lwarp
LaTeX HTML5

# The **lwarp** package

## LaTeX to HTML

v0.44 — 2017/11/22

© 2016–2017 Brian Dunn
`bd@BDTechConcepts.com`

**Abstract**

The lwarp 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 pdfLaTeX, LuaLaTeX, or XeLaTeX. A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `latexmk`. Configuration is automatic at the first manual compile.

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

A modular package-loading system uses the lwarp version of a package for HTML when available. Almost two hundred LaTeX packages are supported with these high-level source compatibility replacements, and many others work as-is.

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

To update existing projects, see section 2, Updates.

**Note that this is still a "beta" version of lwarp, and some things may change in response to user feedback and further project development.**

**License:**

# 1 Supporting TEX development

TEX development:

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

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

Donations may be directed towards individual projects:

**TUG Bursary Fund:** Assistance for attending annual TUG meetings.
**CTAN:** The Comprehensive TEX Archive Network — Central storage for TEX.
**TeX Development Fund:** Support for specific projects.
**EduTeX:** Teaching and using TEX in schools and universities.
**GUST e-foundry fonts:** Enhanced for math and additional language groups.
**LaTeX Project:** Modernizing the LATEX core.
**Libre Font Fund:** Fonts, tools (FontForge), and distribution (the Open Font Library).
**LuaTeX:** Combining the pdfTEX engine and the Lua language.
**MetaPost:** Postscript graphics.
**MacTeX:** TEX for Mac.
**PDF Accessibility:** Modern PDF standards.
**Other:** Additional projects may be specified.

| | |
|---|---|
| To make a contribution: | `https://www.tug.org/donate.html` |
| For users of MiKTEX: | `https://miktex.org/donations.html` |
| For country-specific TEX users groups: | `http://tug.org/usergroups.html` |

# Contents

# List of Figures

# List of Tables

## 2 Updates

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

For a detailed list of changes, see the Change History on page 658.

**v0.44:**

koma-script
- Added koma-script classes (except scrlttr2, scrjura).
- Added scrextend, scrlayer, scrlayer-notecolumn, scrlayer-scrpage, scrhack, tocstyle, tocbasic.

HTML title and author
- Added `\HTMLTitle`. Fixed web page title if `\HTMLTitle` empty and no `\title` given and not using titling package.
- Fixed web page author if `\HTMLauthor` is empty and `\author` is not given.

encodings
- If using `pdflatex`, automatically loads T1 and UTF8 encodings. (Additional fontenc encodings may be loaded after lwarp.)

lists
- Added `list` and `trivlist` environments, hang.

tabular
- Fix: `\multicolumn` alignment if formatting for a word processor.
- Added ltxtable.

math
- Fix: MathJax combined with `lateximages`.
- algorithmicx: Improved comment symbol and floating.

packages
- Completed todonotes and luatodonotes.
- Added todo, easy-todo, fixmetodonotes, fixme.
- Added soulutf8, soulpos, cancel.
- Added section, fancyref, ifoddpage.
- Added preview, atbegshi, watermark.
- Improved tocloft `\newlistof` and `\newlistentry`.

**v0.43:**

- Docs: Reorganized HTML customization, added an HTML settings table. See section 7.3.

footnotes
- Added `FootnoteDepth` to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each `\subparagraph` or higher.

sectioning
- Fix: Expansion in section name.

`tabular`
- Fix: Ignore spaces in tabular column specification.
- Fix: Tabular rules at bottom or when finishing incomplete rows.

- Fix: `\multicolumn` at/bang/before/after specifications, trim, and vertical rules.
- Fix: supertabular and xtab column misalignment.

math

- Fix: `equation*`.
- Fix: SVG math in a section name.
- Fix: `\ref` and `\eqref` in SVG math.

packages

- Added todonotes and luatodonotes (but only `disabled`).
- Added breakurl.
- hyperref: Fix: Several macros were made robust, `\Gauge` added.

**v0.42:**

Support TeX!

- Added TeX development support page.

word-processor conversion

- Improved assitance for word-processor conversions when boolean `FormatWP` is set `true`. See section 10.

  ⚠ name change

  - The boolean `FormatWordProcessor` has been renamed `FormatWP`.

  ⚠ name change

  - The boolean `HTMLMarkFloats` has been renamed `WPMarkFloats`.
  - New booleans control whether to place additional marks around minipages, at the table of contents, at the LOF and LOT, and whether to print math as LaTeX source for copy/paste into the LibreOffice Writer TeXMaths extension.
  - Improved formatting for numerous objects. See section 10.

`tabbing`

- Add: `tabbing` environment.

overpic

- Add: overpic package. See section 202.

math

- Fix: Text copy/paste of AMS math environment numbers and names.
- Improved `\ensuremath`.
- MathJax with siunitx: Updated script and documentation.

symbols

- textcomp: Improved `\interrobangdown`.
- realscripts: Fix for subscripts in a `lateximage`.

load order

- morewrites: Enforces loading before lwarp.

**v0.41:**

`tabular`

- Added tabular vertical rules, subject to some limitations. See the rules section of section 8.7.
- Improved booktabs: Width and trim are honored.

⚠ new syntax

- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 192.2 on page 545.

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

**v0.40:**

graphics, graphicx

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

`\includegraphics` path

⚠ image file extensions

- Improved graphics `\graphicspath` support. Multiple image directories may now be used. Refer to `.pdf` files without a file extension to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 8.6.
- grffile is now directly supported instead of emulated.

bigdelim

- Fix for bigdelim, and improved documentation. See section 101.

symbols

- Improved LaTeX and textcomp symbols.

fixes

- Fix for LaTeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for xltxtra with XeLaTeX.
- Fixes for tocbibind with `\simplechapter`, etc.
- Fixes for `\multicolumnrow` and `\nullfonts` with older versions of multirow and xparse.
- Added `\underline`.

margins

- Added adjmulticol.

columns

- Added cuted, midfloat.

footnotes

- Added pfnote, fnpos, dblfnote.

tabular

- Added stabular, tabls.

sectioning

- Added sectsty, anonchap, quotchap.

**v0.39:**

title pages

⚠ `\published` and `\subtitle`

⚠ load order

- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. titling is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubtitlePublished` replicates them using titling. See section 55.7. authblk is added, and should be loaded before titling. See section 55.

tabular

- `\multirow` now supports the new optional `vpos` argument.

multi column/row cell

- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 192.2.
- Tabular special cases:

⚠ macros inside tabular

  - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 8.7.

⚠ tabular defined inside another environment

- – Added \ResumeTabular for use when a tabular environment is defined inside another environment. See section 8.7.

tabular

- Added supertabular, xtab, bigstrut, bigdelim.

margins

- Added fullwidth.

page layout

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

**v0.38:**

forced single-pass compile

- Added lwarpmk print1 and lwarpmk html1 actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.

starred sections

- Added \ForceHTMLPage and \ForceHTMLTOC to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 8.4.1.

updated tutorial

- Modified the tutorial to use the new \ForceHTMLPage and \ForceHTMLTOC macros.

packages

- Added appendix, tocbibind, fncychap, fix2col.

font size

- Added relsize, scalefnt.
- Added realscripts, metalogo, xltxtra.
- Added grffile, romanbar.

page numbering

- Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.

front & back matter

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

**v0.37:**

\include for HTML

- \include now maintains independent .aux files for HTML versions.

latexmk

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

accents and symbols

- Improved support for LaTeX accents, textcomp, siunitx symbols.

babel-french

- Improved babel-french handling for load order and ~ tilde.

**v0.36:**

- Recorganized the documentation section regarding special cases and limitations. (Section 8)
- Improved source formatting.

boxes and frames

- \fbox and related now use \fboxsep and \fboxrule.
- \makebox and \framebox now use width and position.

- \fcolorbox and related now work inside a lateximage.
- babel-french babel-french: Improvements for French variants, load order, footnotes, ellipses.
- footnotes Improved footnote numbering. lateximage footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.
- siunitx Improved siunitx units.
- Fix for filenames while using MathJax.
- Fix for \rule when xcolor is not loaded.
- Added transparent, upref.

