShipoutPictureBG
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6][]{}
```

## Package 15

### l warp-everypage.sty

#### 83 Everypage

Pkg everypage everypage is emulated during HTML output, and the everypage package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{everypage}
2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}
```

## Package 16

### l warp-extramarks.sty

#### 84 Extramarks

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for l warp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}
2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}){}
4 \newcommand*{\lastleftxmark}){}
5 \newcommand*{\firstrightxmark}){}
6 \newcommand*{\lastrightxmark}){}
7 \newcommand*{\firstxmark}){}
8 \newcommand*{\lastxmark}){}
9 \newcommand*{\topxmark}){}
10 \newcommand*{\topleftxmark}){}
11 \newcommand*{\firstleftmark}){}
12 \newcommand*{\lastrightmark}{})
```

## Package 17

### **lwarf-fancyhdr.sty**

#### 85 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

**for HTML output:** Discard all options for lwarf-fancyhdr:

```
1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] []
3 \newcommand*{\fancyfoot}[2] []
4 \newcommand*{\fancyhf}[2] []
5 \newcommand*{\fancypagestyle}[2] []
6 \newcommand*{\lhead}[1] {}
7 \newcommand*{\chead}[1] {}
8 \newcommand*{\rhead}[1] {}
9 \newcommand*{\lfoot}[1] {}
10 \newcommand*{\cfoot}[1] {}
11 \newcommand*{\rfoot}[1] {}
12 \newcommand*{\headrulewidth} {}
13 \newcommand*{\footrulewidth} {}
14 \newcommand*{\fancyheadoffset}[2] []
15 \newcommand*{\fancyfootoffset}[2] []
16 \newcommand*{\fancyhfoffset}[2] []
17 \newcommand*{\iffloatpage}[2]{#2}
18 \newcommand*{\ifftopfloat}[2]{#2}
19 \newcommand*{\iffbotfloat}[2]{#2}
```

## Package 18

### **lwarf-float.sty**

#### 86 Float and \newfloat

Pkg **float** float is emulated during HTML output, and the float package is ignored.

for **HTML output**: 1 \LWR@ProvidesPackageDrop{float}[2016/03/04]

See section 52.2 for the \listof command.

\newfloat {\langle 1: type\rangle} {\langle 2: placement\rangle} {\langle 3: ext\rangle} [{\langle 4: within\rangle}]

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}{%
4 {%
5 \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
6 }%
7 {\DeclareFloatingEnvironment[fileext=#3]{#1}}}
```

newfloat package automatically creates the \listof command for new floats, but float does not, so remove \listof here in case it is manually created later.

```
8 \cslet{listof#1s}\relax
9 \cslet{listof#1es}\relax
10 }
```

\floatname {\langle type\rangle} {\langle name\rangle}

Sets the text name of the float, such as “Figure”.

```
11 \NewDocumentCommand{\floatname}{m +m}{%
12 \SetupFloatingEnvironment{#1}{name=#2}%
13 }
```

\floatplacement {\langle type\rangle} {\langle placement\rangle}

Float placement is ignored.

```
14 \newcommand*{\floatplacement}[2]{%
15 \SetupFloatingEnvironment{#1}{placement=#2}%
16 }

\floatstyle {⟨style⟩}
```

Float styles are ignored.

```
17 \newcommand{\floatstyle}[1]{%
18 }
```

```
\restylefloat * {⟨style⟩}
```

Float styles are ignored.

```
19 \NewDocumentCommand{\restylefloat}{s m}{%
20 }
```

## Package 19

### **l warp-floatrow.sty**

#### 87 Floatrow

Pkg **floatrow** floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output:

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

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

Only parameters for caption and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

```
2 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
3 \ifbool{LWR@insubfloatrow}
4 {
```

Inside a subfloatrow. subfigure and subtable take a width argument. Defaults to \linewidth if none is given.

```
5 \IfValueTF{#3}
6 {\@nameuse{sub#2}{#3}}
7 {\@nameuse{sub#2}{\linewidth}}
8 }
9 {
```

Not inside a subfloatrow. figure and table do not take a width argument.

```
10 \@nameuse{#2}
11 }
```

```

12 #6
13
14 #7
15 \ifbool{LWR@insubfloatrow}
16 {\@nameuse{endsub#2}}
17 {\@nameuse{end#2}}
18 }

```

Not used:

```

19 \newcommand*{\nocapbeside}={}
20 \newcommand*{\capbeside}={}
21 \newcommand*{\caption}={}
22 \newlength{\FBwidth}
23 \setlength{\FBwidth}{.3\linewidth}
24 \newlength{\FBheight}
25 \setlength{\FBheight}{2in}
26 \newcommand*{\useFCwidth}={}
27 \newcommand{\floatsetup}[2][]{}
28 \newcommand{\thisfloatsetup}[1]({})
29 \newcommand{\clearfloatsetup}[1]({})
30 \newcommand*{\killfloatstyle}={}

```

Preamble and default width are ignored.

```

31 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
32 \@namedef{#1}{%
33 \floatbox{#2}%
34 }%
35 }

```

Preamble and default width are ignored.

```

36 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
37 \@namedef{#1}{%
38 \floatbox{#2}%
39 }%
40 }

41 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []

```

```

42 \newfloatcommand{ttabbox}{table}[\caption][\FBwidth]
43 \newfloatcommand{fcapside}{figure}[\capbeside] []

```

The row of floats is placed into a div of class `floatrow`.

```
44 \newenvironment*{floatrow}[1][2]
```

```
45 {
46 \BlockClass{floatrow}
```

While inside the floatrow, divide the `\ linewidth` by the number of floats.

```
47 \booltrue{LWR@infloatrow}
48 \setlength{\linewidth}{6in/#1}
49 }
50 {
51 \boolfalse{LWR@infloatrow}
52 \endBlockClass
53 }
```

Keys for `\DeclareNewFloatType`:

```
54 \newcommand*{\LWR@frowkeyplacement}={}
55 \newcommand*{\LWR@frowkeyname}={}
56 \newcommand*{\LWR@frowkeyfileext}={}
57 \newcommand*{\LWR@frowkeywithin}={}
58 \newcommand*{\LWR@frowkeycapstyle}={}
59
60 \define@key{frowkeys}{placement}{}%
61 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
62 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
63 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
64 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use `\listof{type}{Title}` to print a list of the floats.

```
65 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
66 \renewcommand*{\LWR@frowkeyplacement}={}
67 \renewcommand*{\LWR@frowkeyname}={}
68 \renewcommand*{\LWR@frowkeyfileext}={}
69 \renewcommand*{\LWR@frowkeywithin}={}
70 \renewcommand*{\LWR@frowkeycapstyle}={}
```

Read new key values:

```
71 \LWR@traceinfo{about to setkeys frowkeys}%
72 \setkeys{frowkeys}{#2}%
73 \LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
74 \ifthenelse{\equal{\LWR@frowkeywithin}{}}
75 {
```

```

76 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
77 \LWR@frowkeyfileext}%
78 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
79 }%
80 {%
81 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
82 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
83 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeywithin}%
84 {\LWR@frowkeyfileext}{\LWR@frowkeywithin}%
85 \LWR@traceinfo{*** finished newfloat #1}%
86 }%

```

Rename the float if a name was given:

```

87 \ifthenelse{\equal{\LWR@frowkeyname}{}}%
88 {}%
89 {\floatname{#1}{\LWR@frowkeyname}}%
90 }%

```

Not used:

```

91 \newcommand{\buildFBB0X}[2]{}
92 \newcommand*{\CenterFloatBoxes}{}%
93 \newcommand*{\TopFloatBoxes}{}%
94 \newcommand*{\BottomFloatBoxes}{}%
95 \newcommand*{\PlainFloatBoxes}{}%
96
97 \newcommand{\capsubrowsettings}{}%
98
99 \NewDocumentCommand{\RawFloats}{o o}{}%

```

To be used inside a minipage or parbox.

```
100 \newcommand{\RawCaption}[1]{#1}
```

Places additional text inside a float, inside a CSS div of class `floatfoot`.

```

101 \NewDocumentCommand{\floatfoot}{s +m}{%
102 \begin{BlockClass}{floatfoot}%
103 #2%
104 \end{BlockClass}%
105 }%

```

Used to compute `\linewidth`.

```

106 \newbool{\LWR@insubfloatrow}%
107 \boolfalse{\LWR@insubfloatrow}%
108 \newenvironment*{subfloatrow}[1][2]%

```

```
109 {
```

The row of floats is placed into a `div` of class `floatrow`:

```
110 \BlockClass{floatrow}
```

While inside the floatrow, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
111 \begingroup
112 \booltrue{LWR@insubfloatrow}
113 }
114 {
115 \endgroup
116 \endBlockClass
117 \boolfalse{LWR@insubfloatrow}
118 }
```

## Package 20

### **l warp- fontenc.sty**

#### 88 Fontenc

Pkg **fontenc** Error if fontenc is loaded after l warp.

Discard all options for l warp- fontenc:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{fontenc}  
2 \LWR@loadbefore{fontenc}

## Package 21

### **l warp- fonts spec.sty**

#### 89 Fonts spec

Pkg **fonts spec** Error if fonts spec is loaded after l warp.

Discard all options for l warp- fonts spec:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{fonts spec}  
2 \LWR@loadbefore{fonts spec}

## Package 22

### **l warp-ftnright.sty**

#### 90 Ftnright

Pkg **ftnright** ftnright is ignored.

**for HTML output:** Discard all options for l warp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

## Package 23

### **l warp-geometry.sty**

#### 91 Geometry

Pkg **geometry** geometry is preloaded by l warp, but must be nullified as seen by the user's source code.

**for HTML output:** Discard all options for l warp-geometry:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*\{\geometry}[1]{}
3 \renewcommand*\{\newgeometry}[1]{}
4 \renewcommand*\{\restoregeometry}{}
5 \renewcommand*\{\savegeometry}[1]{}
6 \renewcommand*\{\loadgeometry}[1]{}
```

## Package 24

### l warp-graphics.sty

#### 92 Graphics

Pkg `graphics` `graphics` is not used. The user is required to use `graphicx` instead.

for HTML output:

```
1 \PackageError{l warp-graphics}{  
2 {The graphics package is not supported by  
3 the l warp package's HTML conversion.  
4 Please use graphicx instead. \protect\usepackage{graphicx}}  
5 {Graphicx requires a different syntax but is more flexible,  
6 and it is emulated by l warp.}}
```

## Package 25

### l warp-graphicx.sty

#### 93 Graphicx

Pkg `graphicx` `graphicx` is emulated during HTML output, and the `graphicx` package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{graphicx}
```

## Package 26

### **lwarf-hyperref.sty**

#### 94 Hyperref

Pkg **hyperref** hyperref is emulated during HTML output, and the hyperref package is ignored.

**for HTML output:**

```

1 \% \LWR@ProvidesPackageDrop{hyperref}
2 \typeout{Using the lwarf html version of package 'hyperref' -- discarding options.}
3 \typeout{    Are not using ProvidesPackage, so that other packages}
4 \typeout{    do not attempt to patch lwarf's version of 'hyperref'.}
5 \% \ProvidesPackage{lwarf-\#1-\#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*\{\hypersetup}[1]{}
9 \newcommand*\{\hyperbaseurl}[1]{}

```

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmntag{img src="#1" alt="#2" class="hyperimage"}{}}%
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }
15
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode`\_=12
19 \LWR@hyperimageb%
20 }

```

Creates an HTML anchor to `category.name` with the given text.

```

21 \NewDocumentCommand{\hyperdef}{m m +m}{%
22 \LWR@ensuredoingapar%
23 \LWR@subsublabel{\#1.\#2}%
24 #3%
25 }

```

Creates an HTML link to `URL#category.name` with the given text.

```

26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmntag{a href="#1\LWR@hashmark\#2.\#3"}%

```

```

28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }

```

Creates text as an HTML link to the L<sup>A</sup>T<sub>E</sub>X label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{0{label} +m}{
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

```

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode`\_=12%
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

Creates an anchor to `name` with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref[#1]{#2}%
49 }

```

For HTML, `\cleverref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleverref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

56 \newcommand{\pdfstringdef}[2]{}}

57 \newcommand{\pdfbookmark}[3]{}{}

```

```
58 \newcommand{\currentpdfbookmark}[2]{}
59 \newcommand{\subpdfbookmark}[2]{}
60 \newcommand{\belowpdfbookmark}[2]{}
61 \newcommand{\texorpdfstring}[2]{#2}
```

From `hyperref`.

```
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%
65 \newcommand{\Acrobatmenu}[2]{}
66 \newcommand*{\TextField}[2][]{}
67 \newcommand*{\CheckBox}[2][]{}
68 \newcommand{\ChoiceMenu}[3][]{}
69 \newcommand*{\PushButton}[2][]{}
70 \newcommand*{\Submit}[2][]{}
71 \newcommand*{\Reset}[2][]{}
72 \newcommand*{\LayoutTextField}[2]{}
73 \newcommand*{\LayoutChoiceField}[2]{}
74 \newcommand*{\LayoutCheckField}[2]{}
75 \newcommand*{\MakeRadioField}[2]{}
76 \newcommand*{\MakeCheckField}[2]{}
77 \newcommand*{\MakeTextField}[2]{}
78 \newcommand*{\MakeChoiceField}[2]{}
79 \newcommand{\MakeFieldButton}[1]{}
```

## Package 27

### **l warp-indentfirst.sty**

#### **95 Indentfirst**

Pkg indentfirst indentfirst is ignored.

Discard all options for l warp-indentfirst:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{indentfirst}

## Package 28

### **l warp-inputenc.sty**

#### **96 Inputenc**

Pkg inputenc Error if inputenc is loaded after l warp.