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

**v0.34:**

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

  BlockClass[style]{class} (NEW)

  instead of:

  BlockClass{class}[style] (OLD)

  This change makes it more consistent with LaTeX standards, and avoids problems with space between arguments.
- ⚠ Optional arguments Likewise, \InlineClass's optional argument now comes before the mandatory arguments:

  \InlineClass[style]{class}{text}
- spans with minipages framing minipages Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an \fbox containing a minipage, although with minimal HTML fomatting. See section 8.2.3. \fboxBlock is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as \fbox for print output. The fminipage environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 8.2.5. \fbox and minipages now often work in SVG math and lateximages. MathJax supports \fbox, but not \fboxBlock nor fminipage.
- lateximage, SVG math, tabular Improved compatibility between lateximage and minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox, tabular, booktabs.
- Improved font control for lateximagees and SVG math.
- eqnarray Added the eqnarray environments.
- verbatim packages fancyvrb is no longer required (preloaded), but is still supported.
- Added verbatim and moreverb.

framing packages

- Added fancybox, boxedminipage2e and shadow.

list packages

- enumitem is no longer required, but is still supported.
- Added enumerate and paralist.
- titleps is no longer required, but is still supported.
- Added crop.
- Added rotfloat, marginfit, and several minor packages; see the change log.

babel-french

- Adds fixed-width HTML spaces around punctuation when using babel-french. LuaTEX does not yet use the extra punctuation spacing.

**v0.33:**

- Tabular `@` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 61.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for amsmath with SVG math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

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

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

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

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

**v0.30:**

⚠ lwarp-newproject

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

  To modify existing documents, remove from the document source:

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

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

⚠ HTML setup changes.

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

| Old Macro | New Package Option |
|---|---|
| `\HomeHTMLFileName` | `HomeHTMLFilename` |
| `\HTMLFileName` | `HTMLFilename` |
| `\useLatexmk` | `latexmk` |
| `\warpOSwindows` | `OSWindows` |

| Old Package Option | New Package Option |
|---|---|
| `lwarpmklang` | `IndexLanguage` |
| (new) | `xdyFilename` |

| Old Macro | New Macro |
|---|---|
| `\MetaLanguage` | `\HTMLLanguage` |
| `\HTMLauthor` | `\HTMLAuthor` |
| `\NewHTMLdescription` | `\HTMLDescription` |
| `\SetFirstPageTop` | `\HTMLFirstPageTop` |
| `\SetPageTop` | `\HTMLPageTop` |
| `\SetPageBottom` | `\HTMLPageBottom` |
| `\NewCSS` | `\CSSFilename` |

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

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

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

  - For pdflatex and texindy, the -C utf8 option is used. This is supported in modern distributions, but a customized lwarpmk.lua may need to be created for use with older distributions.

**v0.29:**

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

**v0.28:**

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

# 3 Introduction

The lwarp project aims to allow a rich LaTeX document to be converted to a reasonable HTML interpretation. No attempt has been made to force LaTeX to provide for every HTML-related possibility, and HTML cannot exactly render every possible LaTeX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

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

`pdflatex`, `xelatex`, or `lualatex` is used, allowing lwarp to process the usual image formats. While generating HTML output, SVG files are used in placed 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 Ti*k*z 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[1] 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 LaTeX source for the expression, allowing it to be copy/pasted into other documents.[2] Custom LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside LaTeX.

The MathJax JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files, one per math expression. lwarp maintains LaTeX control for cross-referencing and equation numbering / formatting.

---

The lwarp package allows LaTeX to directly generate HTML5 tags from a LaTeX source document, with only minor intervention on the user's part. A `texlua` program called `lwarpmk` is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from a LaTeX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

[1] Firefox has had an on-again/off-again bug for quite some time regareding printing SVGs at high resolution.

[2] There seems to be some debate as to whether MathML is actually an improvement over LaTeX for sharing math. The author has no particular opinion on the matter, except to say that in this case LaTeX is much easier to implement!

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

When requesting packages during HTML conversion, `lwarp` first looks to see if it has its own modified version to use instead of the usual LaTeX version. These `lwarp-packagename.sty` files contain code used to emulate or replace functions for HTML output.

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

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

## 3.1 Supported packages and features

Supported classes include book, report, and article, and the Koma-script classes scrbook, scrreprt, and scrartcl. memoir is planned, but in the meantime many of the packages used by memoir are already supported.

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

Table 1: LaTeX–HTML generation — lwarp package — Supported functions

| Category | Status |
|---|---|
| Engines: | pdfLaTeX, XeLaTeX, LuaLaTeX |
| Classes: | book, report, article, scrbook, scrreprt, scrartcl. memoir is planned. |
| Koma-script: | scrextend, scrlayer, scrhack. Others as listed below. |

lwarp Supported Functions — continued

| Category | Status |
|---|---|
| Page layout: | Emulates geometry, fancyhdr, titleps, scrlayer-scrpage, typearea, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, ebook, preview, draftwatermark, watermark, everyshi, atbegshi. |
| Sectioning: | Supported, with hyperlinks. Adds `FileDepth` for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames. Emulates titlesec, fncychap, sectsty, section, anonchap, quotchap. |
| Table of contents, figures, tables: | Supported, with hyperlinks. Supports tocbibind. Emulates titletoc, tocloft, tocbasic, and tocstyle. |
| Title page: | `\maketitle`, `titlepage`, titling, authblk. |
| Front & back matter: | abstract, appendix |
| Indexing: | `texindy` is used, with hyperlinks. idxlayout is emulated. |
| Glossary: | glossaries and `xindy` are used. |
| Bibliography: | Supported text-only. |
| Cross-references: | Emulated, with hyperlinks. hyperref, cleveref, varioref, fancyref, url, breakurl. |
| Languages: | babel. (polyglossia is untested.) |
| Margin notes: | marginfit, marginfix, scrlayer-notecolumn. |
| Footnotes: | Adds `FootnoteDepth` to print footnotes at section breaks. footnote, footmisc, marginnote, sidenote, pagenote, endnotes. |
| Math: | Supported. Converted to SVG images with HTML `<alt>` tags containing the LaTeX source for the math expression. MathJax supported as an alternative. $\mathcal{AMS}$ environments are supported. User-defined macros are available during converson, due to native LaTeX processing. |
| Theorems: | Support for native LaTeX theorems, plus theorem, amsthm, ntheorem. |

lwarp Supported Functions — continued

| Category | Status |
|---|---|
| Add'l math: | delarray, bm, math fonts via SVG images. |
| Units and fractions: | siunitx, xfrac, nicefrac, units |
| Floats: | Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt. |
| Tabular | `tabular` environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, ltxtable, booktabs. |
| Graphics | graphics and graphicx are emulated. `\includegraphics` supports `width`, `height`, `origin`, `angle`, and `scale` tags, and adds `class`. References to PDF files are changed to SVG, other image types are accepted as well. `\rotatebox` and `\scalebox` are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture and tikz are converted to an SVG image. grffile and overpic are supported. |
| xcolor: | Supported. Full package color names, any color models, and mixing. `\textcolor`, `\colorbox`, `\fcolorbox`. Enhanced for HTML compatibility. |
| Lists: | Standard LaTeX environments are supported, along with enumitem, enumerate, paralist, hang. |
| Environments: | Standard LaTeX environments are supported. |
| `minipage`: | Supported with some HTML5-imposed limitations. Nested minipages are supported. |
| Quotations: | verse, csquotes, epigraph |
| Verbatim: | verbatim, moreverb, fancyvrb (except for verbatim footnotes). |
| Frames: | framed, fancybox, mdframed, boxedminipage2e, shadow. |

lwarp Supported Functions — continued

| Category | Status |
|---|---|
| multicol: | Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported. |
| `fullwidth`: | Emulated. |
| Todo: | todo, todonotes, easy-todo, fixmetodonotes, fixme. |
| Direct formatting: | `\emph`, `\textsuperscript`, `\textbf`, etc are supported. `\bfseries`, etc. are not yet supported. lettrine, ulem, soul, soulutf8, soulpos, cancel, relsize, scalefnt, and realscripts are supported. |
| Ordinals: | nth, fmtcount, and engord are supported. |
| Text ligatures: | Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly. |
| Horizontal space: | HTML output for thin-unbreakable, unbreakable, `\enskip`, `\quad`, `\qquad`, `\hspace`. |
| Rules: | `\rule` with width, height, raise, text color. |
| HTML reserved characters: | `\&`, `\textless`, and `\textgreater` are converted to HTML entities. |
| Fonts: | Used as-is. Appear in math expressions or embedded image environments. |
| Symbols: | Native LaTeX diacriticals, textcomp. |
| Working as-is: | Various utility, calculation, file, and text-only packags, such as calc, fileerr, somedefs, trace, xspace. |
| Where: | |
| **Supported:** | The existing LaTeX package is used. |
| **Emulated:** | The LaTeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility. |

# 4 Alternatives

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

## 4.1 Internet class

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

## 4.2 TeX4ht

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

Prog `htlatex`

This system uses native LaTeX processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In most cases, lwarp provides a better HTML conversion, while supporting more packages. TeX4ht produces several other forms of output beyond HTML.

## 4.3 Translators

These systems use external programs to translate a subset of LaTeX syntax into HTML. Search for each on CTAN (`http:\ctan.org`).

Prog `Hevea` **HEVEA:** `http://hevea.inria.fr/` (not on CTAN)

Prog `TtH` **TTH:** `http://hutchinson.belmont.ma.us/tth/`

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

Prog `LaTeXML` **LaTeXML:** `http://dlmf.nist.gov/LaTeXML/`

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

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

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

Finally, GladTEX may used to directly insert LATEX math into HTML:

Prog GladTeX

**GladTEX:** http://humenda.github.io/GladTeX/

## 4.4 AsciiDoc

AsciiDoc is one of the most capable markup languages, providing enough features to produce the typical technical-writing document with cross-references, and it writes LATEX 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 LATEX-related features.

**Asciidoctor-LateX:**
http://www.noteshare.io/book/asciidoctor-latex-manual
https://github.com/asciidoctor/asciidoctor-latex

Prog Asciidoctor-LaTeX

## 4.5 Pandoc

Prog Pandoc

A markup system which also reads and writes LATEX and HTML.

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

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

## 4.6 Word processors

Prog Word
Prog LibreOffice
Prog OpenOffice

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

## 4.7 Commercial systems

Prog `Adobe`
Prog `FrameMaker`
Prog `InDesign`
Prog `Flare`
Prog `Madcap`

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 ™.

## 4.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that LaTeX is comparably easy to learn, while LaTeX provides many more advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

Text-based document-markup systems share some of the advantages of LaTeX vs. a typical word processor. Documents formats are stable. The documents themselves are portable, work well with revision control, do not crash or become corrupted, and are easily generated under program control. Formatting commands are visible, cross-referencing is automatic, and editing is responsive. Search/replace with regular expressions provides a powerful tool for the manipulation of both document contents and structure. Markup systems and some commercial systems allow printed output through a LaTeX back end, yielding high-quality results especially when the LaTeX template is adjusted, but they lose the ability to use LaTeX macros and other LaTeX source-document features.

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

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

# 5 Installation

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

## 5.1 Installing the lwarp package

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

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

Enter ⇒ `kpsewhich lwarp.sty`

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

**TeX Live:** If using a TeX Live distribution, try installing via `tlmgr`:

Enter ⇒ `tlmgr install lwarp`

**MiKTeX:** If using MiKTeX, try using the package installer to install the package `lwarp`. Also update the package `miktex-misc`, which will install the `lwarpmk` executable.

**Operating-system package:** The operating-system package manager may already have `lwarp`, perhaps as part of a set of TeX-related packages.

**CTAN TDS archive:** lwarp may be downloaded from the Comprehensive TeX Archive:

1. See `http://ctan.org/pkg/lwarp` for the lwarp package.
2. Download the TDS archive: `lwarp.tds.zip`
3. Find the TeX `local` directory:

   **TeX Live:**

   Enter ⇒ `kpsewhich -var-value TEXMFLOCAL`

   **MiKTeX:**

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

   This should be something like:

   `/usr/local/texlive/texmf-local/`
4. Unpack the archive in the TDS local directory.
5. Renew the cache:

   Enter ⇒ `mktexlsr`

   — or —

   Enter ⇒ `texhash`

Table 2: Required software programs

**Provided by your LaTeX distribution:**

From TeXLive: http://tug.org/texlive/.

**LaTeX:** `pdflatex`, `xelatex`, or `lualatex`.

**The lwarp package:** This package.

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

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

**xindy:** The xindy package is used by lwarp to create indexes. On a MiKTeX 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 LaTeX code. On a MiKTeX system, Perl may need to be installed first.

**pdfcrop:** Used to pull images out of the LaTeX PDF.

**Poppler PDF utilities:**

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

**pdfseparate:** Used to pull images out of the LaTeX PDF.

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

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

From Poppler: poppler.freedesktop.org.

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

Enter ⇒ `brew install poppler`

For Windows, see:
https://sourceforge.net/projects/poppler-win32/ and:
http://blog.alivate.com.au/poppler-windows/

**Perl:**

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

perl.org, 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

Or, for Windows MiKTEX, start the program called `MiKTeX Settings (Admin)` and click on the button called Refresh FNDB.

**CTAN `.dtx` and `.ins` files:** Another form of 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 lwarp package.
2. Download the zip archive `lwarp.zip` into your own `lwarp` directory.
3. Unpack `lwarp.zip`.
4. Locate the contents `lwarp.dtx` and `lwarp.ins`
5. Create the documentation:

   Enter ⇒ `pdflatex lwarp.dtx`

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

   Enter ⇒ `pdflatex lwarp.ins`
7. Copy the `.sty` files somewhere such as the 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 5.2.1 to generate your local copy of `lwarpmk`.
11. Once the local version of `lwarpmk.lua` is installed, it may be made available system-wide as per section 5.2.

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

Just testing!

## 5.2 Installing the lwarpmk utility

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

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

1. At a command line, try executing lwarpmk. If the lwarpmk help message appears, then lwarpmk is already set up. If not, it is easiest to generate and use a local copy. See section 5.2.1.

2. For MiKTeX, try updating the miktex-misc package. This may install the lwarpmk executable for you.

   Otherwise, continue with the following:

3. Locate the file lwarpmk.lua, which should be in the scripts directory of the TDS tree. On a 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 5.2.1.)

4. Create lwarpmk:

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

   (a) Locate the TeX Live binaries:
   Enter ⇒ `kpsewhich -var-value TEXMFROOT`
   This will be something like:
   `/usr/local/texlive/<year>`
   The binaries are then located in the bin/<arch> directory under the root:
   `/usr/local/texlive/<year>/bin/<architecture>/`
   In this directory you will find programs such as pdflatex and makeindex.

   (b) In the binaries directory, create a new symbolic link from the binaries directory to lwarpmk.lua:
   Enter ⇒ `ln -s <pathtolwarpmk.lua> lwarpmk`

   (c) Make the link executable:
   Enter ⇒ `chmod 0755 lwarpmk`

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

   (a) Locate the TeX Live binaries as shown above for Unix.

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

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

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

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

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

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

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

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

### 5.2.1 Using a local copy of lwarpmk

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

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

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

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

3. `lwarpmk.lua` may be used for this project:

   **Unix:**

   (a) Make `lwarpmk.lua` executable:

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

   (b) Compile documents with

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

   etc.

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

   **Windows:**

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

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

   etc.

   Or:

   ```
   Enter ⇒  lwarpmk html
   Enter ⇒  lwarpmk print
   ```

   etc.

## 5.3 Installing additional utilities

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

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 51.

Enter ⇒ `luatex -version`

Enter ⇒ `xindy -version`

Enter ⇒ `latexmk -version`

Enter ⇒ `perl -version`

Enter ⇒ `pdfcrop -version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate -version`

Enter ⇒ `pdftocairo -v`

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

The TEX utilities `xindy`, `latexmk`, and `pdfcrop` may be provided by your operating system's package manager, and are also provided by the CTAN archive:

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

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

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

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

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

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

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

1. See table 2 on page 51.
2. Download and extract the Poppler utilities `pdftotext`, `pdfseparate`, and `pdftocairo` to a directory, such as `Poppler`.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.

6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the bin directory of the Poppler utilities, such as:

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

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

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

1. Download and install a version of Perl, such as Straweberry Perl, to a directory without a space in its name, such as C:\Strawberry.
2. Edit the Path as seen above for the Poppler utilities.
3. Enter the bin directory of the Perl utility, such as:

   ```
   C:\Strawberry\perl\bin
   ```

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

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

# 6 Tutorial

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

## 6.1 Starting a new project

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

File `tutorial.tex`

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

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

   **Copy from the lwarp documentation directory:**

   File `lwarp_tutorial.txt`

   ⚠ Note: `.txt` suffix!