Discard all options for l warp-inputenc:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

## Package 29

### lwarf-keyfloat.sty

#### 97 Keyfloat

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output:

```
1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

2 \AtBeginDocument{

3 % \newsavebox{\KFLT@envbox}
4
5 \let\KFLT@boxinner\relax
6 \let\endKFLT@boxinner\relax
7
8 \NewEnviron{KFLT@boxinner}
9 {%
10 \LWR@traceinfo{kflt@boxinner}%
11 \LWR@stoppars%
12 \KFLT@frame{\BODY}%
13 \LWR@startpars%
14 \LWR@traceinfo{ended kflt@boxinner}%
15 }

16 \RenewDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
17 {%
18 \LWR@maybeinthisfloat%
19 % \BlockClass{marginblock}
20 \LWR@stoppars%
21 \LWR@htmntag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
22 \LWR@startpars%
23 \captionsetup{type=#2}%
24 }
25 {
26 % \endBlockClass
27 \LWR@htmldivclassend{div}
28 }

29 }% AtBeginDocument
```

## Package 30

### **l warp-layout.sty**

#### 98 Layout

Pkg **layout** layout is ignored.

**for HTML output:** Discard all options for l warp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}{}
```

## Package 31

### **l warp-letterspace.sty**

#### 99 Letterspace

Pkg **letterspace** letterspace is a subset of microtype, which is pre-loaded by l warp. All user options and macros are ignored and disabled.

**for HTML output:** Discard all options for l warp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][]{}
4 \def\textls#1{}
5 \newcommand*\lslig[1]{#1}
```

## Package 32

### **lwarf-listings.sty**

#### 100 Listings

Pkg `listings` `listings` is supported with some limitations. Text formatting is not yet supported.

for HTML output:

```
1 \begin{warpHTML}
2 \LWR@ProvidesPackagePass{listings}
```

Patches to embed listings inside `pre` tags:

```
3 \let\LWR@origlst@Init\lst@Init
4 \let\LWR@origlst@DeInit\lst@DeInit
5
6 \let\LWR@origlst@EveryPar\lst@EveryPar
7
8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{toclstlisting}{#1}{#2}}
```

Done at the start of a listing.

```
9 \renewcommand{\lst@Init}[1]{%
```

First, perform the `listings` initialization:

```
10 \LWR@traceinfo{lst@Init}%
11 \renewcommand*{@capttype}{lstlisting}%
12 \LWR@origlst@Init{#1}%
13 \LWR@traceinfo{finished origlst@Init}%
14 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `<pre>`, then reenable line numbers.

```
15 \LWR@traceinfo{About to create verbatim.}%
16 \let\lst@EveryPar\relax%
17 \LWR@atbeginverbatim{programlisting}%
18
19 \let\lst@EveryPar\LWR@origlst@EveryPar%
20 \else%
```

Inline, so open a `<span>`

```

21 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}{}}{%
22 \fi%
23 }

24 \renewcommand*\lst@DeInit{%
25 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```

26 \let\lsthk@EveryPar\relax%
27
28 \LWR@afterendverbatim%
29 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
30 \else%

```

Inline, so create the closing </span>:

```

31 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}{}}{%
32 \fi%

```

Final listings deinit:

```

33 \LWR@origlst@DeInit%
34 }

```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```

35 \def\lst@MakeCaption#1{%
36 \LWR@traceinfo{MAKING CAPTION at #1}%
37 \lst@ifdisplaystyle
38 \LWR@traceinfo{making a listings display caption}%
39 \ifx #1t%
40 \ifx\lst@@caption\empty\expandafter\lst@HRefStepCounter \else
41 \expandafter\refstepcounter
42 \fi {lstlisting}%
43 \LWR@traceinfo{About to assign label: !\lst@label!}%
44 \ifx\lst@label\empty\else
45 \label{\lst@label}\fi
46 \LWR@traceinfo{Finished assigning the label.}%
47 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
48 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
49 \lst@ifnolol\else
50 \ifx\lst@@caption\empty
51 \ifx\lst@caption\empty
52 \ifx\lst@intname\empty \else \def\lst@temp{ }%

```

```
53           \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for `lwarp`:

```
54 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
55 %                         \addcontentsline{loll}{lstlisting}{\lst@name}
56         \fi\fi
57     \fi
58 \else
```

This would have to be modified for `lwarp`:

```
59 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
60             \addcontentsline{loll}{lstlisting}%
61 {\protect\numberline{\thelstlisting}%
62 {\protect\ignorespaces \lst@@caption \protect\relax}}%
63         \fi
64     \fi
65     \fi
66     \ifx\lst@caption\@empty\else
67 \LWR@traceinfo{lst@caption not empty-}%
68     \list@ifSubstring #1\lst@captionpos
69     {\begin{group}
70 \LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
71 %          \let\@vskip\vskip
72 %          \def\vskip{\afterassignment\lst@vskip \tempskipa}%
73 %          \def\lst@vskip{\nobreak\@vskip\tempskipa\nobreak}%
74 %          \par\@parboxrestore\normalsize\normalfont \% \noindent (AS)
75 %          \ifx #1t\allowbreak \fi
76          \ifx\lst@title\@empty
```

New `lwarp` code to create a caption:

```
77           \list@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}%
78           \else
```

New `lwarp` code to create a title:

```
79 %          \list@maketitle\lst@title \% (AS)
80 \LWR@traceinfo{Making title: \lst@title}%
81 \begin{BlockClass}{lstlistingtitle}\% lwarp
82 \list@maketitle\lst@title\% lwarp
83 \end{BlockClass}\% lwarp
84         \fi
85 \LWR@traceinfo{About to assign label: !\lst@label!}%
```

```

86      \ifx\lst@label\empty\else
87 \leavevmode% gets rid of bad space factor error
88 \GetTitleStringExpand{\lst@caption}%
89 \edef\LWR@lntemp{\GetTitleStringResult}%
90 \edef@\currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
91 \label{\lst@label}\fi
92 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for \l warp:

```

93 %           \ifx #1b\allowbreak \fi
94           \endgroup{}%
95           \fi
96 \LWR@traceinfo{end of making a listings display caption}%
97   \else
98 \LWR@traceinfo{INLINE}%
99   \fi
100 \LWR@traceinfo{DONE WITH CAPTION at #1}%
101 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

102 \lst@Key{numbers}{none}{%
103   \let\lst@PlaceNumber\empty
104   \lstKV@SwitchCases{#1}%
105   {none&\%
106     left&\def\lst@PlaceNumber{%
107 % \llap{
108 \LWR@orignormalfont%
109 \lst@numberstyle{\the\lstnumber}\kern\lst@numbersep%
110 % }
111 }
112 \%
113   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
114     \kern\VerbatimHTMLWidth \kern\lst@numbersep
115     \lst@numberstyle{\the\lstnumber}}}%
116   }{\PackageError{Listings}{Numbers #1 unknown}@\ehc}%
117
118 \end{warpHTML}

```

## Package 33

# **lwarf-longtable.sty**

## 101 Longtable

Pkg **longtable** longtable is emulated during HTML output, and the longtable package is ignored.

for **HTML output:** 1 \LWR@ProvidesPackageDrop{longtable}

To emulate longtable:

For longtable \endhead, \endfoot, and \endlastfoot rows, use

 \warpprintonly{row contents}

instead of

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

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

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

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env **longtable** \* [*horizontalignment*] {*colspec*} Emulates the **longtable** environment.

Per the **caption** package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```

2 \newenvironment{longtable*}[2] []{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaptype}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{\LWR@starredlongtable}%
9 \let\captionlistentry{\LWR@LTcaptionlistentry}%
10 \LWR@tabular{#2}%
11 }%
12 {\endLWR@tabular\endLWR@floatend}%
13 %
14 \newenvironment{longtable}[2] []{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaptype}%

```

```
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaptere}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\endLWR@floatend}
25
```

Provided for compatibility, but ignored:

```
26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% throws away options // [dim] and // *
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTright}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

\kill is ignored, place a \kill line inside
\begin{warpprint} ... \end{warpprint} or \warpingprintonly.

38 \renewcommand*{\kill}{\LWR@tabularendofline}
```

## Package 34

### **l warp-l scape.sty**

#### 102 L scape

Pkg l scape l scape is nullified.

for HTML output: Discard all options for l warp-l scape.

```
1 \LWR@ProvidesPackageDrop{l scape}
2 \newenvironment*{landscape}{}{}
```

## Package 35

### **l warp-l tcaption.sty**

#### 103 Lt caption

Pkg lt caption lt caption is emulated during HTML output, and the lt caption package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{lt caption}

\LTcaptype is already defined by l warp.

longtable\* is already defined by l warp-longtable.

```
2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginsfalse}{}{}
```

## Package 36

### **lwarf-mcaption.sty**

#### **104 Mcaption**

Pkg **mcaption** **mcaption** is nullified.

**for HTML output:** Discard all options for lwarf-mcaption:

```
1 \LWR@ProvidesPackageDrop{mcaption}
2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapssep}
```

## Package 37

# l warp-mdframed.sty

## 105 Mdframed

Pkg **mdframed** `mdframed` is loaded with options forced to `framemethod=none`.

for **HTML output**: 1 `\LWR@ProvidesPackageDrop{mdframed}`

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

**⚠ loading** When used, l warp loads `mdframed` in HTML with `framemethod=none`.

For title font, use

`frametitlefont=\textbf,`

**font** instead of

`frametitlefont=\bfseries,`

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

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

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

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

**CSS classes** Environments created or encapsulated by `mdframed` are enclosed in a `div` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `span` of class `mdframedtitle`. Subtitles are in a `span` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

`amsthm` must be loaded before `mdframed`

2 `\LWR@origRequirePackage{amsthm}`

Do not require Tikz or pstricks:

```
3 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents:

```
4 \mdfsetup{  
5 startcode={\LWR@mdframedstart},  
6 endcode={\LWR@mdframedend},  
7 startinnercode={\LWR@startpars},  
8 endinnercode={\LWR@stoppars},  
9 }
```

Given the `mdframed` key, print the color.

```
10 \newcommand*{\LWR@mdfprintcolor}[1]{%  
11 \convertcolorspec{named}{\csuse{mdf@\#1}}{HTML}\LWR@tempcolor%  
12 \#\LWR@tempcolor  
13 }
```

Given the `mdframed` key, print the length.

```
14 \newcommand*{\LWR@mdfprintlength}[1]{%  
15 \rndprintlength{\csuse{mdf@\#1@length}}  
16 }
```

Actions before an mdframe starts.

Encapsulate a frame inside a `div` of the desired `class`.

```
17 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
18 \LWR@stoppars%
```

Below, print HTML pt units:

```
19 \uselengthunit{PT}%
```

Open a `div` and with custom `class` and custom `style`:

```
20 \LWR@htmllagc{div class="\LWR@mdthisenv" \LWR@orignewline  
21 style="" \LWR@orignewline
```

Convert and print the background color:

```
22 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
23 border: \LWR@mdfprintlength{linewidth} solid
24 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
25 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
26 \ifbool{mdf@shadow}{%
27 box-shadow:
28 \LWR@mdfprintlength{shadowsize}
29 \LWR@mdfprintlength{shadowsize}
30 \LWR@mdfprintlength{shadowsize}
31 \LWR@mdfprintcolor{shadowcolor} ;
32 }
33 {box-shadow: none ;}
34 \LWR@orignewline

35 "}
36 % \LWR@htmldivclass{\LWR@mdthisenv}
```

**mdframed** environment may not work with the modified `\hspace` and `\rule`, so restore them to their originals while inside **mdframed**:

```
37 \let\hspace\LWR@orighspace%
38 \let\rule\LWR@origrule%
39 }
```

Actions after an mdframe ends.

After closing the `div`, globally restore to the default environment type:

```
40 \newcommand*{\LWR@mdframedend}{
```

Close the custom `div`:

```
41 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
42 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
43 \LWR@startpars%
44 }
```

Encapsulation of the original which places the title inside a `span` of class `mdframedtitle`:

```
45 \let\LWR@origmdfframedtitleenv\mdfframedtitleenv
46
47 \newlength{\LWR@titleroundcorner}
48
49 \renewrobustcmd\mdfframedtitleenv[1]{%
50 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
51 \uselengthunit{PT}{%
```

Open a `span` with a custom `class` and custom `style`:

```
52 \LWR@htmtagc{span class="mdframedtitle" \LWR@orignewline
53 style=" \LWR@orignewline
```

Convert and print the title background color:

```
54 background:
55 \LWR@mdfprintcolor{frametitlebackgroundcolor}
56 ; \LWR@orignewline
```

Convert and print the title rule:

```
57 \ifbool{mdf@frametitlerule}{%
58 border-bottom:
59 \LWR@mdfprintlength{frametitlerulewidth}
60 solid
61 \LWR@mdfprintcolor{frametitlerulecolor}
62 ; \LWR@orignewline
63 }{}
```

The title's top border radius is adjusted for the line width:

```
64 border-radius:
65 \setlength{\LWR@titleroundcorner}
66 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
67 \rndprintlength{\LWR@titleroundcorner}
68 \rndprintlength{\LWR@titleroundcorner}
69 Opt Opt
70 \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
71 " \LWR@orignewline
72 }% span
```

Restrict paragraph tags inside a span:

```
73 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
74 #1%
```

Closee the span and unnest the paragraph tag restriction:

```
75 \LWR@htmlltagc{/span}%
76 \end{LWR@nestspan}%
77 }
78 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `span` of class `mdframedsubtitle`:

```
79 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}%
80 {%
81   the following empty line is required
82 }
```

Special handling for mdframed: Subtitles have `\pars` around them, so temporarily disable them here.