   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`.

   ⚠ Bad formatting!

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

3. Compile the project:

   Enter ⇒
   ```
   pdflatex tutorial.tex
   ```

   (several times)

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

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

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

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

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

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

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

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

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

```
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{lwarp_sagebrush.css}

\begin{document}

\maketitle                      % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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


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

\end{document}
```

Table 3: Files created along with the print version

**`tutorial.pdf`:** The PDF output from LaTeX. The print version of the document.

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

**Auxiliary files:** The usual LaTeX files `.aux`, `.log`, `.out`, `.toc`, `.lof`, `.idx`. When an HTML version of the document is created, `_html` versions of the auxiliary files will also be generated.

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

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

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

---

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

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

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

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

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

## 6.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

   Enter ⇒ `lwarpmk print`

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

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

3. Recompile again.

   Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk print`

   `lwarpmk again` updates the date code for the file, triggering a recompile the next time the document is made.[3]

5. Process the index.[4][5]

   Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

   Enter ⇒ `lwarpmk print`

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

   Enter ⇒ `lwarpmk print1`

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

[3] 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.

[4] A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

[5] Also see section 8.4.9 for index options.

## 6.3 Compiling the HTML version with `lwarpmk`

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

1. Compile the HTML version:

   Enter ⇒ `lwarpmk html`

   (a) `lwarpmk` uses LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.

   (b) `pdftotext` is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.

   (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`[6], `Some-math.html`, and the document's index in `_Index.html`.[7]

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 LaTeX source until the images of the math expressions have been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 6.4 and 6.5.

3. Force a recompile:

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk html`

   Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile:[8][9]

   Enter ⇒ `lwarpmk htmlindex`

   Enter ⇒ `lwarpmk html`

   `_Index.html` is updated for the new LaTeX index.

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

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

   Enter ⇒ `lwarpmk html1`

[6] `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`.

[7] `index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

[8] A `lwarpmk htmlglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

[9] Also see section 8.4.9 for index options.

## 6.4 Generating the SVG images

math as SVG images

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

`picture` and Ti*k*z

lwarp uses the same mechanism for `picture` and Ti*k*z environments.

1. Create the SVG images:

   Enter ⇒ `lwarpmk limages`

   Enter ⇒ `lwarpmk html`

2. Move to the tutorial's math page and reload.

3. The math images are displayed using the same font and formatting as the printed version.

4. Copy/paste a math expression into a text editor to see the LaTeX source.

⚠ Adding/removing

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

⚠ HTML instead of images

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

⚠ Lots of files!

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

## 6.5 Using MathJax for math

math with MathJax

Math may also be represented using the MathJax Javascript project.

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

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

2. Recompile

   Enter ⇒ `lwarpmk html`

3. Reload the math page.

⚠ MathJax requirements

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

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

## 6.6 Changing the CSS style

For a formal css style, add to the preamble:

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

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

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

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

## 6.7 Customizing the HTML output

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

See section 7.3 for more information.

## 6.8 Using `latexmk`

`latexmk` is a LaTeX utility used to monitor changes in source files and recompile as needed.

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

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

2. Recompile the printed version of the document.

   Enter ⇒ `lwarpmk print`

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

3. Recompile the document.

   Enter ⇒ `lwarpmk print`

   and/or

   Enter ⇒ `lwarpmk html`

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

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

Enter ⇒ `lwarpmk print1`

and/or

Enter ⇒ `lwarpmk html1`

## 6.9 Using XeLaTeX or LuaLaTeX

XƎLaTeX or LuaLaTeX may be used instead of LaTeX.

1. Remove the auxiliary files for the project:

   Enter ⇒ `lwarpmk cleanall`

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

   Enter ⇒ `xelatex tutorial.tex`

   -or-

   Enter ⇒ `lualatex tutorial.tex`

   When the recompile occurs, the configuration files for `lwarpmk` are modified to remember which TeX engine was used. XƎLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

3. To recompile the document:

   Enter ⇒ `lwarpmk print`

   -and-

   Enter ⇒ `lwarpmk html`

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

## 6.10 Using a glossary

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

Opt IndexLanguage

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

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

To process the glossary for the print version:

Enter ⇒ `lwarpmk printglossary`

To process the glossary for the HTML version:

Enter ⇒ `lwarpmk htmlglossary`

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

## 6.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

## 6.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

## 6.13 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with lwarkmk:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

## 6.14 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

# 7 Additional details

## 7.1 Font and UTF-8 support

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

pdflatex, T1, UTF8

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

vector fonts
Computer Modern
⚠

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

```
usepackage{lmodern}
```

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

```
\usepackage{dejavu}
```

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

⚠ XeLaTeX and LuaLaTeX users must use the fontspec package. Do NOT use fontenc!

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

1. documentclass{article/book/report} goes here, followed by any of:
2. Font and UTF-8 related commands:
   - For XeLaTeX or LuaLaTeX:
     - Pkg fontspec — fontspec and font choices

       ligatures

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

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

   - For pdflatex:
     - Pkg lmodern — lmodern or other font-related packages

Pkg fontenc
- fontenc

Pkg inputenc
- inputenc

Pkg newunicodechar
- newunicodechar

File glyphtounicode
- `\input glyphtounicode.tex`
- `\input glyphtounicode-cmr.tex% from the pdfx package`
- `\pdfgentounicode=1`

Pkg cmap
- cmap

Pkg textcomp
- textcomp

Pkg microtype

ligatures
- microtype is automatically used by lwarp to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off TEX ligatures.

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

### 7.1.1 Indexes and UTF-8

lwarp uses the `xindy` program to processes indexes.

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

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

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

Table 4: Package options

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

## 7.2 lwarp package loading and options

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

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

Opt warpprint
Opt warpHTML
Usually controlled by lwarpmk, and not set in the document. 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 lwarp 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{lwarp} without any options to create a printed version, while <project>_html.tex will create an HTML version.

Opt mathsvg
Opt mathjax
For math display, select mathsvg (default), or mathjax. For more information about the math options, see section 8.5.

Opt `OSWindows` See section 7.5 if using Windows.

Opt `BaseJobname` Not intended for the user. Used internally by lwarp when creating the `*_html.tex` file used to compile the HTML version. See section 21.

Opt `HomeHTMLFilename` See section 7.3.

Opt `HTMLFilename` See section 7.3.

Opt `IndexLanguage` If using an index or glossary, see section 21.

Opt `latexmk` Has `lwarpmk` use `latexmk` to recompile the document several times if necessary. Otherwise, `lwarpmk` attempts to determing for itself whether to recompile. See section 7.3.

Opt `lwarpmk` If you wish to have lwarp generate a local copy of `lwarpmk.lua` for archival or local-installation purposes, compile the print version with the `lwarpmk` option set. See section 21.

Opt `xdyFilename` The default `xindy` filename is `lwarp.xdy`. If you wish to use a custom `.xdy` file for index generation, see section 21.

Table 5: HTML settings

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

## 7.3 Customizing the HTML output

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

⚠ Placement!

⚠ Changes!

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

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

   Enter ⇒ `lwarpmk clearall`

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

   Enter ⇒ `lwarpmk print`

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

   Enter ⇒ `lwarpmk html`

**Options for the lwarp package:**

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

Opt HomeHTMLFilename
Default: \BaseJobname

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

```
HomeHTMLFilename=index
```

filename underscores

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

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

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

Opt HTMLFilename
Default: <empty>

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

Opt latexmk
Default: false

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

Opt mathsvg
Default: true

Opt mathjax
Default: false

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

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

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

Ctr tocdepth

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

Ctr SideTOCDepth
Default: 1

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

sideTOC

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

It is recommended to set:

```
SideTOCDepth = FileDepth
```

or

```
SideTOCDepth = FileDepth+1
```

⚠ inaccessible pages

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

Ctr FileDepth
Default: -5

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

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

Bool CombineHigherDepths
Default: true

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

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

⚠ Inaccesible pages!

⚠ Lost in an old page!