```
82 \let\par\LWR@origpar%
```

Open a `span` with a custom `class` and custom `style`:

```
83 \LWR@htmlltagc{span class="mdframed#1title"
84 style=" \LWR@orignewline
```

Convert and print the background color:

```
85 background:
86 \LWR@mdfprintcolor{#1titlebackgroundcolor}
87 ; \LWR@orignewline
```

Convert and print the above line:

```
88 \ifbool{mdf@#1titleaboveline}{%
89 border-top:
90 \LWR@mdfprintlength{#1titleabovelinewidth}
91 solid
92 \LWR@mdfprintcolor{#1titleabovelinecolor}
93 ; \LWR@orignewline
94 }{}
```

Convert and print the below line:

```

95 \ifbool{mdf@#1titlebelowline}{%
96 border-bottom:
97 \LWR@mdfprintlength{#1titlebelowlinewidth}
98 solid
99 \LWR@mdfprintcolor{#1titlebelowlinecolor}
100 ; \LWR@orignewline
101 }{}%
```

Finish the custom style and the opening span tag:

```
102 "}% span
```

Restrict paragraph tags inside a span:

```
103 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

104 \IfNoValueTF{#2}
105 {\csuse{LWR@origmdf#1title}{#3}}%
106 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

107 \LWR@htmntagc{/span}%
the following empty line is required
108 \end{LWR@nestspan}%
must follow the /span or an extra <p> appears
109
110 }

111 \let\LWR@origmdfsubtitle\mdfsubtitle
112
113 \newcommand*\{\LWR@mdfsubtitle}{%
114 \LWR@mdfsubtitlecommon{sub}%
115 }
116 \let\mdfsubtitle\LWR@mdfsubtitle

117 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
118
119 \newcommand*\{\LWR@mdfsubsubtitle}{%
120 \LWR@mdfsubtitlecommon{subsub}%
121 }
122 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```
123 \newcommand*\{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

124 \renewrobustcmd*\newmdenv[2] [] {%
125 \newenvironment{#2}%
126 {%
127 \mdfsetup{#1}%
128 \renewcommand*{\LWR@mdthisenv}{md#2}%
129 \begin{mdframed}%
130 }%
131 {\end{mdframed}}%
132 }

```

Modified from the original to remember the environment.

```

133 \renewrobustcmd*\surroundwithmdframed}[2] [] {%
134 \BeforeBeginEnvironment{#2}%
135 \renewcommand*{\LWR@mdthisenv}{md#2}%
136 \begin{mdframed}[#1]}%
137 \AfterEndEnvironment{#2}{\end{mdframed}}%
138 }

```

[*numberedlike*] {[*caption*] [*within*]}

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{O{} m o m o}{%
142 \LWR@origmdtheorem[#1]{#2}[#3]{#4}[#5]%
143 \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}%
144 }%
145
146 \let\mdtheorem\LWR@mdtheorem

```

[*numberedlike*] {[*caption*] [*within*]}

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
148 \ifboolexpr{ test {\IfNoValueTF{#3}} and test {\IfNoValueTF{#5}} }{%
149   \newtheorem{#2}{#4}%
150   \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
151   \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
152   }%
153 \BeforeBeginEnvironment{#2}{%
154 \renewcommand*{\LWR@mdthisenv}{md#2}%
155 \begin{mdframed}[#1]}%
156 \AfterEndEnvironment{#2}{%
157 \end{mdframed}}%
158 }

```

## Package 38

### **lwarf-microtype.sty**

#### 106 Microtype

Pkg **microtype** microtype is pre-loaded by lwarf. All user options and macros are ignored and disabled.

for **HTML output**: Discard all options for lwarf-microtype:

```
1 \LWR@ProvidesPackageDrop{microtype}

2 \renewcommand*\DeclareMicrotypeSet[3][]{}
3 \renewcommand*\UseMicrotypeSet[2][]{}
4 \renewcommand*\DeclareMicrotypeSetDefault[2][]{}
5 \renewcommand*\SetProtrusion[3][]{}
6 \renewcommand*\SetExpansion[3][]{}
7 \renewcommand*\SetTracking[3][]{}
8 \renewcommand*\SetExtraKerning[3][]{}
9 \renewcommand*\SetExtraSpacing[3][]{}
10 \renewcommand*\DisableLigatures[2][]{}
11 \renewcommand*\DeclareCharacterInheritance[3][]{}
12 \renewcommand*\DeclareMicrotypeVariants[1]({})
13 \renewcommand*\DeclareMicrotypeAlias[2]({})
14 \renewcommand*\LoadMicrotypeFile[1]({})
15 \renewcommand*\DeclareMicrotypeBabelHook[2]({})
16 \renewcommand*\microtypesetup[1]({})
17 \renewcommand*\microtypecontext[1]({})
18 \renewcommand*\textmicrotypecontext[2]{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \renewcommand*\lsstyle{}}
21 \renewcommand\textls[2][]{}
22 \def\textls#1#2{#2}
23 \renewcommand*\lslig[1]{#1}
24 }
25 \def\DeclareMicrotypeSet#1{\@gobbletwo}
26 \def\DeclareMicrotypeVariants#1{\@gobble}
27 \onlypreamble\DeclareMicrotypeSet
28 \onlypreamble\UseMicrotypeSet
29 \onlypreamble\DeclareMicrotypeSetDefault
30 \onlypreamble\DisableLigatures
31 \onlypreamble\DeclareMicrotypeVariants
32 \onlypreamble\DeclareMicrotypeBabelHook
```

## Package 39

### **lwarf-multicol.sty**

#### 107 Multicol

Pkg **multicol** multicol is emulated during HTML output, and the multicol package is ignored.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a div of class `multicolsheading`.

The content is placed inside a div of class `multicols`.

```
2 \begin{warpHTML}
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML div class to contain everything:

```
4 {\BlockClass{multicols}}
```

Optional HTML div class for the heading:

```
5 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
6 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
7 \newcommand*{\columnbreak}{}
8 \newcommand*{\RLmulticolcolumns}={}
9 \newcommand*{\LRmulticolcolumns}={}
10
11 \newlength{\premulticols}
12 \newlength{\postmulticols}
13 \newlength{\multicolsep}
14 \newlength{\multicolbaselineskip}
15 \newlength{\multicoltolerance}
16 \newlength{\multicolpretolerance}
17 \newcommand*{\columnseprulecolor}{\normalcolor}
```

```
18 \newcounter{columnbadness}
19 \newcounter{finalcolumnbadness}
20 \newcounter{collectmore}
21 \newcounter{unbalance}
22 \newlength{\multicolovershoot}
23 \newlength{\multicolundershoot}

24 \end{warpHTML}
```

## Package 40

### l warp-multirow.sty

#### 108 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

## Package 41

### l warp-nameref.sty

#### 109 Nameref

Pkg nameref nameref is emulated by l warp.

for HTML output: Discard all options for l warp-nameref:

```
1 \typeout{Using the l warp html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch l warp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

## Package 42

### l warp-needspace.sty

#### 110 Needspace

Pkg **needspace** needspace is not used during HTML conversion.

**for HTML output:** Discard all options for l warp-needspace:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}  
}
```

## Package 43

### l warp-newclude.sty

#### 111 Newclude

Pkg **newclude** Error if newclude is loaded after l warp.

Discard all options for l warp-newclude:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{newclude}  
2 \LWR@loadbefore{newclude}

## Package 44

### l warp-newunicodechar.sty

#### 112 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after l warp.

Discard all options for l warp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}  
2 \LWR@loadbefore{newunicodechar}

## Package 45

### l warp-nextpage.sty

#### 113 Nextpage

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for l warp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}  
2 \newcommand{\cleartoevenpage}[1] {}  
3 \newcommand{\movetoevenpage}[1] {}  
4 \newcommand{\cleartooddpage}[1] {}  
5 \newcommand{\movetooddpage}[1] {}

## Package 46

### **l warp-nowidow.sty**

#### 114 Nowidow

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for l warp-nowidow:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}
2 \newcommand*{\nowidow}[1] []
3 \newcommand*{\setnowidow}[1] []
4 \newcommand*{\noclub}[1] []
5 \newcommand*{\setnoclub}[1] []
```

## Package 47

### **l warp-pagenote.sty**

#### 115 Pagenote

Pkg pagenote pagenote is used by l warp, and its user-interface macros have been nullified.

Both footnotes and pagenotes appear as footnotes in the HTML output.

Any manipulation of pagenote in the user's document after the preamble should be encased within a warpprint environment.

Discard all options for l warp-pagenote:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{pagenote}
\pagenote is already modified to work with l warp.

2 \renewcommand*{\makepagenote} {}
3 \RenewDocumentCommand{\printnotes}{s} {}
4 \renewcommand{\printnotes}[2] []
5 \renewcommand{\addtonotes}[1] {}
```

## Package 48

### **l warp-parskip.sty**

#### **116 Parskip**

Pkg **parskip** pars skip is ignored.

**for HTML output:** Discard all options for l warp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

## Package 49

### **l warp-placeins.sty**

#### **117 Placeins**

Pkg **placeins** placeins is not used during HTML conversion.

Discard all options for l warp-placeins:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{placeins}

```
2 \newcommand*{\FloatBarrier}{}{}
```

## Package 50

### l warp-ragged2e.sty

#### 118 Ragged2e

Pkg **ragged2e** ragged2e is not used during HTML conversion.

Discard all options for l warp-ragged2e:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}
```

## Package 51

### **lwarf-rotating.sty**

#### 119 Rotating

Pkg **rotating** rotating is emulated during HTML output, and the rotating package is ignored.

This code is untested.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotating}
2 \let\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \let\sidewaysfigure\figure
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption
```

## Package 52

### **lwarf-setspace.sty**

#### 120 Setspace

Pkg **setspace** setspace is not used during HTML conversion.

Discard all options for lwarf-setspace:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {\BlockClass{singlespace}}
11 {\endBlockClass}
12
13 \newenvironment*{singlespace*}
14 {\BlockClass{singlespace*}}
15 {\endBlockClass}
16
17 \newenvironment*{spacing}[1]{}
18
19 }{
20
21 }
22
23 \newenvironment*{onehalfspace}
24 {\BlockClass{onehalfspace}}
25 {\endBlockClass}
26
27 \newenvironment*{doublespace}
28 {\BlockClass{doublespace}}
29 {\endBlockClass}
```

## Package 53

### **l warp-showkeys.sty**

#### **121 Showkeys**

Pkg showkeys showkeys is ignored.

**for HTML output:** Discard all options for l warp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
2 \NewDocumentCommand{\showkeys}{s}{}{}
```

## Package 54

### **l warp-sidecap.sty**

#### 122 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for l warp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

[http://tex.stackexchange.com/questions/45401/  
use-the-s-star-argument-with-newdocumentenvironment](http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment)  
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpx {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {\endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpx {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {\endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

## Package 55

### **l warp-tabularx.sty**

#### **123 Tabularx**

Pkg tabularx tabularx is emulated by l warp.

**for HTML output:** Discard all options for l warp-tabularx:

```
1 \LWR@ProvidesPackageDrop{tabularx}
2 \newenvironment{tabularx}[2]
3 {\begin{array}{#2}}
4 {\end{array}}
5
6 \newenvironment{tabularx*}[2]
7 {\begin{array}{#2}}
8 {\end{array}}
```

## Package 56

### **lwarf-textpos.sty**

#### 124 Textpos

Pkg **textpos** `textpos` is emulated during HTML output, and the `textpos` package is ignored.

for **HTML output:**

```
1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r(){}{}}
3 \NewDocumentEnvironment{textblock*}{m o r(){}{}}
4 \newcommand*\{\\TPGrid}[3][]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*\{\\textblockcolour}[1]{}
7 \newcommand*\{\\textblockrulecolour}[1]{}
8 \newcommand*\{\\textblockcolor}[1]{}
9 \newcommand*\{\\textblockrulecolor}[1]{}
10 \newcommand*\{\\tekstblokkulur}[1]{}
11 \newcommand*\{\\tekstblokrulekulur}[1]{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{}
16 \newcommand*\{\\showtextsize}{}
17 \newcommand{\textblockorigin}[2]{}
```

## Package 57

### lwarf-threeparttable.sty

#### 125 Threeparttable

Pkg **threeparttable** threeparttable is emulated during HTML output, and the threeparttable package is ignored.

for **HTML output:** 1 \LWR@ProvidesPackageDrop{threeparttable}

Prints the table note item header inside a CSS class of `tnoteitemheader`.

2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}

To emulate threeparttable:

3 \newenvironment\*{threeparttable}[1][b]{}{}

4 \newenvironment\*{tablenotes}[1][]  
5 {}%  
6 \BlockClass{tnotes}-%  
7 \setlist[description]{format=\LWR@printtablenote}-%  
8 \description%  
9 }%  
10 {}%  
11 \enddescription%  
12 \endBlockClass%  
13 }

14 \newcommand{\tnote}[1]{\textsuperscript{#1}}

## Package 58

### **lwarf-titleps.sty**

#### **126 Titleps**

Pkg **titleps** titleps is loaded and used by lwarf during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarf-titleps:

**for HTML output:** 1 \LWR@ProvidesPackageDrop{titleps}

\pagestyle and \thispagestyle are already disabled in the lwarf code.

```
2 \RenewDocumentCommand{\newpagestyle}{m o m}{}  
3 \RenewDocumentCommand{\renewpagestyle}{m o m}{}  
  
4 \RenewDocumentCommand{\sethead}{o o o m m m}{}  
5 \RenewDocumentCommand{\setfoot}{o o o m m m}{}  
  
6 \RenewDocumentCommand{\settitemarks}{s m}{}  
  
7 \renewcommand*\headrule{}  
8 \renewcommand*\footrule{}  
  
9 \renewcommand*\setheadrule[1]{}  
10 \renewcommand*\setfootrule[1]{}  
  
11 \newcommand*\makeheadrule{}  
12 \newcommand*\makefootrule{}  
  
13 \renewcommand{\setmarkboth}[1]{}  
  
14 \RenewDocumentCommand{\widenhead}{s o o m m}{}  
  
15 \renewcommand*\bottitemarks{}  
16 \renewcommand*\toptitemarks{}  
17 \renewcommand*\firstrtitlemarks{}  
18 \renewcommand*\nexttoptitemarks{}  
19 \renewcommand*\outertitlemarks{}  
20 \renewcommand*\innertitlemarks{}  
  
21 \RenewDocumentCommand{\newtitlemark}{s m}{}  

```

```
22 \RenewDocumentCommand{\pretitlemark}{s m m}{}  
23 \renewcommand{\ifsamemark}[4]{}  
  
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}  
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}  
  
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}  
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}  
  
28 \newcommand{\newmarkset}[1]{}  
  
29 \NewDocumentCommand{\newextramarkset}{s m m}{}  
  
30 \newcommand{\botextramarks}[1]{}  
31 \newcommand{\topextramarks}[1]{}  
32 \newcommand{\firstextramarks}[1]{}  
33 \newcommand{\nexttopextramarks}[1]{}  
34 \newcommand{\outerextramarks}[1]{}  
35 \newcommand{\innerextramarks}[1]{}  

```

## Package 59

### l warp-titlesec.sty

#### 127 Titlesec

Pkg **titlesec** titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for l warp-titlesec:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

2 \newcommand*\{\titlelabel}[1]{}

3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}{%
5     {\ttl@format@i}}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*\{\filright}{}
12 \newcommand*\{\filcenter}{}
13 \newcommand*\{\filleft}{}
14 \newcommand*\{\fillast}{}
15 \newcommand*\{\filinner}{}
16 \newcommand*\{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \oplus
18   \fontdimen\thr@@\font \ominus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*\{\ttl@rule}[1][]{}
22 \newcommand*\{\ttl@row}[2][]{}

23 \newcommand{\iftitlemeasuring}[2]{#2}

24 \newcommand{\assignpagestyle}[2]{#2}

25 \NewDocumentCommand{\titleclass}{m o m o}
```

## Package 60

### **lwarf-titletoc.sty**

#### 128 Titletoc

Pkg **titletoc** titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarf-titletoc:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{}{}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o o}){}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m o}{}{}

6 \newcommand{\contentsmargin}[2][]{}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2][]{\thecontentslabel}
10 \newcommand{\contentspage}[1][]{\thecontentspage}

11 \newcommand{\contentspush}[1]{{}

12 \newcommand{\contentsuse}[2]{{}

13 \newcommand*{\startcontents}[1][]{}
14 \newcommand*{\stopcontents}[1][]{}
15 \newcommand*{\resumecontents}[1][]{}

16 \newcommand{\printcontents}[4][]{}

17 \newcommand{\startlist}[2][]{}
18 \newcommand{\stoplist}[2][]{}
19 \newcommand{\resumelist}[2][]{}

20 \newcommand{\printlist}[4][]{}
```

## Package 61

### **lwarf-titling.sty**

#### **129 Titling**

Pkg **titling** titling is used by lwarf. The following patches are not needed by lwarf, but are required if the user requests titling.

lwarf uses page notes for footnotes, so the various titling footnote restyling commands have no effect.

Pass all options to lwarf-titling:

for **HTML output:** 1 \LWR@ProvidesPackagePass{titling}

Patch \@bsmtitleempty:

```
2 \let\LWR@orig@\bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@\bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch \keepthetitle:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*{\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch \killtitle:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*{\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```

## Package 62

### **lwarf-tocloft.sty**

#### 130 Tocloft

Pkg **tocloft** tocloft is emulated. Most user options and macros are ignored and disabled.  
\newlistof and \cftchapterprecis are supported.

Discard all options for lwarf-tocloft:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*{\cftmarktoc}{} 
4 \newcommand*{\cfttoctitlefont}{} 
5 \newcommand*{\cftaftertoctitle}{} 

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*{\cftmarklof}{} 
9 \newcommand*{\cftloftitlefont}{} 
10 \newcommand*{\cftafterloftitle}{} 

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

13 \newcommand*{\cftmarklot}{} 
14 \newcommand*{\cftlottitlefont}{} 
15 \newcommand*{\cftafterlottitle}{} 

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}{1}
20 \newcommand*{\cftnodots}{5000}
21 
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*{\cftsetpnumwidth}[1]{}
24 \newcommand*{\cftsetrmarg}[1]{}
```

```
25 \newcommand*{\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartindent}
29 \newlength{\cftpartnumwidth}
30 \newcommand*{\cftpartfont}={}
31 \newcommand*{\cftpartpresnum}{}
32 \newcommand*{\cftpartaftersnum}{}
33 \newcommand*{\cftpartaftersnumb}{}
34 \newcommand*{\cftpartleader}{}
35 \newcommand*{\cftpartdotsep}{1}
36 \newcommand*{\cftpartpagefont}{}
37 \newcommand*{\cftpartafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}={}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchappagefont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}={}
53 \newcommand*{\cftsecpresnum}{}
54 \newcommand*{\cftsecaftersnum}{}
55 \newcommand*{\cftsecaftersnumb}{}
56 \newcommand*{\cftsecleader}{}
57 \newcommand*{\cftsecdotsep}{1}
58 \newcommand*{\cftsecpagefont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}={}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

```
69 \newcommand*{\cftsubsecpagefont}{}  
70 \newcommand*{\cftsubsecafterpnum}{}  
  
71 \newlength{\cftbeforesubsubsecskip}  
72 \newlength{\cftsubsubsecindent}  
73 \newlength{\cftsubsubsecnumwidth}  
74 \newcommand*{\cftsubsubsecfont}{}  
75 \newcommand*{\cftsubsubsecpresnum}{}  
76 \newcommand*{\cftsubsubsecaftersnum}{}  
77 \newcommand*{\cftsubsubsecaftersnumb}{}  
78 \newcommand*{\cftsubsubsecleader}{}  
79 \newcommand*{\cftsubsubsecdotsep}{1}  
80 \newcommand*{\cftsubsubsecpagefont}{}  
81 \newcommand*{\cftsubsubsecafterpnum}{}  
  
82 \newlength{\cftbeforeparaskip}  
83 \newlength{\cftparraindent}  
84 \newlength{\cftparranumwidth}  
85 \newcommand*{\cftparrafont}{}  
86 \newcommand*{\cftparrapresnum}{}  
87 \newcommand*{\cftparraaftersnum}{}  
88 \newcommand*{\cftparraaftersnumb}{}  
89 \newcommand*{\cftparraleader}{}  
90 \newcommand*{\cftparradotsep}{1}  
91 \newcommand*{\cftparrapagefont}{}  
92 \newcommand*{\cftparraafterpnum}{}  
  
93 \newlength{\cftbeforesubparaskip}  
94 \newlength{\cftsubparaindent}  
95 \newlength{\cftsubparanumwidth}  
96 \newcommand*{\cftsubparafont}{}  
97 \newcommand*{\cftsubparapresnum}{}  
98 \newcommand*{\cftsubparaaftersnum}{}  
99 \newcommand*{\cftsubparaaftersnumb}{}  
100 \newcommand*{\cftsubparaleader}{}  
101 \newcommand*{\cftsubparadotsep}{1}  
102 \newcommand*{\cftsubparapagefont}{}  
103 \newcommand*{\cftsubparaafterpnum}{}  
  
104 \newlength{\cftbeforefigskip}  
105 \newlength{\cftfigindent}  
106 \newlength{\cftfignumwidth}  
107 \newcommand*{\cftfigfont}{}  
108 \newcommand*{\cftfigpresnum}{}  
109 \newcommand*{\cftfigaftersnum}{}  
110 \newcommand*{\cftfigaftersnumb}{}  
111 \newcommand*{\cftfigleader}{}  
112 \newcommand*{\cftfigdotsep}{1}  
113 \newcommand*{\cftfigpagefont}{}  

```

```

114 \newcommand*{\cftfigafterpnum}{}

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*{\cftsubfigfont}={}
119 \newcommand*{\cftsubfigpresnum}{}
120 \newcommand*{\cftsubfigaftersnum}{}
121 \newcommand*{\cftsubfigaftersnumb}{}
122 \newcommand*{\cftsubfigleader}={}
123 \newcommand*{\cftsubfigdotsep}{1}
124 \newcommand*{\cftsubfigpagefont}={}
125 \newcommand*{\cftsubfigafterpnum}{}

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*{\cfttabfont}={}
130 \newcommand*{\cfttabpresnum}{}
131 \newcommand*{\cfttabaftersnum}{}
132 \newcommand*{\cfttabaftersnumb}{}
133 \newcommand*{\cfttableader}={}
134 \newcommand*{\cfttabdotsep}{1}
135 \newcommand*{\cfttabpagefont}={}
136 \newcommand*{\cfttabafterpnum}{}

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*{\cftsubtabfont}={}
141 \newcommand*{\cftsubtabpresnum}{}
142 \newcommand*{\cftsubtabaftersnum}{}
143 \newcommand*{\cftsubtabaftersnumb}{}
144 \newcommand*{\cftsubtableader}={}
145 \newcommand*{\cftsubtabdotsep}{1}
146 \newcommand*{\cftsubtabpagefont}={}
147 \newcommand*{\cftsubtabafterpnum}{}

148 \newcommand{\cftsetindents}[3] {}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberson}[1]{}

```

Emulated through the `\newfloat` mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}{}

```

```
154 {\newfloat{#2}{tbp}{#3}{#1}}
155 {\newfloat{#2}{tbp}{#3}}
156 \cnamedef{listof#2}{\listof{#2}{#4}}
157 \cnamedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \cnamedef{cft#2font}{}
162 \cnamedef{cft#2presnum}{}
163 \cnamedef{cft#2aftersnum}{}
164 \cnamedef{cft#2aftersnumb}{}
165 \cnamedef{cft#2leader}{}
166 \cnamedef{cft#2dotsep}{1}
167 \cnamedef{cft#2pagefont}{}
168 \cnamedef{cft#2afterpnum}{}
169 }
```

\cftchapterprecis from tocloft:

```
170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{%
176   \addtocontents{toc}{%
177     \begin{protect}\begin{quote}#1\end{protect}\end{quote}}
178   \begin{protect}\begin{quote}#1\end{protect}\end{quote}}
179 }
180 }
```

## Package 63

### **l warp-trivfloat.sty**

#### 131 Trivfloat

Pkg **trivfloat** trivfloat is forced to use the built-in l warp emulation for floats.

Discard all options for l warp-trivfloat. This tells trivfloat not to use floatrow or memoir.

**for HTML output:** 1 \LWR@ProvidesPackageDrop{trivfloat}  
2 \LWR@origRequirePackage{trivfloat}

**for HTML & PRINT:** 3 \begin{warpall}

To create a new float type and change its name:

---

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

---

4 \end{warpall}

\tfl@chapter@fix Nullified at the beginning of the document. Is used by trivfloat to correct float chapter numbers, but is not needed for l warp.

**for HTML output:** 5 \begin{warpHTML}

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

7 \end{warpHTML}

##### 131.1 Combining \newfloat, \trivfloat, and algorithmicx

**for HTML & PRINT:** 8 \begin{warpall}

For both print and HTML output:

 When using float, trivfloat, or algorithmicx at the same time, be aware of conflicting file usage. algorithmicx uses .loa. trivfloat by default starts with .loa and goes up

for additional floats, skipping .lof and .lot.

 When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid .loa used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using .lof and .lot.

 When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with .lob:

---

```
\makeatletter
\setcounter{tflfloatcnt}{1} % start trivfloats with .lob
\makeatletter
```

---

9 \end{warpall}

## Package 64

### **l warp-verse.sty**

#### 132 Verse

Pkg **verse** `verse` is supported and patched by `l warp`.

for **HTML output**: Pass all options for `l warp-verse`:

```
1 \LWR@ProvidesPackagePass{verse}
```

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

---

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

---

Len `\leftskip`  
 Len `\leftmargini`  
 Len `\TMLvleftskip`  
 Len `\TMLleftmargini`

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

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

Env **verse** The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
```

At the beginning of the `verse` environment:

```

3 \AtBeginEnvironment{verse}
4 {%
Pkg  verse   The verse or memoir packages can place stanza numbers to the left with their
Pkg  memoir  \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:
Len  \leftskip 5 \ifdef{\vleftskip}{%
6 \setlength{\vleftskip}{\HTMLvleftskip}
7 \setlength{\leftmargini}{\HTMLleftmargini}
8 }{}%
9 \LWR@atbeginverbatim{verse}
10 \unskip\vspace{-\baselineskip}
11 }

```

After the end of the **verse** environment, which places the **pre** tag at the regular left margin:

```

12 \AfterEndEnvironment{verse}{%
13 \unskip\vspace{-\baselineskip}
14 \LWR@afterendverbatim
15 }

```

Patch to place poemtitle inside an HTML span of class **poemtitle**:

```

16 \ifdef{\poemtitle}{%
17 \DeclareDocumentCommand{@vstypepoemtitle}{m}{%
18   \vspace{\beforepoemtitleskip}%
19   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
20   \vspace{\afterpoemtitleskip}%
21 }
22 }{}%
23
24 }

```

## Package 65

### **lwarf-wallpaper.sty**

#### 133 Wallpaper

Pkg **wallpaper** wallpaper is emulated during HTML output, and the wallpaper package is ignored.

**for HTML output:**

```
1 \LWR@ProvidesPackageDrop{wallpaper}

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

## Package 66

### lwarf-wrapfig.sty

#### 134 Wrapfig

Pkg **wrapfig** wrapfig is emulated during HTML output, and the wrapfig package is ignored.