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

Bool FileSectionNames
Default: true

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

Ctr FootnoteDepth
Default: 5

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

\abstractname
Default: Abstract

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

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

`\CSSFilename`
Default: `lwarp.css`

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

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

`\HTMLLanguage`
Default: `en-US`

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

`\HTMLTitle`
Default: `\thetitle`

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

`\HTMLAuthor`
Default: `\theauthor`

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

`\HTMLDescription`
Default: `<empty>`

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

`\HTMLFirstPageTop`
Default: `<empty>`

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

`\HTMLPageTop`
Default: `<empty>`

**`\HTMLPageTop:`** `{`⟨*contents*⟩`}` A user-definable custom action applied to the top of pages other than the home page. Useful for logos, etc. Defaults empty. `\LinkHome` may be used to place a link back to the homepage. Ignored in print output.

`\HTMLPageBottom`
Default: `<empty>`

**`\HTMLPageBottom:`** `{`⟨*contents*⟩`}` A user-definable custom action applied to the bottom of each web page. Useful for authors, copyright notices, contact information, etc. Defaults empty. `\LinkHome` may be used to place a link back to the homepage. Ignored in print output.

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

\tableofcontents
⚠ TOC on the homepage!

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

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

**Placed in the document wherever necessary:**

Env warpprint

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

Env warpHTML

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

\warpprintonly

**\warpprintonly:** {⟨*contents*⟩} A macro version of the warpprint environment.

\warpHTMLonly

**\warpHTMLonly:** {⟨*contents*⟩} A macro version of the warpHTML environment.

### 7.3.1 Example HTML file naming

Examples of ways to name or number HTML files:

**Numbered HTML nodes:**

Example: Homepage index.html, and node-1, node-2. [10]

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

[10] See \SetHTMLFileNumber to number in groups by chapter, for example.

**Named HTML sections, no prefix:**

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

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

**Named HTML sections, with prefix:**

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

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

## 7.4 Customizing the CSS

\CSSFilename
Default: lwarp.css

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

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

The styles provided by lwarp include:

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

**lwarp_formal.css:** A formal style with a serif fonts and a traditional look.

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

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

Custom css

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

⚠ Rename it!

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

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

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

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

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

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

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

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

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

## 7.5 Selecting the operating system

Prog Unix
Prog Mac OS
Prog Linux
Prog MS-Windows
Prog Windows
Opt OSWindows

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

If WINDOWS is not correctly detected, use the lwarp option OSWindows.

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

## 7.6 Selecting actions for print or HTML output

The following environments and macros are used to select actions which only apply to either traditional 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.

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

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

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

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

Also see section 12: Troubleshooting.

## 7.8 Title page

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

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

The title is used in the meta tags in the HTML files, unless overridden by `\HTMLTitle`, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 55.7.

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

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

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

`\title` {⟨*title*⟩}

⚠ Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 55.7). The title will appear in the document `\maketitle` as a heading `<h1>`. The HTML meta `title` tag will also have this title, unless `\HTMLTitle` is used to set the meta title to something else instead.

`\author` {⟨*author*⟩}

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

`\date` {⟨*date*⟩}

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

`\thanks` {⟨*text*⟩}

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

## 7.9 HTML page meta descriptions

`\HTMLDescription` {⟨*A description of the web page.*⟩} The default is no description.

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

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

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

```
\HTMLDescription{}
```

### 7.10 HTML page meta title

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

### 7.11 HTML page meta author

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

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

### 7.12 Modifying `xindy` index processing

Prog xindy
File lwarp.xdy

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

To customize index processing:

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

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

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

Opt xdyFilename

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

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

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

# 8 Special cases and limitations

Also see section 12: Troubleshooting.

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

## 8.1 Formatting

### 8.1.1 Text formatting

⚠ \bfseries, etc. — \textbf, etc. are supported, but \bfseries, etc. are not yet supported.

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

### 8.1.2 Horizontal space

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

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

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

### 8.1.3 Text alignment

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

### 8.1.4 Accents

Native LaTeX accents such as \" will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

### 8.1.5 Textcomp

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

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

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

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

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

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

### 8.1.8 Filenames and URLs in lists or footnotes

filename underscore — Escape underscores in the filenames:

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

### 8.1.9 relsize package

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

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

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

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

## 8.2 Boxes and minipages

### 8.2.1 Marginpars

`\marginpar` [⟨*left*⟩] {⟨*right*⟩} `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `<br />` tags.

`\marginparBlock` `[⟨left⟩] {⟨right⟩}` To include block-related macros, use `\marginparBlock`, which takes the same arguments but creates a `<div>` instead of a `<span>`. A line break will occur in the text where the `\marginBlock` occurs.

### 8.2.2 Save Boxes

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

### 8.2.3 Minipages

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

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

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

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

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

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

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

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

### 8.2.4 Side-by-side minipages

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

### 8.2.5 Framed minipages and other environments

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

`\fbox`
`\fboxBlock`
Env `fminipage`

For a framed object, options include:

**To remove the frame in HTML output:** Place the `\fbox` command and its closing brace inside `warpprint` environments. This will nullify the frame for HTML output.

For inline text:

**To frame the contents inline with some formatting losses in HTML:** This is the default action of `\fbox` when enclosing a minipage. During HTML output, `\fbox` nullifies the HTML tags for `minipage`, `\parbox`, and lists. The contents are included as inline text inside the `\fbox`'s `<span>` of class `framebox`. For lists, line breaks are converted to HTML breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra HTML markup.

For inline `minipage` and lists:

**To frame the contents on their own line with improved formatting in HTML:** A new command `\fboxBlock` is included, intended to be a direct replacement for `\fbox` for cases where the `\fbox` surrounds a minipage, table, or list. For print output, this behaves as `\fbox`. For HTML output, the contents are placed inside an HTML `<div>` with the class `framed`, resulting in the contents being placed on their own line with a frame surrounding them. The contents preserve their HTML formatting, so lists and minipages look nicer, and valid HTML is created for a `tabular`. While an `\fbox` containing a `tabular` is valid LaTeX code, the result in HTML is problematic since a table is a `<div>` not a `<span>`, so use `\fboxBlock` around a `tabular`, or else place the `tabular` inside a `minipage`, or use `fminipage`, described next. Also see below regarding the "Misplaced alignment tab character &" error.

For display `tabular`, minipages, and lists:

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

colored boxes and frames:

**To create colored frames and boxes:** See section 272 for xcolor's \colorbox and \fcolorbox, and lwarp's additional \colorboxBlock and \fcolorboxBlock.

⚠ Misplaced alignment tab character &

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

**To frame equations:** See section 133 for the fancybox package.

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

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

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

### 8.2.6 fancybox package

framed equation example

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

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

framing alternatives

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

framed table example

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

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

framed verbatim

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

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

framed \VerbBox

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

indented alignment

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

### 8.2.7 mdframed package

support

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

⚠ loading

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

font

For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

theoremtitlefont

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

footnotes

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

ignored options

`userdefinedwidth` and `align` are currently ignored.

## 8.3 Cross-references

labels
⚠ underscores

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

`\nameref`
⚠ empty link

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

### 8.3.1 Page references

⚠ LaTeX page numbers

The printed page does not translate to the HTML page, so `\pageref` references are converted to parentheses containing `\pagerefPageFor`, which defaults to "see ", followed by a hyperlink to the appropriate object.

Ex:

```
\ref{sec:name} on page \pageref{sec:name}
```

in HTML becomes:
"Sec. 1.23 on page (**see** sec. 1.23)".

`\pagerefPageFor` may be redefined to "page for ", empty, etc. See page 345.

### 8.3.2 cleveref and varioref packages

⚠ cleveref page numbers

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

Ex:

```
    \cpageref{tab:first,tab:second}
 in HTML becomes:
```

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

See `\cpagerefFor` at page 402 to redefine the message which is printed for page number references.

### 8.3.3 Hyperlinks, hyperref, and url

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

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

### 8.3.4 Footnotes and page notes

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

⚠ pfnote numbers

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

## 8.4 Front and back matter

### 8.4.1 Starred chapters and sections

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

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

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

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

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

⚠ inaccessible HTML page

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

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

### 8.4.2 abstract package

⚠ missing TOC

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

### 8.4.3 titling and authblk

package support

⚠ load order

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

\published and \subtitle

If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 55.7.

### 8.4.4 tocloft package

Opt tocloft titles

⚠ tocloft & other packages

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

### 8.4.5 appendix package

⚠ incorrect TOC link

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

### 8.4.6 pagenote package

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

### 8.4.7 endnotes package

table of contents

To place the endnotes in the TOC, use:

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

HTML page

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

```
\ForceHTMLPage
\theendnotes
```

### 8.4.8 glossaries package

Pkg glossaries

`xindy` is required for glossaries.

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

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

placement and TOC options

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

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

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

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

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

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

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

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

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

### 8.4.9 Index and the tocbibind package

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

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

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

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

A commonly-used method to introduce an index in a LaTeX document:

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

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

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

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

Pkg tocbibind

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

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

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

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

Opt tocbibind numindex
numbered index section

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

See section 65 for lwarp's core index and glossary code, and section 258 for tocbibind.

## 8.5 Math

### 8.5.1 Rendering tradeoffs

Math rendering

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

SVG files

In its current implementation, rendering math as images creates a new SVG file for each expression. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and check-summing may be used to remove the need for duplicate files.

SVG inline

Another approach could be to in-line the SVG files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files

Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML

Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

### 8.5.2 SVG option

SVG math option

For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[11], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size

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

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

SVG math copy/paste

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

### 8.5.3 MathJax option

MathJax math option

Prog `MathJax`

The popular MathJax alternative (`mathjax.org`) may be used to display math.

When MathJax is enabled, math is rendered twice:

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

[11] See section 273 regarding fonts and fractions.

2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

MathJax limitations
Prog MathJax

Limitations when using MathJax include:

chapter numbers

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

  `https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY`

subequations

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

footnotes in math

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

lateximage

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

siunitx

⚠ siunitx inside an equation

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

  `https://github.com/burnpanck/MathJax-siunitx`

  Also see section 8.5.5.

⚠ other macros and packages

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

custom MathJax macros

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

```
\begin{document}
(…)
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
    % New macros for MathJax are
    % placed inside a math expression:
    \(
        \newcommand{\expval}[1]{\langle#1\rangle}
        \newcommand{\abs}[1]{\lvert#1\rvert}
    \)
}{}
\end{warpHTML}
```

### 8.5.4 ntheorem package

⚠ Font control

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

⚠ Equation numbering

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

### 8.5.5 siunitx package

Pkg `siunitx`

⚠ per-mode

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

⚠ math mode required

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

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

### 8.5.6 units and nicefrac packages

Pkg `units`

Pkg `nicefrac`

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

### 8.5.7 newtxmath package

Pkg newtxmath

⚠ loading sequence

The proper load order is:

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

## 8.6 Graphics

⚠ .pdf image files

⚠ no file extension

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

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

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

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

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

other image files

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

⚠ graphics vs. graphicx

⚠ viewports

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

units

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

options

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

HTML class

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

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

⚠ browser support `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

### 8.6.1 grffile package

grffile is supported as-is. File types known to the browser are displayed, and unknown

⚠ matching PDF and SVG file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

### 8.6.2 color package

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

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

### 8.6.3 xcolor package

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

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

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

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

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

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

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

### 8.6.4 overpic package

⚠ scaling

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

## 8.7 Tabular

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

**Defining environments:**

⚠ misplaced alignment alignment tab character &

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

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

⚠ tabular inside another environment

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

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

**Cell contents:**

⚠ paragraphs

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

⚠ \multirow

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

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

vposn

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

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 192.2 for \multicolumrow.

⚠ \multicolumn & \multirow

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

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

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

⚠ skipped cells

⚠ empty cells

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

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

vposn

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

⚠ macro in a table
custom macros

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

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

**Column specifiers:**

⚠ * column specification

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

@ and !

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

\multirow

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

⚠ \newcolumntype

- \newcolumntype is ignored; unknown column types are set to l.

**Rules:**

vertical rules

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

width and trim

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

full-width rules

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

combined rules

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

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

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

⚠ \warpprintonly
misplaced \noalign

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a "misplaced \noalign" error occurs, change

  ```
  This & That \endhead
  ```

  to

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

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

**Other:**

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

longtable headings

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

⚠ S columns

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

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

### 8.7.1 longtable package

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

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

instead of

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

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

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

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

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

or place it inside \warpingprintonly.

### 8.7.2 supertabular and xtab packages

⚠ misplaced alignment alignment tab character &

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

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

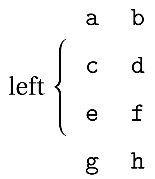
See section 8.7.

### 8.7.3 bigdelim package

⚠ use \mrowcell

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

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



## 8.8 Floats

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

⚠ package conflicts If using `\newfloat`, trivfloat, and/or algorithmicx together, see section 264.1.

### 8.8.2 caption and subcaption packages

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

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

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

### 8.8.3 subfig package

⚠ `lof/lotdepth` At present, the package options for `lofdepth` and `lotdepth` are not working. These counters must be set separately after the package has been loaded.

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

### 8.8.4 floatrow package

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

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

⚠ `\FBwidth`, `\FBheight`

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

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

### 8.8.5 keyfloat package

⚠ `keywrap`

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

## 8.9 Koma-Script

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

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

⚠ Not yet tested!

Please send bug reports!

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

## 8.10 Miscellaneous

### 8.10.1 verse and memoir

`\attrib`

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

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

Len \leftskip
Len \leftmargini
Len \TMLvleftskip
Len \TMLleftmargini

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

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

### 8.10.2 newclude package

Pkg newclude
⚠ loading

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

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

### 8.10.3 babel package

Pkg babel

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

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

punctuation spaces

Also when French is used, lwarp creates fixed-width space around punctuation by patching \FBcolonspace, \FBthinspace, \FBguillspace, \FBmedkern, \FBthickkern, \FBtextellipsis, and the tilde. If the user's document also changes these parameters, the user's changes should be placed inside a warpprint environment so that the user's changes do not affect the HTML output.

⚠ customized spacing

### 8.10.4 todonotes and luatodonotes packages

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

### 8.10.5 fixme

⚠ external layouts

External layouts (\fxloadlayouts) are not supported.

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

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

# 9 EPUB conversion

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

<meta> author

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

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

*FormatEPUB*

Bool FormatEPUB
Default: `false`

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

Enter ⇒ `lwarpmk cleanall`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk limages`

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

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

Calibre

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

⚠ search order

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

Preferences ⟶ Plugins ⟶ File type plugins ⟶ HTML to Zip

Check the box Add linked files in breadth first order.

⚠ section breaks

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

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

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

> Match HTML tags with tag name:

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

> //h:h4 — or — //h:h3

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

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

*MathJax support*

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

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

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

# 10 Word-processor conversion

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

## 10.1 Activating word-processor conversion

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

*FormatWP*

Bool FormatWP
Default: false

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

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

```
\booltrue{FormatWP}
```

formatting adjustments

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

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

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

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

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

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

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

  **`WPMarkMath`:** Prints LaTeX math instead of using images.

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

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

## 10.2 Additional modifications

*WPMarkFloats*

Bool WPMarkFloats
Default: false

Adds

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

or

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

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

*WPMarkMinipages*

Bool WPMarkMinipages
Default: false

Adds

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

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

*WPMarkTOC*

Bool WPMarkTOC
Default: true

While formatting for word processors, adds

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

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

*If set `false`, the actual TOC is printed instead.*

*WPMarkLOFT*

Bool WPMarkLOFT
Default: false

While formatting for word processors, adds

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

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

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

*WPMarkMath*

Bool WPMarkMath
Default: false
Prog TeXMaths
siunitx

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

When using the siunitx package, enter

```
\usepackage{siunitx}
```

in the TeXMaths preamble. Equation numbering is problematic for AMS math environments.

*WPTitleHeading*

Bool WPTitleHeading
Default: false
section headings

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

See table 6 on page 116.

## 10.3 Recommendations

TOC, LOF, LOT

For use with LibreOffice Writer, it is recommended to:

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

Table 6: Section HTML headings for word-processor conversion

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

* For default depths when not `FormatWP`, see table 7 on page 125.

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

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

siunitx

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

LO view border options

LibreOffice has options in the View menu to turn on/off the display of thin borders around table cells and text objects.

## 10.4 Limitations

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

Cross references are not actual LibreOffice linked cross references.