**for HTML output:**

```

1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the HTML style.

2 \newlength{\LWR@wrapwidth}

3
4 \newcommand*{\LWR@wrapposition}{}%
5
6 \newcommand*{\LWR@subwrapfigure}[2]{%
7 \LWR@maybeinthisfloat%
8 \renewcommand*{\LWR@wrapposition}{}%
9 \ifthenelse{%
10 \equal{#1}{r}\OR\equal{#1}{R}\OR%
11 \equal{#1}{o}\OR\equal{#1}{O}%
12 }{%
13 {\renewcommand*{\LWR@wrapposition}{float:right}}%
14 {\renewcommand*{\LWR@wrapposition}{float:left}}%
15 \setlength{\LWR@wrapwidth}{#2}%
16 \addtolength{\LWR@wrapwidth}{4em}%
17 \uselengthunit{PT}%
18 % \BlockClass{marginblock}[%
19 % width:\rndprintlength{\LWR@wrapwidth} ; %
20 % \LWR@wrapposition%
21 % ]%
22 \LWR@stoppars%
23 \LWR@htmlltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"%
24 style="width:\rndprintlength{\LWR@wrapwidth} ; %
25 \LWR@wrapposition"%
26 }
27 \LWR@startpars
28 }
29
30
31 \NewDocumentEnvironment{wrapfigure}{o m o m}
32 {%
33 \LWR@subwrapfigure{#2}{#4}%
34 \captionsetup{type=figure}%

```

```
35 }
36 {
37 \LWR@htmldivclassend{div}
38 }
39
40
41 \NewDocumentEnvironment{wraptable}{o m o m}
42 {%
43 \LWR@subwrapfigure{#2}{#4}%
44 \captionsetup{type=table}%
45 }
46 {
47 \LWR@htmldivclassend{div}
48 }
49
50
51 \NewDocumentEnvironment{wrapfloat}{m o m o m}
52 {%
53 \LWR@subwrapfigure{#3}{#5}%
54 \captionsetup{type=#1}%
55 }
56 {
57 \LWR@htmldivclassend{div}
58 }
59
60 \newlength{\wrapoverhang}
```

## Package 67

### **lwarf-xcolor.sty**

#### 135 Xcolor

Pkg **xcolor** `xcolor` is supported by `lwarf`.

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

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

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

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

**for HTML output:** 1 `\LWR@ProvidesPackagePass{xcolor}`

2 `\newcommand*{\LWR@tempcolor}{}%`

defaulting to black.

3 `\newcommand*{\LWR@currenttextcolor}{black}`

Creates a styled span with a color converted to HTML hex colorspace. Uses `\LWR@spandepth` to prevent paragraph tags inside the span. If used for `\textcolor`, with a styletext of `color:`, then the new color is copied into `\LWR@currenttextcolor` for possible re-use in `\rule`.

4 `\NewDocumentCommand{\LWR@colorstyle}{m m m +m}{%`

Use the `xcolor` package to convert to an HTML color space:

5 `\convertcolorspec{#2}{#3}{HTML}\LWR@tempcolor%`

If is a `\textcolor`, save a copy of this color for use by `\rule`:

6 `\ifthenelse{\equal{#1}{color:}}%`  
 7 `\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}{}%`

Create the HTML `span` with the styled color:

8 `\LWR@htmntagc{span style="#1\#\LWR@tempcolor"{}%}`  
 9 `\begin{LWR@nestspan}%`

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```
10 #4%
11 \LWR@htmntagc{/span}%
12 \end{LWR@nestspan}%
```

For paragraph-tag handling:

```
13 \LWR@ensuredoingapar%
14 }
```

\color appears in the L<sup>A</sup>T<sub>E</sub>X PDF output, but is ignored by pdftotext and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is \let to the final name once the HTML conversion starts.

\textcolor [*model*] {*color*} {*text*} is converted into an HTML hex color span.

```
15 \NewDocumentCommand{\LWR@textcolor}{O{named} m +m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }
```

\pagecolor [*model*] {*color*} is ignored. Use \NewCSS instead.

```
20 \newcommand*{\LWR@pagecolor}[2][named]{}
```

\colorbox [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```
21 \NewDocumentCommand{\LWR@colorbox}{O{named} m m}{%
22 \begingroup%
23 \LWR@colorstyle{background:}{#1}{#2}{#3}%
24 \endgroup%
25 }
```

\fcolorbox [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*} is converted into a framed HTML hex background color span.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{O{named} m O{named} m m}{%
27 \begingroup%
28 \LWR@colorstyle{border:1px solid }{#1}{#2}%
29 }
```

```
29 {\LWR@colorstyle{background:#3}{#4}{#5}}%  
30 \endgroup%  
31 }
```

## Package 68

### **lwarf-xfrac.sty**

#### 136 Xfrac

Pkg **xfrac** Supported by adding `xfrac` instances.

for **HTML output:** 1 `\LWR@ProvidesPackagePass{xfrac}`

**⚠ font size** In the user's document preamble, `lwarf` should be loaded after font-related setup. During HTML conversion, this font is used by `lwarf` to generate its initial PDF output containing HTML tags, later to be converted by `pdftotext` to a plain text file. While the text may be in any font which `pdftotext` can read, the math is directly converted into SVG images using this same user-selected font. `xfrac` below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` [*<instance>*] [*<num>*] [*<sep>*] [*<denom>*]

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for **HTML & PRINT:** 2 `\begin{warpall}`

User-redefinable macro which controls the font size of the fraction.

3 `\newcommand*\{\xfracHTMLfontsize\}{.6em}`

4 `\end{warpall}`

for **HTML output:** 5 `\begin{warpHTML}`

**font size** A span for a small font, used in the numerator and denominator:

```
6 \newcommand*\{\LWR@htmlsmallfontstart\}{%
7 \LWR@htmntagc{span style="font-size:\xfracHTMLfontsize\{}%
8 \LWR@nestspan\%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmlltagc{/span}%
14 \endLWR@nestspan%
15 }

```

**\scalebox** A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```
16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}
```

**instances** Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```

17 \DeclareInstance{xfrac}{default}{text}{%
18 numerator-format = {%
19 \let\scalebox\LWR@noscalebox%
20 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
21 denominator-format = {%
22 \let\scalebox\LWR@noscalebox%
23 \LWR@htmlsmallfontstart{}\,,#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