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

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

As of when last checked, LibreOffice ignores the following:

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

Libreoffice also has limitations with frames and backgrounds:

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

# 11 Modifying lwarp

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

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

An environment may be converted to a `lateximage` then displayed with an image of the resulting LaTeX output. See section 75 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 42.8.

## 11.1 Creating an lwarp version of a package

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

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

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

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If Dropped, the original print-version package is ignored, and only the lwarp- version is used. Use this where the original print version is useless for HTML. If Passed, the original package is loaded first, with the user-supplied options, then the lwarp- version continues loading as well. See section 201 (ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a lwarp- version.

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

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

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

to lwarp.dtx in section 22. This causes lwarp to stop with an error if packagename is loaded before lwarp.

## 11.3 Testing lwarp

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

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

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

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

## 11.4 Modifying lwarpmk

Prog lwarpmk
File lwarpmk.lua

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

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

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

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

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

# 12 Troubleshooting

## 12.1 Using the `lwarp.sty` package

Also see:
Section 7.7: Commands to be placed into the `warpprint` environment
Section 8: Special cases and limitations

**Text is not converting:**

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

**Undefined HTML settings:**

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

**Tabular problems:** See section 8.7.

**Obscure error messages:**

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

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

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

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

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

inside a group first.

**Missing sections:**

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

**Missing HTML files:**

- See the warning regarding changes to the HTML settings at section 7.3.

**Missing / incorrect cross-references:**

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

labels

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

⚠ underscores

\nameref

⚠ empty link

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

⚠ cleveref page numbers

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

  Ex:

  ```
      \cpageref{tab:first,tab:second}
   in HTML becomes:
  ```

  "pages **for** table 4.1 and **for** table 4.2"

  See `\cpagerefFor` at page 402 to redefine the message which is printed for page number references.

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

Use XƎLaTeX or LuaLaTeX.

**Floats out of sequence:**

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

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

**Print document contains HTML tags:**

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

**Images are appearing in strange places:**

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

**SVG images:**

⚠ Adding/removing

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

⚠ HTML instead of images

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

⚠ Lots of files!

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

**Plain-looking document:**

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

**Broken fragments of HTML:**

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

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

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

**Un-matched conditional compiles:**

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

### 12.1.1 Debug tracing output

`\tracinglwarp`

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

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

## 12.2 Compiling the `lwarp.dtx` file

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

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

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

**Un-nested environments:**

Be sure to properly nest:

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

# File 1 lwarp.sty

## 13 Implementation

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

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

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

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

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

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

Table 7: Section depths and HTML headings

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

* If `FormatWP` is true, section headings may be adjusted, depending on `WPTitleHeading`. See table 6 on page 116.

# 14 Section depths and HTML headings

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

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

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

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

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

# 15 Source Code

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

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

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

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

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

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

⚠ warnings Special warnings are marked with a warning icon.

**for HTML output:**
**for PRINT output:**
**for HTML & PRINT:**

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

— lwarp source code begins on the following page —

# 16 Detecting the TeX Engine — pdflatex, lualatex, xelatex

```
\RequirePackage{iftex}

\ifLuaTeX
\RequirePackage{luatex85}% until the geometry package is updated
\fi
```

# 17 pdfLaTeX T1 and UTF8 encoding

When using pdfLaTeX, lwarp required T1 and UTF8 encoding.

XeLaTeX and LuaLaTeX are both UTF8 by nature.

```
\ifPDFTeX
\RequirePackage[T1]{fontenc}
\RequirePackage[utf8]{inputenc}
\fi
```

# 18 Unicode input characters

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

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

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

```

\RequirePackage{newunicodechar}

\newunicodechar{×}{\texttimes}

\ifPDFTeX
\newunicodechar{ff}{ff}% the first arguments are ligatures
\newunicodechar{fi}{fi}
\newunicodechar{fl}{fl}
\newunicodechar{ffi}{ffi}
\newunicodechar{ffl}{ffl}
\newunicodechar{--}{---}
\newunicodechar{-}{--}
```

In PDFTEX, preserve upright quotes in verbatim text:

```
\RequirePackage{upquote}
\else
\fi
```

# 19 Early package requirements

Pkg etoolbox Provides \ifbool and other functions.

Pkg xpatch Patches macros with optional arguments.

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

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

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

Pkg letltxmacro Used to redefine \textbf and friends.

```
\RequirePackage{letltxmacro}
```

# 20 Operating-System portability

Prog Unix
Prog Mac OS
Prog Linux

lwarp tries to detect which operating system is being used. Unix / Mac OS / Linux is the default (collectively referred to as "Unix" in the configuration files), and MS-Windows is supported as well.

Prog MS-Windows

If Windows is not correctly detected, use the lwarp option `OSWindows`.

Prog Windows
Opt OSWindows

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

## 20.1 Common portability code

Bool usingOSWindows

Set if the `OSWindows` option is used.

```
\newbool{usingOSWindows}
\boolfalse{usingOSWindows}
```

## 20.2 Unix, Linux, and Mac OS

\OSPathSymbol

Symbol used to separate directories in a path.

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

## 20.3 MS-Windows

For MS-Windows:

\LWR@setOSWindows

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

```
\newcommand*{\LWR@setOSWindows}
{
\booltrue{usingOSWindows}
\renewcommand*{\OSPathSymbol}{\@backslashchar}
}
```

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

```
\ifwindows
\LWR@setOSWindows
\fi
```

# 21 Package options

Pkg `kvoptions` Allows key/value package options.

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

Bool `warpingprint`

Bool `warpingHTML`

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

```
\newbool{warpingprint}
\newbool{warpingHTML}
\newbool{mathjax}
```

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

```
\booltrue{warpingprint}%
\boolfalse{warpingHTML}%
\boolfalse{mathjax}%
```

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

```
\DeclareVoidOption{warpprint}{%
\PackageInfo{lwarp}{Using option 'warpprint'}
\booltrue{warpingprint}%
\boolfalse{warpingHTML}%
}
```

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

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

```
\DeclareVoidOption{warpHTML}{%
\PackageInfo{lwarp}{Using option 'warpHTML'}%
\booltrue{warpingHTML}%
\boolfalse{warpingprint}%
}
```

Opt mathsvg

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

```
\DeclareVoidOption{mathsvg}{%
\PackageInfo{lwarp}{Using option 'mathsvg'}
\boolfalse{mathjax}%
}
```

Opt mathjax

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

```
\DeclareVoidOption{mathjax}{%
\PackageInfo{lwarp}{Using option 'mathjax'}
\booltrue{mathjax}%
}
```

Opt BaseJobname

Option BaseJobname sets the \BaseJobname for this document.

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

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

Opt IndexLanguage

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

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

Opt xdyFilename

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

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

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

Opt lwarpmk

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

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

```
\newbool{LWR@creatinglwarpmk}
\boolfalse{LWR@creatinglwarpmk}

\DeclareVoidOption{lwarpmk}{
\PackageInfo{lwarp}{Using option ’lwarpmk’}
\booltrue{LWR@creatinglwarpmk}
}
```

Opt OSWindows

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

```
\DeclareVoidOption{OSWindows}{
\PackageInfo{lwarp}{Using option ’OSWindows’}
\LWR@setOSWindows
}
```

Opt HomeHTMLFilename
Default: `\lwarp`

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

```
\DeclareStringOption[]{HomeHTMLFilename}
```

Opt HTMLFilename
Default: `<empty>`

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

```
\DeclareStringOption[]{HTMLFilename}
```

Opt latexmk

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

```
\DeclareBoolOption[false]{latexmk}
```

Execute options

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

```
\ProcessKeyvalOptions*\relax
```

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

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

Defaults unless already over-ridden by the user:

```
\ifcsempty{LWR@HomeHTMLFilename}{
\newcommand*{\HomeHTMLFilename}{\BaseJobname}
}{
\csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
}

\csedef{HTMLFilename}{\LWR@HTMLFilename}
```

## 21.1 Conditional compilation

\warpprintonly {⟨*contents*⟩}

Only process the contents if producing printed output.

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

\warpHTMLonly {⟨*contents*⟩}

Only process the contents if producing HTML output.