24 scaling = false
25 }
26 \DeclareInstance{xfrac}{lmr}{text}{%
27 numerator-format = {%
28 \let\scalebox\LWR@noscalebox%
29 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
30 denominator-format = {%
31 \let\scalebox\LWR@noscalebox%
32 \LWR@htmlsmallfontstart{}\,,#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{%
36 numerator-format = {%
37 \let\scalebox\LWR@noscalebox%
38 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend},%
39 denominator-format = {%
40 \let\scalebox\LWR@noscalebox%
41 \LWR@htmlsmallfontstart{}\,,#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{
45 numerator-format = {%
46 \let\scalebox\LWR@noscalebox%
47 \LWR@htmlsmallfontstart{textsuperscript{#1}\,\LWR@htmlsmallfontend},
48 denominator-format = {%
49 \let\scalebox\LWR@noscalebox%
50 \LWR@htmlsmallfontstart{}{#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
51 scaling = false
52 }
53 \end{warpHTML}
```

# Change History and Index

## Change History

v0.10	General: 2016/03/08 Initial version	1	Docs: Commands for a successful HTML conversion. . . . .	54
v0.11	General: 2016/03/11	1	Docs: Commands into a warpprint environment. . . . .	53
	Added section: Operating-System portability.	87	Docs: Newclude limitations. . . .	64
	Added section: Selecting the operating system. . . . .	52	Docs: Table: Cross-referencing data structures. . . . .	188
	Test Suite: limages and index in README.txt . . . . .	1	Docs: Table: Float data structures. . . . .	198
	Test Suite: MS-Windows in README.txt . . . . .	1	Docs: Trademarks section. . . .	267
v0.12	\LWR@newhtmlfile: Bugfix: TOC with numbered files. . . . .	118	Docs: Troubleshooting cross-references. . . . .	67
	General: 2016/03/14	1	floatrow: Added package . . . . .	327
	Global: Uses \p@(type) in float captions. . . . .	1	Pagenote: Assigned cleveref name. . . . .	243
	Test Suite: Sub-figures . . . . .	1	Test Suite: Assigned cleveref name for Test Float. . . . .	1
v0.13	\CaptionSeparator: Fix for newer babel package. . . . .	200	Test Suite: Floatrow . . . . .	1
	\LWR@LwarpStart: \up and \fup .	130	v0.15	
	\published: Default to empty published. . . . .	136	\printpublished: No div if empty. . . . .	140
	\subtitle: Default to empty subtitle. . . . .	137	\printsubtitle: No div if empty. .	140
	General: 2016/03/24	1	General: 2016/04/06	1
	Removed package: subfig . . . .	1	Added . . . . .	330
	Test Suite: Ordinals, Subcaption	1	Ampersand (&): Fixed handling when passed as an argument. .	164
	tikzpicture: Fix dollar-redefined bug for newer package. . . . .	245	Docs: Added warning icons for items needing special attention. .	71
v0.14	\LWR@htmlsectionfilename: Fix: Links to home page. . . . .	94	Docs: Clarify print/HTML output. . . . .	52
	\noteentry: Fix for label inside a footnote. . . . .	113	Docs: Moved the supported functions table to the introduction. . . . .	21
	General: 2016/03/31	1	Files: lwarp_formal.css added. .	1
			Fix: steps counter . . . . .	330
			Fixed & handling. . . . .	328
			Test Suite: test_suite_formal.css file added. . . . .	1

v0.16	
General: 2016/04/11 . . . . .	1
\titlingpage: Improved print-output spacing. . . . .	141
Added XeLaTeX, LuaLaTeX support. . . . .	77
Docs: Font and UTF-8 support. . . . .	51
Docs: Moved location of \usepackage{l warp}. . . . .	52
Docs: Text not converting. . . . .	67
Fix: amsmath options clash . . . . .	81
Fix: newtxmath compatibility. . . . .	81
Lwarp no longer selects fonts. . . . .	51, 77
Removed package: suffix . . . . .	1
Test Suite: Improved titlingpage. . . . .	141
Test Suite: Lwarp no longer selects fonts. . . . .	1
Test Suite: Supports XeLaTeX, LuaLaTeX. . . . .	1
xfrac: Adjusted for the use of any font: . . . . .	393
v0.17	
\lWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. . . . .	94
General: 2016/04/14 . . . . .	1
mdframed: Added. . . . .	349
Test Suite: Fix: Print-version front-matter page numbers. . . . .	1
Test Suite: Mdframed . . . . .	1
v0.18	
\lWR@hspace: \hspace supported . . . . .	258
\lWR@includegraphicsb: Add: svgz file extension. . . . .	237
em, ex, %, px dimensions preserved. . . . .	237
Fix: \linewidth, \textwidth, \textheight inside a minipage. . . . .	237
Improved HTML output linebreaks. . . . .	237
\lWR@myshorttoc: Reorganize \HomeHTMLfilename logic. . . . .	204
\lWR@newhtmlfile: Sidetoc after title, improving responsive design. . . . .	118
\lWR@requesttoc: Reorganize \HomeHTMLfilename logic. . . . .	132
v0.19	
\HTMLFileName: Docs: Escape filename underscores. . . . .	93
\HomeHTMLFileName: Docs: Escape filename underscores. . . . .	93
\lWR@LwarpStart: Enabled \\ equal to \newline. . . . .	129
\lWR@doubledollar: MathJax support. . . . .	214
\lWR@filestart: l warp_mathjax.txt loaded. . . . .	127
\lWR@hspace: Fix: \hspace length computations . . . . .	258
\lWR@minipagestartpars: Suppresses paragraph tags between minipages. . . . .	257
\lWR@singledollar: MathJax support. . . . .	214
\lWR@tabledatamultirowtag: Added optional args. . . . .	183
\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. . . . .	227

\minipagefullwidth:	Added: No width tag for the next minipage in HTML.	249
\rule:	Added	262
\warpHTMLonly:	Added	73
\warpprintonly:	Replaces \rowprintedonly.	73
General:	2016/06/08	1
Added.		392
Avoi		213
cleveref:	Loaded	
\AtEndPreamble.		243
CSS for table note item.		371
Docs:	multirow browser bug workaround.	61, 161
Docs:	Math options.	52
Docs:	Table: Cross-referencing data structures, updated.	188
File:	l warp_mathjax.txt added.	1
File:	l warp.css: tnoteitemheader added	1
Introduction:	MathJax support mentioned.	20
MathJax support	added.	215, 220, 221
Options:	mathsvg and mathjax	74
Supports colored \rule.		389
titlesp:	null \pagestyle and \thispagestyle for HTML	81
tikzpicture:	Adapts to tikz version.	245
equation:	MathJax support.	218
v0.20		
\BlockClassSingle:	Renamed from "LWR@htmldivclassline".	102
\HTMLFileName:	No longer escape underscores.	93
\HomeHTMLFileName:	No longer escape underscores.	93
\InlineClass:	Renamed from "inlineclass".	103
\LWR@LwarpStart:	Fix: math cross references.	131
\LWR@closeparagraph:	\unskip extra spaces.	106
No break tags in the start/end of a tabular.		106
\LWR@endofline:	Fix: \\	257
\LWR@filestart:	Adds meta description.	127
\LWR@hspace:	Add: Supports HTML thin breakable space	258
\LWR@htmldivclass:	Added optional style.	101
\LWR@htmlelementclass:	Added optional style.	100
\LWR@htmlfilename:	HTMLFileName: removed additional trailing '^', and may be empty.	94
Sections called "Index" or "index"	have an underscore prepended to their filenames if no prefix.	94
\LWR@includegraphicsb:	Fix: \linewidth in a floatrow.	237
Fix: Expands filename.		237
\LWR@longtabledatacaptiontag:	Fix: Pars in captions.	181
\LWR@section:	Combined higher-level sections together into files.	121
\LWR@subhtmlelementclass:	Factored code.	100
\SetHTMLFileName:	Add: Control file numbers.	93
\cpagerefFor:	User-redefinable word for page references.	244
\dotfill:	Inserts an ellipsis.	255
\hfill:	Inserts a \quad.	255
\hrulefill:	Inserts a short rule.	255
\hyperindexref:	Print mode provided in case hyperref not used.	211
\pageref:	Added.	194
\tracingl warp:	Added.	85
\warpOSwindows:	Auto-detects operating system.	87
General:	2017/02/09	1
afterpage:	Added.	315
alltt:	Added.	317
bookmark:	Added.	317
caption and subcaption supported.		1
cleveref and referencing patches:		
Applied \AfterEndPreamble.		243
draftwatermark:	Added.	319
eso-pic:	Added.	322
everypage:	Added.	323
extramarks:	Added.	323
fancyhdr:	Added.	324

<b>hyperref:</b> Additional user macros	335	Tabular: \unskip extra spaces.	161
<b>keyfloat:</b> Added	339	Test Suite: HTML meta descriptions	1
<b>letterspace:</b> User-interface emulated	340	<b>Verbatim:</b> Added	152
<b>listings:</b> Added	341	<b>verbatim:</b> Added	150
<b>ltcaption:</b> Added	347	<b>BlockClass:</b> Added optional style.	102
<b>l warp-newproject:</b> Added	268	Renamed from "blockclass". . .	102
<b>microtype:</b> User-interface emulated	356	<b>LWR@nestspan:</b> Fix: Minipages inside a span	98
<b>needspace:</b> Added	360	v0.21	
<b>nowidow:</b> Added	362	\LWR@LwarpStart: Changed <i>latexitimages</i> to a .txt file.	129
<b>pagenote:</b> Added	362	\LWR@filestart: Skip title if not given	127
<b>placeins:</b> Added	363	\LWR@newhtmlfile: Skip title if not given	118
<b>ragged2e:</b> Added	364	\marginpar: Fixed source listing.	114
<b>setspace:</b> Improved support	366	General: 2017/02/23	1
<b>textpos:</b> Added	370	<b>fontenc:</b> Added	332
<b>titlesp:</b> Added	372	<b>fontspec:</b> Added	332
<b>titlesec:</b> Added	374	<b>inputenc:</b> Added	338
<b>titletoc:</b> Added	375	<b>newclude:</b> Added	360
<b>titling:</b> Improved compatibility	376	<b>newunicodechar:</b> Added	361
<b>tocloft:</b> Added	377	Add: Errors for misplaced packages	75
<b>wallpaper:</b> Added	386	Docs: Added <i>internet</i> class.	25
<b>wrapfig:</b> Added	387	Docs: Added TeX2page, GladTeX.	25
Added @, <, > columns	161	Docs: Installing on Windows.	31
Added \NewHTMLdescription	111	File <i>l warp_tutorial.txt</i> added.	35
Added single-expansion data arrays	92	<b>l warpmk:</b> Fix: <i>l warpmk</i> again for Windows	304
Code factored into independent <i>l warp_html</i> files	312	<b>l warpmk:</b> Fix: <i>l warpmk l images</i> for Windows	304
Docs: Examples for generating HTML file names	45	<b>l warpmk:</b> Fix: <i>l warpmk</i> uses <i>latexitimages</i> text file instead of shell script	304
Docs: Improved index	1	v0.22	
Enhanced titling support	138	\LWR@parseDcolumn: Added tabular D column	170
File: <i>l warp.css</i> : Minor fixes for validation	1	\LWR@parsebangcolumn: Added tabular ! column	167
File: <i>l warpmk</i> used to compile print, HTML, indexes, and <i>latexitimages</i>	1	\LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns	172
Fix: \linewidth in a floatrow	328	\LWR@printmccoldata: Added tabular D, !, and X columns	178
Improved float caption type handling	325	General: 2017/03/02	1
Moved sidebar and example code to test suite	1	<b>abstract:</b> Added	313
Page geometry set to 6in wide with large margins	79	<b>changepage:</b> Added	318
Parallel versions of aux files for print/HTML	1		
Removed reliance on make, grep, gawk	1		

dcolumn:	Added.	319	showkeys:	Added.	367
enumerate:	Added.	320	sidecap:	Added.	368
ftnright:	Added.	333	tabularx:	Added.	369
geometry:	Nullified commands.	333	variorref:	Supported.	63
indentfirst:	Added.	338	verse:	Added.	384
layout:	Added.	340	v0.23		
lscape:	Added.	347	\LWR@parsetablecols:	Fix for vert	
mcaption:	Added.	348	bar column type.	.....	172
nameref:	Added.	359	\LWR@printmccoldata:	Fix for vert	
nextpage:	Added.	361	bar column type.	.....	178
parskip:	Added.	363	General:	2017/03/02	1

## Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	A
\\$ .....	<i>213</i>
\& .....	<i>92</i> , <u>2099</u>
\( .....	<u>3076</u>
\) .....	<u>3076</u>
\@setpageref .....	<u>3779</u>
\@setcref .....	<u>3775</u>
\@setcrefrange .....	<u>3776</u>
@author .....	<i>135</i>
\begin{theorem} .....	<u>1955</u>
\caption .....	<u>2820</u>
\currentlabel .....	<i>113</i>
\currentlabelname .....	<u>2609</u>
@date .....	<i>135</i>
\dbfloat .....	<u>2803</u>
\endtheorem .....	<u>1965</u>
\float .....	<u>2803</u>
\fnsymbol .....	<u>1688</u>
\makecaption .....	<u>2822</u>
\maketitle .....	<u>1699</u>
\namedef .....	<u>69</u>
\opargbegintheorem .....	<u>1960</u>
@published .....	<i>135</i>
@subtitle .....	<i>135</i>
@title .....	<i>135</i>
\\" .....	<u>257</u>
\$ .....	<u>214</u>
\$\$ .....	<u>214</u>
	<b>B</b>
	babel (package) .....
	<i>65</i>

\backmatter . . . . .	<u>1085</u>	\centering . . . . .	<u>130</u>
BaseJobName (option) [l warp- newproject] . . . . .	<u>268</u>	changepage (package) . . . . .	<u>318</u>
\BaseJobname . . . . .	<u>484</u>	\chapter . . . . .	<u>1203</u>
BaseJobname (option) [l warp] . . . . .	<u>74</u>	\citetitle . . . . .	<u>1745</u>
baseline		classses:	
tabular . . . . .	<u>171</u>	internet . . . . .	<u>25</u>
\bfseries . . . . .	<u>3976</u>	cleveref (package) . . . . .	<u>243</u>
\BibTeX . . . . .	<u>4156</u>	cmap (package) . . . . .	<u>52</u>
BlockClass (environment) . . . . .	<u>699</u>	CombineHigherDepths (boolean) . . . . .	<u>45, 115</u>
\BlockClassSingle . . . . .	<u>711</u>	comment (package) . . . . .	<u>73</u>
bookmark (package) . . . . .	<u>317</u>	Computer Modern . . . . .	<u>51</u>
booktabs (package) . . . . .	<u>318</u>	\ConTeXt . . . . .	<u>4153</u>
booleans:		counters:	
CombineHigherDepths . . . . .	<u>45, 115</u>	FileDepth . . . . .	<u>44, 115</u>
FileSectionNames . . . . .	<u>45, 94</u>	LWR@externalfilecnt . . . . .	<u>213</u>
LWR@amsmultiline . . . . .	<u>220</u>	LWR@htmldocumentnumber . . . . .	<u>94</u>
LWR@doingapar . . . . .	<u>104</u>	LWR@latestautopage . . . . .	<u>201</u>
LWR@doinghline . . . . .	<u>162</u>	LWR@teximagedepth . . . . .	<u>226</u>
LWR@doingstartpars . . . . .	<u>104</u>	LWR@teximagenumber . . . . .	<u>226</u>
LWR@doingtbrule . . . . .	<u>162</u>	LWR@midrulecounter . . . . .	<u>176</u>
LWR@freezethisfloat . . . . .	<u>200</u>	LWR@minipagedepth . . . . .	<u>246</u>
LWR@infloatarrow . . . . .	<u>236</u>	LWR@nextautofloat . . . . .	<u>201</u>
LWR@intabularmetadata . . . . .	<u>163</u>	LWR@nextautopage . . . . .	<u>201</u>
LWR@minipagewidth . . . . .	<u>249</u>	LWR@nextequation . . . . .	<u>215</u>
LWR@minipagethispar . . . . .	<u>249</u>	LWR@nextpage . . . . .	<u>226</u>
LWR@skippingmrowcell . . . . .	<u>163</u>	LWR@prevFileDepth . . . . .	<u>121</u>
LWR@starredlongtable . . . . .	<u>180</u>	LWR@spandepth . . . . .	<u>104</u>
LWR@startedrow . . . . .	<u>162</u>	LWR@tablecolspos . . . . .	<u>166</u>
LWR@tableparcell . . . . .	<u>162</u>	LWR@tablecolswidth . . . . .	<u>166</u>
LWR@tikzbabel . . . . .	<u>245</u>	LWR@tabletotalcols . . . . .	<u>166</u>
LWR@tracingl warp . . . . .	<u>85</u>	LWR@tabletotalcolsnext . . . . .	<u>166</u>
LWR@validtablecol . . . . .	<u>170</u>	LWR@thisfloat . . . . .	<u>200</u>
LWR@verbtags . . . . .	<u>150</u>	SideTOCDepth . . . . .	<u>44, 207</u>
mathjax . . . . .	<u>73</u>	tocdepth . . . . .	<u>44</u>
usingOSwindows . . . . .	<u>87</u>	\cpagerefFor . . . . .	<u>3778</u>
warpingHTML . . . . .	<u>73</u>	\cref@currentlabel . . . . .	<u>113</u>
warpingprint . . . . .	<u>73</u>	cross-references	
bugs . . . . .	<u>67</u>	missing or incorrect . . . . .	<u>67</u>
BVerbatim (environment) . . . . .	<u>1941</u>	CSS	
		file selection . . . . .	<u>57</u>
		l warp.css . . . . .	<u>57</u>
		per HTML page . . . . .	<u>57</u>
		project-specific changes . . . . .	<u>57</u>
		D	
		danger icon . . . . .	<u>71</u>
		\date . . . . .	<u>56</u>
		dcolumn (package) . . . . .	<u>319</u>
		debugging . . . . .	<u>67</u>
		\DeclareGraphicsExtensions . . . . .	<u>3527</u>

\DeclareGraphicsRule	3527	titlepage	55, 1438
Deja Vu	51	titlingpage	55, 1441, 1454
description		Verbatim	1809
HTML meta tag	56, 112	verbatim	1768
description (environment)	2028	VerbatimClass	1801
displaymath (environment)	3085	verse	2
\dotfill	3992	warpall	34, 53
draftwatermark (package)	319	warpHTML	42, 47, 53
		warpprint	35, 47, 53
		epigraph (package)	321
		equation (environment)	3152
\emph	3965	error messages	67
\end@dblfloat	2812	eso-pic (package)	322
\end@float	2812	etoolbox (package)	72
\enlargethispage	4091	everyhook (package)	79
\enskip	257, 4057	everypage (package)	323
enumerate (environment)	1997	expl3 (package)	79
enumerate (package)	65, 320	extramarks (package)	323
enumitem (package)	65, 80		
environ (package)	81		
environments:			
abstract	1729	fancyhdr (package)	324
align	3288	fancyvrb (package)	79
align*	3310	FAQ	67
BlockClass	699	fbox	
BVerbatim	1941	around a minipage	54
center	3506	\fbox	3955
description	2028	FileDepth (counter)	44, 115
displaymath	3085	files	
enumerate	1997	inaccessible	44, 45
equation	3152	files:	
flalign	3332	glyptounicode	51
flalign*	3354	lwarf.css	56, 271
flushleft	3512	lwarf_formal.css	296
flushright	3509	lwarf_html.xdy	301
gather	3244	lwarf_mathjax.txt	301
gather*	3266	lwarf_sagebrush.css	291
itemize	1979	lwarf Tutorial.txt	35
lateximage	3428	lwarfpmk	304
longtable	2	lwarfpmk.conf	266
LVerbatim	1952	project.css	56
LWR@nestspan	594	project.lwarfpmkconf	270
LWR@tabular	2567	project_html.tex	270
math	3084	sample_project.css	56, 301
minipage	3885	tutorial.tex	35
multiline	3197	FileSectionNames (boolean)	45, 94
multiline*	3220	\flagverse	385
picture	244, 3796	flalign (environment)	3332
quote	1753	flalign* (environment)	3354
theindex	3011	Flare (program)	27
tikzpicture	245, 3815	float (package)	325

\floatname	11	horizontal space
\floatplacement	14	between minipages
floatrow (package)	327	\href
\floatstyle	17	\hrulefill
flushleft (environment)	3512	\hspace
flushright (environment)	3509	HTML
font		commands for a successful conversion
bitmapped	51	54
Computer Modern	51	conversion settings
Deja Vu	51	44
ligatures	51, 52	filename generation
packages	51	45
selection	51	headings
size — lateximage	58, 211, 226	70
size — math, SVG	58, 211, 226	meta tag
size — xfrac	392	description
fontenc (package)	51, 332	generator
fontspec (package)	51, 332	viewport
\footnote	916	tabular column conversion
footnotes	112	474
for		\HTMLentity
HTML & PRINT	71	485
HTML output	71	\HTMLFileName
PRINT output	71	HTMLleftmargini (length)
\framebox	3952	64, 149, 150, 384
FrameMaker (program)	27	\HTMLunicode
Frequently Asked Questions	67	475
\frontmatter	1082	HTMLvleftskip (length)
ftnright (package)	333	64, 149, 384
\fup	3989	\hyperindexref
G		3028
gather (environment)	3244	hyperref (package)
gather* (environment)	3266	195, 335
GELLMU (program)	25	\hypertoc
generator		2970
HTML meta tag	128	\hypertocfloat
geometry (package)	78, 333	2979
gettitlestring (package)	79	I
GladTeX (program)	26	icon
glyptounicode (file)	51	warning
graphics (package)	232, 334	71
\graphicspath	3525	ifplatform (package)
graphicx (package)	232, 334	73
H		images
Hevea (program)	25	graphicx package
\hfill	3990	232
\HomeHTMLFileName	486	in strange places
\HomeHTMLFilename	44	68
J		\includegraphics
JavaScript		3707
MathJax		338
K		InDesign (program)
keyfloat (package)		27
kvoptions (package)		\InlineClass
		719
		inputenc (package)
		51, 338
		internet (class)
		25
		item, list, empty
		54
		itemize (environment)
		1979
		\itshape
		3981
		J
		JavaScript
		MathJax
		57, 211
		K
		keyfloat (package)
		339
		kvoptions (package)
		73

<b>L</b>	[l warp]:
label	
in HTML . . . . .	131
math environment . . . . .	220
language HTML metadata . . . . .	127
\LaTeX . . . . .	4139
\TeX2HTML (program) . . . . .	25
\TeXe . . . . .	4139
\teximage (environment) . . . . .	3428
\LateximageFontSizeName . . . . .	3392
lateximages	
font size . . . . .	58, 211, 226
\TeXML (program) . . . . .	25
layout (package) . . . . .	340
lengths:	
HTMLleftmargini . . . . .	64, 149, 150, 384
HTMLvleftskip . . . . .	64, 149, 384
LWR@minipageheight . . . . .	247
LWR@minipagewidth . . . . .	247
LWR@tempheight . . . . .	258
LWR@tempraise . . . . .	258
LWR@tempwidth . . . . .	258
\VerbatimHTMLWidth . . . . .	150
vleftmargini . . . . .	64, 149, 384
vleftskip . . . . .	64, 149, 384, 385
letterspace (package) . . . . .	340
\LibreOffice (program) . . . . .	26
ligatures . . . . .	51, 52, 78
line numbers . . . . .	71
\linebreak . . . . .	4085
\LinkHome . . . . .	534
list item, empty . . . . .	54
\listings (package) . . . . .	341
\listof . . . . .	2937
\modern (package) . . . . .	51
\longtable (environment) . . . . .	2
\longtable (package) . . . . .	345
\scape (package) . . . . .	347
\tcaption (package) . . . . .	347
\LuaTeX	
detection . . . . .	77
file & section names . . . . .	120
\LuaTeX . . . . .	4147
\LuaTeX (program) [requirement] . . . . .	29
\LuaTeX . . . . .	4147
\Verbatim (environment) . . . . .	1952
l warp	
loading . . . . .	52
options . . . . .	52
BaseJobname (option) . . . . .	74
mathjax (option) . . . . .	52, 74
mathsvg (option) . . . . .	52, 74
warpHTML (option) . . . . .	52, 74
warpprint (option) . . . . .	52, 73
l warp (package) . . . . .	52
[l warp-newproject]:	
BaseJobName (option) . . . . .	268
l warpmk (option) . . . . .	268
warpHTML (option) . . . . .	268
warpprint (option) . . . . .	268
l warp-newproject (package) . . . . .	268
l warp.css (file) . . . . .	56, 271
l warp_formal.css (file) . . . . .	296
l warp_html.xdy (file) . . . . .	301
l warp_mathjax.txt (file) . . . . .	301
l warp_sagebrush.css (file) . . . . .	291
l warp Tutorial.txt (file) . . . . .	35
l warpmk (file) . . . . .	304
l warpmk (option) [l warp-newproject]	268
l warpmk.conf (file) . . . . .	266
\LWR@addmathjax . . . . .	3141
\LWR@afterendverbatim . . . . .	1793
\LWR@amsmultiline (boolean) . . . . .	220
\LWR@atbeginverbatim . . . . .	1785
\LWR@beginhideamsmath . . . . .	3182
\LWR@botnavigation . . . . .	542
\LWR@caption@begin . . . . .	2834
\LWR@caption@end . . . . .	2843
\LWR@clearmidrules . . . . .	2324
\LWR@closeparagraph . . . . .	786
\LWR@closeprevious . . . . .	562
\LWR@closetabledatacell . . . . .	2077
\LWR@colaferspec . . . . .	166
\LWR@colatspec . . . . .	166
\LWR@colbangspec . . . . .	166
\LWR@colbeforespec . . . . .	166
\LWR@commentfooterstyle . . . . .	226
\LWR@copyfile . . . . .	301
\LWR@createautosec . . . . .	1091
\LWR@currentcss . . . . .	887
\LWR@currentHTMLdescription . . . . .	902
\LWR@descitem . . . . .	2016
\LWR@docmidrule . . . . .	2343
\LWR@doingapar (boolean) . . . . .	104
\LWR@doinghline (boolean) . . . . .	162
\LWR@doingstartpars (boolean) . . . . .	104
\LWR@doingtbrule (boolean) . . . . .	162
\LWR@domulticolumn . . . . .	2416

\LWR@doubleollar .....	3054	\LWR@lateximagedepthref .....	2640
\LWR@endhideamsmath .....	3190	\LWR@lateximagenumber (counter) ..	<i>226</i>
\LWR@endminipage .....	3847	\LWR@lateximagenumberref .....	2643
\LWR@endofline .....	4035	\LWR@loadafter .....	70
\LWR@endsubminipage .....	3870	\LWR@loadbefore .....	81
\LWR@ensuredoingapar .....	760	\LWR@longtabledatacaptiontag ..	2437
\LWR@externalfilecnt (counter) ..	<i>213</i>	\LWR@lookforpackagename .....	240
\LWR@filenamenoblocks .....	953	\LWR@LwarpEnd .....	<i>1378</i> , 4183
\LWR@filestart .....	1255	\LWR@LwarpStart .....	<i>1309</i> , 4183
\LWR@findword .....	237	\LWR@makefnmark .....	3846
\LWR@floatbegin .....	2792	\LWR@makefntext .....	3839
\LWR@floatend .....	2805	\LWR@maketitlesetup .....	1679
\LWR@freezethisfloat (boolean) ..	<i>200</i>	\LWR@maybeincthisfloat .....	2817
\LWR@futureonospacelet .....	2056	\LWR@maybenewtablerow .....	2273
\LWR@getexparray .....	470	\LWR@midrulecounter (counter) ..	<i>176</i>
\LWR@getmynexttoken .....	2063	\LWR@midrules .....	<i>176</i>
\LWR@hidelatexequation .....	3127	\LWR@minipagedepth (counter) .....	<i>246</i>
\LWR@hspace .....	4065	\LWR@minipagefullwidth (boolean) ..	<i>249</i>
\LWR@htmlblockcomment .....	638	\LWR@minipageheight (length) .....	<i>247</i>
\LWR@htmlblocktag .....	646	\LWR@minipagestartpars .....	4039
\LWR@htmlblocktagcomment .....	651	\LWR@minipagestoppars .....	4045
\LWR@htmclosecomment .....	625	\LWR@minipagethispar (boolean) ..	<i>249</i>
\LWR@htmlcomment .....	634	\LWR@minipagewidth (length) .....	<i>247</i>
\LWR@htmdivclass .....	676	\LWR@multicolother .....	2368
\LWR@htmdivclassend .....	679	\LWR@multicolpartext .....	2363
\LWR@htmlelement .....	689	\LWR@multicolskip .....	2374
\LWR@htmlelementclass .....	665	\LWR@mynexttoken .....	2055
\LWR@htmlelementclassend .....	670	\LWR@myshorttoc .....	2895
\LWR@htmlelementclassline .....	682	\LWR@nameref .....	2632
\LWR@htmlelementend .....	692	\LWR@nestspan (environment) .....	594
\LWR@htmlfilename (counter) ..	<i>94</i>	\LWR@newhtmlfile .....	1031
\LWR@htmlfilerref .....	2635	\LWR@newlabel .....	2681
\LWR@htmlmathlabel .....	3172	\LWR@nextautofloat (counter) .....	<i>201</i>
\LWR@htmlopencomment .....	625	\LWR@nextautopage (counter) .....	<i>201</i>
\LWR@htmrefsectionfilename .....	527	\LWR@nextequation (counter) .....	<i>215</i>
\LWR@HTMLSanitize .....	3395	\LWR@nextpage (counter) .....	<i>226</i>
\LWR@HTMLSanitizeexpand .....	3413	\LWR@nullrule .....	3838
\LWR@htmsectionfilename .....	496	\LWR@openparagraph .....	765
\LWR@htmspan .....	600	\LWR@origcolspec .....	2130
\LWR@htmspanclass .....	608	\LWR@originname .....	3719
\LWR@htmntag .....	616	\LWR@originnames .....	3726
\LWR@htmtagc .....	584	\LWR@parseaftercolumn .....	2186
\LWR@includegraphicsb .....	3618	\LWR@parseatcolumn .....	2135
\LWR@indexitem .....	3016, 3020, 3024	\LWR@parsebangcolumn .....	2156
\LWR@indexsection .....	3000	\LWR@parsebeforecolumn .....	2177
\LWR@infloatrow (boolean) .....	<i>236</i>	\LWR@parseDcolumn .....	<i>2213</i>
\LWR@intabularmetadata (boolean) ..	<i>163</i>	\LWR@parsenormalcolumn .....	2199
\LWR@itemizeitem .....	1972	\LWR@parsepcolumn .....	2209
\LWR@latestautopage (counter) ..	<i>201</i>	\LWR@parseskipcolumn .....	2196
\LWR@lateximagedepth (counter) ..	<i>226</i>	\LWR@parsetablecols .....	2218

\LWR@prevFileDepth (counter) . . . . .	<i>121</i>	\LWR@tabletotalcols (counter) . . . . .	<i>166</i>
\LWR@printmcldata . . . . .	<i>2377</i>	\LWR@tabletotalcolsnext (counter)	<i>166</i>
\LWR@printmcctype . . . . .	<i>2347</i>	\LWR@tabular (environment) . . . . .	<i>2567</i>
\LWR@printpendingnotes . . . . .	<i>927</i>	\LWR@tabularendofline . . . . .	<i>2112</i>
\LWR@printthetitle . . . . .	<i>1530</i>	\LWR@tempheight (length) . . . . .	<i>258</i>
\LWR@ProvidesPackageDrop . . . . .	<i>291</i>	\LWR@tempraise (length) . . . . .	<i>258</i>
\LWR@ProvidesPackagePass . . . . .	<i>279</i>	\LWR@tempwidth (length) . . . . .	<i>258</i>
\LWR@pushoneclose . . . . .	<i>1094</i>	\LWR@thisfilename . . . . .	<i>951</i>
\LWR@requeststoc . . . . .	<i>1371</i>	\LWR@thisfloat (counter) . . . . .	<i>200</i>
\LWR@requirepackagenames . . . . .	<i>236</i>	\LWR@thisnewfilename . . . . .	<i>952</i>
\LWR@restoremathlatexformatting	<i>3109</i>	\LWR@tikzbabel (boolean) . . . . .	<i>245</i>
\LWR@rotboxorigin . . . . .	<i>3718</i>	\LWR@titlingmaketitle . . . . .	<i>1716</i>
\LWR@rotstyle . . . . .	<i>3608</i>	\LWR@topnavigation . . . . .	<i>539</i>
\LWR@rule . . . . .	<i>4092</i>	\LWR@traceinfo . . . . .	<i>320</i>
\LWR@scalestyle . . . . .	<i>3611</i>	\LWR@tracingl warp (boolean) . . . . .	<i>85</i>
\LWR@section . . . . .	<i>1101</i>	\LWR@validtablecol (boolean) . . . . .	<i>170</i>
\LWR@sectionnumber . . . . .	<i>1088</i>	\LWR@Verbatimclass . . . . .	<i>1800</i>
\LWR@setexpparray . . . . .	<i>467</i>	\LWR@verbtags (boolean) . . . . .	<i>150</i>
\LWR@setlatestname . . . . .	<i>2611</i>	\LWRsetnextfloat . . . . .	<i>2825</i>
\LWR@sidetoc . . . . .	<i>2950</i>	\LyX . . . . .	<i>4164</i>
\LWR@singledollar . . . . .	<i>3066</i>		
\LWR@skippingmrowcell (boolean) . . . . .	<i>163</i>		<b>M</b>
\LWR@spandepth (counter) . . . . .	<i>104</i>	Madcap (program) . . . . .	<i>27</i>
\LWR@splabel . . . . .	<i>2646</i>	\mainmatter . . . . .	<i>1078</i>
\LWR@starredlongtable (boolean) . . . . .	<i>180</i>	\makebox . . . . .	<i>3949</i>
\LWR@startedrow (boolean) . . . . .	<i>162</i>	\makeidx (package) . . . . .	<i>80</i>
\LWR@startnewdepth . . . . .	<i>1095</i>	\MakeIndex . . . . .	<i>4156</i>
\LWR@startpars . . . . .	<i>824</i>	\maketitle . . . . .	<i>55, 142, 1691</i>
\LWR@startref . . . . .	<i>2688</i>	\maketitlehookaa . . . . .	<i>134</i>
\LWR@stopars . . . . .	<i>836</i>	\maketitlehookaaa . . . . .	<i>134</i>
\LWR@stripperiod . . . . .	<i>2610</i>	margin	
\LWR@strresult . . . . .	<i>2129</i>	numbers . . . . .	<i>71</i>
\LWR@subcmidrule . . . . .	<i>2335</i>	tags . . . . .	<i>71</i>
\LWR@subhtmelementclass . . . . .	<i>656</i>	\marginpar . . . . .	<i>940</i>
\LWR@subhyperref . . . . .	<i>2746</i>	markup languages . . . . .	<i>27</i>
\LWR@subhyperrefclass . . . . .	<i>2750</i>	math	
\LWR@subinlineimage . . . . .	<i>2785</i>	font size — SVG . . . . .	<i>58, 211, 226</i>
\LWR@sublabel . . . . .	<i>2670</i>	mathjax option . . . . .	<i>74</i>
\LWR@subminipage . . . . .	<i>3861</i>	MathJax summary . . . . .	<i>58, 212</i>
\LWR@subsublabel . . . . .	<i>2649</i>	mathsvg option . . . . .	<i>74</i>
\LWR@subtableofcontents . . . . .	<i>2910</i>	SVG summary . . . . .	<i>57, 211</i>
\LWR@syncmathjax . . . . .	<i>3088</i>	math (environment) . . . . .	<i>3084</i>
\LWR@tablecolspec . . . . .	<i>2128</i>	MathJax	
\LWR@tablecolspos (counter) . . . . .	<i>166</i>	mathjax option . . . . .	<i>74</i>
\LWR@tablecolswidth (counter) . . . . .	<i>166</i>	subequations . . . . .	<i>59, 212</i>
\LWR@tabledatacolumntag . . . . .	<i>2521</i>	summary . . . . .	<i>58, 212</i>
\LWR@tabledatamulticolumntag . . . . .	<i>2501</i>	tagged equations . . . . .	<i>58, 212</i>
\LWR@tabledatamultirowtag . . . . .	<i>2507</i>	MathJax (program) . . . . .	<i>58, 212</i>
\LWR@tabledatasinglecolumntag . . . . .	<i>2289</i>	MathJax (program) [requirement] . . . . .	<i>29</i>
\LWR@tableparcell (boolean) . . . . .	<i>162</i>	mathjax (boolean) . . . . .	<i>73</i>

mathjax (option) [l warp] . . . . .	52, 74	\noteentry . . . . .	919
mathsvg (option) [l warp] . . . . .	52, 74	\notenuminnotes . . . . .	917
mcaption (package) . . . . .	348	nowidow (package) . . . . .	362
mdframed (package) . . . . .	349	\numberline . . . . .	2967
\mdseries . . . . .	3975	numbers	
memoir		left margin . . . . .	71
verse . . . . .	385		
memoir (package) . . . . .	385		
meta tag, HTML			
description . . . . .	56, 112	O	
generator . . . . .	128	OpenOffice (program) . . . . .	26
viewport . . . . .	128	options:	
\MetaLanguage . . . . .	46, 1254	[l warp-newproject]:	
microtype (package) . . . . .	52, 78, 356	BaseJobName . . . . .	268
\MiKTeX . . . . .	4163	l warpmk . . . . .	268
minipage		warpHTML . . . . .	268
horizontal space between . . . . .	257	warpprint . . . . .	268
in an \fbox . . . . .	54		
minipage (environment) . . . . .	3885	P	
\minipagefullwidth . . . . .	3877	packages	
missing sections . . . . .	44, 45	required . . . . .	77
\mrowcell . . . . .	2563	packages:	
MS-Windows (program) . . . . .	52, 87	abstract . . . . .	313
multicol (package) . . . . .	357	afterpage . . . . .	315
multirow (package) . . . . .	183, 359	algorithmicx . . . . .	316
multiline (environment) . . . . .	3197	alltt . . . . .	317
multiline* (environment) . . . . .	3220	amsmath . . . . .	81
		array . . . . .	188
N		babel . . . . .	65
\Nameref . . . . .	2743	bookmark . . . . .	317
\nameref . . . . .	2734	booktabs . . . . .	318
nameref (package) . . . . .	359	calc . . . . .	80
needspace (package) . . . . .	360	caption . . . . .	80
newclude (package) . . . . .	64, 360	changepage . . . . .	318
\NewCSS . . . . .	43, 46, 57, 888	cleveref . . . . .	243
newfloat		cmap . . . . .	52
with trivfloat, algorithmic . . . . .	382	comment . . . . .	73
\newfloat . . . . .	2	dcolumn . . . . .	319
newfloat (package) . . . . .	80	draftwatermark . . . . .	319
\NewHTMLdescription . . . . .	47, 56, 903	enumerate . . . . .	65, 320
\newline . . . . .	4033	enumitem . . . . .	65, 80
\newlist . . . . .	2046	environ . . . . .	81
\newtheorem . . . . .	156	epigraph . . . . .	321
newtxmath (package) . . . . .	64	eso-pic . . . . .	322
newunicodechar (package) . . . . .	51, 361	etoolbox . . . . .	72
nextpage (package) . . . . .	361		
\nolinebreak . . . . .	4086		
\nolinkurl . . . . .	2767		
\nopagebreak . . . . .	4090		
\notedivision . . . . .	913		

everyhook	79	setspace	366
everypage	323	showkeys	367
expl3	79	sidecap	368
extramarks	323	siunitx	64, 232
fancyhdr	324	subcaption	80
fancyvrb	79	tabularx	369
float	325	textcomp	52
floatrow	327	textpos	370
fontenc	51, 332	threeparttable	371
fontspec	51, 332	tikz	245
ftnright	333	titleps	81, 372
geometry	78, 333	titlesec	374
gettitlestring	79	titletoc	375
graphics	232, 334	titling	81, 376
graphicx	232, 334	tocloft	377
hyperref	195, 335	trivfloat	382
ifplatform	73	verse	384, 385
indentfirst	338	wallpaper	386
inputenc	51, 338	wrapfig	387
keyfloat	339	xcolor	389
kvoptions	73	xfrac	82, 392
layout	340	xifthen	80
letterspace	340	xparse	79
listings	341	xstring	80
lmodern	51	zref	81
longtable	345	\pagebreak	4087
lscape	347	\pagenote (package)	80, 362
ltcaption	347	\pagenotesubhead	915
lwarp	52	\pagenumbering	229
lwarp-newproject	268	\pageref	2729
makeidx	80	\pagerefPageFor	2728
mcaption	348	pages	
mdframed	349	inaccessible	44, 45
memoir	385	\pagestyle	225
microtype	52, 78, 356	Pandoc (program)	26
multicol	357	\paragraph	1238
multirow	183, 359	\parbox	3943
nameref	359	\parsemulticolumnalignment	2404
needspace	360	\parskip (package)	363
newclude	64, 360	\part	1192
newfloat	80	pdfcrop (program) [requirement]	29
newtxmath	64	pdfLaTeX (program) [requirement]	29
newunicodechar	51, 361	pdfseparate (program) [requirement]	
nextpage	361		29, 33
nowidow	362	pdftocairo (program) [requirement]	
pagenote	80, 362		29, 33
parskip	363	pdftotext (program) [requirement]	29, 33
placeins	363	Perl	34
ragged2e	364	perl (program) [requirement]	34
rotating	365	\phantomsection	4133

picture (environment) . . . . .	244, 3796	TeX2page . . . . .	25
placeins (package) . . . . .	363	TeX4ht . . . . .	25
Plastex (program) . . . . .	25	TtH . . . . .	25
\popclose . . . . .	440	Windows . . . . .	52, 87
Poppler . . . . .	29, 33	Word . . . . .	26
\postpublished . . . . .	134	project.css (file) . . . . .	56
\postsubtitle . . . . .	134	project.lwarpmkconf (file) . . . . .	270
\prepublished . . . . .	134	project_html.tex (file) . . . . .	270
\presubtitle . . . . .	134, 144	\published . . . . .	56, 1414
\printauthor . . . . .	135, 1547, 1562	\pushclose . . . . .	413
\printdate . . . . .	135, 1552, 1564		
\printindex . . . . .	3003	<b>Q</b>	
\printpublished . . . . .	135, 1513, 1559	\qqquad . . . . .	257, 4056
\PrintStack . . . . .	547	\quad . . . . .	257, 4051
\printsubtitle . . . . .	135, 1538, 1561	quote (environment) . . . . .	1753
\printthanks . . . . .	134, 138		
\printtitle . . . . .	135, 1522, 1560	<b>R</b>	
problems . . . . .	67	ragged2e (package) . . . . .	364
programs		\raggedleft . . . . .	130
utility . . . . .	28	\raggedright . . . . .	130
programs:		\raisebox . . . . .	3960
[requirement]:		\ref . . . . .	2717
LuaTeX . . . . .	29	\reflectbox . . . . .	3764
MathJax . . . . .	29	\renewlist . . . . .	2046
pdfcrop . . . . .	29	[requirement]:	
pdfLaTeX . . . . .	29	LuaTeX (program) . . . . .	29
pdfseparate . . . . .	29, 33	MathJax (program) . . . . .	29
pdftocairo . . . . .	29, 33	pdfcrop (program) . . . . .	29
pdftotext . . . . .	29, 33	pdfLaTeX (program) . . . . .	29
perl . . . . .	34	pdfseparate (program) . . . . .	29, 33
XeTeX . . . . .	29	pdftocairo (program) . . . . .	29, 33
Adobe . . . . .	27	pdftotext (program) . . . . .	29, 33
AsciiDoc . . . . .	26	perl (program) . . . . .	34
AsciiDoctor . . . . .	26	XeTeX (program) . . . . .	29
Asciidoc-LaTeX . . . . .	26	\RequirePackage . . . . .	255
Flare . . . . .	27	\resizebox . . . . .	3767
FrameMaker . . . . .	27	\restylefloat . . . . .	19
GELLMU . . . . .	25	\rmfamily . . . . .	3977
GladTeX . . . . .	26	\rotatebox . . . . .	3738
Hevea . . . . .	25	rotating (package) . . . . .	365
InDesign . . . . .	27	\rule . . . . .	4130
LaTeX2HTML . . . . .	25		
LaTeXML . . . . .	25	<b>S</b>	
LibreOffice . . . . .	26	sample_project.css (file) . . . . .	56, 301
Madcap . . . . .	27	\sb . . . . .	3985
MathJax . . . . .	58, 212	\scalebox . . . . .	3753
MS-Windows . . . . .	52, 87	\scshape . . . . .	3982, 3983
OpenOffice . . . . .	26	section	
Pandoc . . . . .	26	depths . . . . .	70
Plastex . . . . .	25	missing . . . . .	44, 45

\section	1216	\textless	477
\SetFirstPageTop	46, 865	\textmd	3966
\SetHTMLFileName	487	\textnormal	3974
\SetPageBottom	46, 135, 881	textpos (package)	370
\SetPageTop	46, 873	\textrm	3968
setspace (package)	366	\textsc	3973
settings		\textsf	3969
CSS project-specific	57	\textsubscript	3987
CSS selection	57	\textsuperscript	3986
HTML conversion	44	\texttt	3970
lwarw package options	52	\textup	3971
selecting output	52	\tfl@chapter@fix	382
title page	55	\thanks	56
\sffamily	3978	\thanksmarkseries	1720
\sfrac	392	\theauthor	135
showkeys (package)	367	\thedate	135
sidecap (package)	368	theindex (environment)	3011
SideTOCDepth (counter)	44, 207	\thepublished	135
\sidetocname	2947	\thesubtitle	135
siunitx (package)	64, 232	\thetitle	135
\sp	3984	\thispagestyle	227
space		threeparttable (package)	371
horizontal between minipages	257	tikz	
stack depths	70	catcodes	245
subcaption (package)	80	dollar redefined	245
subequations		tikz (package)	245
MathJax	59, 212	tikzpicture (environment)	245, 3815
\subparagraph	1245	\title	55
\subsection	1223	titlepage (environment)	55, 1438
\subsubsection	1230	titleps (package)	81, 372
\subtitle	56, 1425	titlesec (package)	374
SVG		titletoc (package)	375
math summary	57, 211	titling (package)	81, 376
mathsvg option	74	titlingpage (environment)	55, 1441, 1454
T		tocdepth (counter)	44
\tableofcontents	46	tocloft (package)	377
tabular		\tracinglwarw	68, 319
baseline	171	trivfloat	
HTML columnn conversion	171	with newfloat, algorithmic	382
\tabularx (package)	369	trivfloat (package)	382
tagged equations		troubleshooting	67
Mathjax	58, 212	\ttfamily	3979
\TeX	4136	TtH (program)	25
\TeX2page (program)	25	tutorial.tex (file)	35
\TeX4ht (program)	25	U	
\textbf	3967	Unicode	
\textcomp	52	enhanced coverage	51
\textgreater	477	file & section names	120
\textit	3972	input characters	72

selection . . . . .	51	warpHTML (environment) . . . . .	42, 47, 53
\up . . . . .	3988	warpHTML (option) [lwarp-newproject]	268
\upshape . . . . .	3980	warpHTML (option) [lwarp] . . . . .	52, 74
\url . . . . .	2774	\warpHTMLonly . . . . .	33, 47, 53
\UseLatexmk . . . . .	44	warpingHTML (boolean) . . . . .	73
\UseVerbatim . . . . .	1953	warpingprint (boolean) . . . . .	73
usingOSwindows (boolean) . . . . .	87	\warpOSwindows . . . . .	52, 87, 368
UTF-8		warpprint (environment) . . . . .	35, 47, 53
enhanced coverage . . . . .	51	warpprint (option) [lwarp-newproject]	268
file & section names . . . . .	120	warpprint (option) [lwarp] . . . . .	52, 73
selection . . . . .	51	\warpprintonly . . . . .	32, 47, 53
utility		Windows (program) . . . . .	52, 87
programs . . . . .	28	Word (program) . . . . .	26
		wrapfig (package) . . . . .	387
<b>V</b>			
Verbatim (environment) . . . . .	1809		
verbatim (environment) . . . . .	1768	xcolor (package) . . . . .	389
VerbatimClass (environment) . . . . .	1801	XeLaTeX	
\VerbatimHTMLWidth (length) . . . . .	150	detection . . . . .	77
verse (environment) . . . . .	2	file & section names . . . . .	120
verse (package) . . . . .	384, 385	\XeLaTeX . . . . .	4149
viewport		XeLaTeX (program) [requirement] . . . . .	29
HTML meta tag . . . . .	128	\XeTeX . . . . .	4149
vleftmargini (length) . . . . .	64, 149, 384	xfrac (package) . . . . .	82, 392
vleftskip (length) . . . . .	64, 149, 384, 385	xifthen (package) . . . . .	80
<b>W</b>			
wallpaper (package) . . . . .	386	xparse (package) . . . . .	79
warning icon . . . . .	71	xstring (package) . . . . .	80
warpall (environment) . . . . .	34, 53		
		zref (package) . . . . .	81
<b>Z</b>			