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

Pkg comment Provides conditional code blocks.

```
\RequirePackage{comment}
```

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

```
\ifbool{warpingHTML}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
}{}

\ifbool{warpingprint}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
}{}
```

```
\excludecomment{testing}
```

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

```
\includecomment{warpall}
```

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

For HTML output:

```
\ifbool{warpingHTML}{%
\includecomment{warpHTML}
}
{\excludecomment{warpHTML}}%
```

```
\ifbool{warpingprint}
{\includecomment{warpprint}}
{\excludecomment{warpprint}}
```

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

```
\ifbool{LWR@creatinglwarpmk}
{\includecomment{LWR@createlwarpmk}}
{\excludecomment{LWR@createlwarpmk}}
```

## 22 Misplaced packages

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

Packages which should only be loaded before lwarp have their own

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

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

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

```
\newcommand*{\LWR@loadafter}[1]{%
\@ifpackageloaded{#1}
{
\PackageError{lwarp}
{Package #1, or one which uses #1, must be loaded after lwarp}
{Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
Package #1 may also be loaded by something else, which must also be moved
after lwarp.}
}
{}
}
```

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

```
\newcommand*{\LWR@loadbefore}[1]{%
\@ifpackageloaded{#1}
{}
{
\PackageError{lwarp}
{Package #1 must be loaded before lwarp}
{Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
}
}
```

\LWR@loadnever {⟨*badpackagename*⟩} {⟨*replacementpkgname*⟩}

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

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

Packages which should only be loaded after lwarp are tested here to trip an error of they have already been loaded.

The following packages must be loaded after lwarp:

```
\LWR@loadafter{a4}
\LWR@loadafter{a4wide}
\LWR@loadafter{a5comb}
\LWR@loadafter{abstract}
\LWR@loadafter{adjmulticol}
\LWR@loadafter{addlines}
\LWR@loadafter{afterpage}
\LWR@loadafter{algorithmicx}
\LWR@loadafter{alltt}
\LWR@loadafter{amsmath}
\LWR@loadafter{amsthm}
\LWR@loadafter{anonchap}
\LWR@loadafter{anysize}
\LWR@loadafter{appendix}
\LWR@loadafter{arabicfront}
\LWR@loadafter{array}
\LWR@loadafter{atbegshi}
\LWR@loadafter{authblk}
\LWR@loadafter{balance}
\LWR@loadafter{bigdelim}
```

```
\LWR@loadafter{bigstrut}
\LWR@loadafter{bookmark}
\LWR@loadafter{booktabs}
\LWR@loadafter{boxedminipage}
\LWR@loadafter{boxedminipage2e}
\LWR@loadafter{breakurl}
\LWR@loadafter{cancel}
\LWR@loadafter{ccaption}
\LWR@loadafter{changepage}
\LWR@loadafter{chngpage}
\LWR@loadafter{chappg}
\LWR@loadafter{color}
\LWR@loadafter{crop}
\LWR@loadafter{cuted}
\LWR@loadafter{cutwin}
\LWR@loadafter{dblfnote}
\LWR@loadafter{dcolumn}
\LWR@loadafter{draftwatermark}
\LWR@loadafter{easy-todo}
\LWR@loadafter{ebook}
\LWR@loadafter{ellipsis}
\LWR@loadafter{emptypage}
\LWR@loadafter{enumerate}
\LWR@loadafter{enumitem}
\LWR@loadafter{epigraph}
\LWR@loadafter{eso-pic}
\LWR@loadafter{everypage}
\LWR@loadafter{everyshi}
\LWR@loadafter{extramarks}
\LWR@loadafter{fancybox}
\LWR@loadafter{fancyhdr}
\LWR@loadafter{fancyref}
\LWR@loadafter{fancyvrb}
\LWR@loadafter{figcaps}
\LWR@loadafter{fix2col}
\LWR@loadafter{fixme}
\LWR@loadafter{fixmetodonotes}
\LWR@loadafter{float}
\LWR@loadafter{floatflt}
\LWR@loadafter{floatpag}
\LWR@loadafter{floatrow}
\LWR@loadafter{flushend}
\LWR@loadafter{fncychap}
\LWR@loadafter{fnpos}
% fontenc must be loaded before lwarp
% fontspec must be loaded before lwarp
\LWR@loadafter{ftnright}
\LWR@loadafter{fullpage}
\LWR@loadafter{fullwidth}
\LWR@loadafter{geometry}
```

```
\LWR@loadafter{glossaries}
% \LWR@loadafter{graphics}% pre-loaded by xunicode
% \LWR@loadafter{graphicx}% pre-loaded by xunicode
\LWR@loadafter{grffile}
\LWR@loadafter{hang}
\LWR@loadafter{hyperref}
\LWR@loadafter{hyperxmp}
\LWR@loadafter{idxlayout}
\LWR@loadafter{ifoddpage}
\LWR@loadafter{indentfirst}
% inputenc must be loaded before lwarp
\LWR@loadafter{keyfloat}
\LWR@loadafter{layout}
\LWR@loadafter{letterspace}
\LWR@loadafter{lettrine}
\LWR@loadafter{lips}
\LWR@loadafter{listings}
\LWR@loadafter{longtable}
\LWR@loadafter{lscape}
\LWR@loadafter{ltcaption}
\LWR@loadafter{ltxtable}
\LWR@loadafter{luatodonotes}
\LWR@loadafter{marginfit}
\LWR@loadafter{marginfix}
\LWR@loadafter{marginnote}
\LWR@loadafter{mcaption}
\LWR@loadafter{mdframed}
\LWR@loadafter{metalogo}
\LWR@loadafter{microtype}
\LWR@loadafter{midfloat}
\LWR@loadafter{moreverb}
% morewrites must be loaded before lwarp
\LWR@loadafter{mparhack}
%\LWR@loadafter{multicol}% loaded by ltxdoc
\LWR@loadafter{multirow}
\LWR@loadafter{nameref}
\LWR@loadafter{needspace}
% newclude must be loaded before lwarp
\LWR@loadafter{newtxmath}
% newunicodechar must be loaded before lwarp
\LWR@loadafter{nextpage}
\LWR@loadafter{nonumonpart}
\LWR@loadafter{nopageno}
\LWR@loadafter{nowidow}
\LWR@loadafter{ntheorem}
\LWR@loadafter{overpic}
\LWR@loadafter{pagenote}
\LWR@loadafter{paralist}
\LWR@loadafter{parskip}
\LWR@loadafter{pdflscape}
```

```
\LWR@loadafter{pdfsync}
\LWR@loadafter{pfnote}
\LWR@loadafter{placeins}
\LWR@loadafter{prelim2e}
\LWR@loadafter{preview}
\LWR@loadafter{quotchap}
\LWR@loadafter{ragged2e}
\LWR@loadafter{realscripts}
\LWR@loadafter{relsize}
\LWR@loadafter{romanbar}
\LWR@loadafter{romanbarpagenumber}
\LWR@loadafter{rotating}
\LWR@loadafter{rotfloat}
\LWR@loadafter{savetrees}
% \LWR@loadafter{scalefnt}% loaded by babel-french
\LWR@loadafter{scrextend}
\LWR@loadafter{scrhack}
\LWR@loadafter{scrlayer}
\LWR@loadafter{scrlayer-notecolumn}
\LWR@loadafter{scrlayer-scrpage}
\LWR@loadafter{section}
\LWR@loadafter{sectsty}
\LWR@loadafter{setspace}
\LWR@loadafter{shadow}
\LWR@loadafter{showidx}
\LWR@loadafter{showkeys}
\LWR@loadafter{sidecap}
\LWR@loadafter{sidenotes}
\LWR@loadafter{siunitx}
\LWR@loadafter{soul}
\LWR@loadafter{soulpos}
\LWR@loadafter{soulutf8}
\LWR@loadafter{stabular}
\LWR@loadafter{subfig}
\LWR@loadafter{supertabular}
\LWR@loadafter{tabls}
\LWR@loadafter{tabularx}
\LWR@loadafter{tabulary}
\LWR@loadafter{textarea}
% \LWR@loadafter{textcomp}% maybe before lwarp with font packages
\LWR@loadafter{textpos}
\LWR@loadafter{theorem}
\LWR@loadafter{threeparttable}
\LWR@loadafter{tikz}
\LWR@loadafter{titleps}
\LWR@loadafter{titlesec}
\LWR@loadafter{titletoc}
\LWR@loadafter{titling}
% \LWR@loadafter{tocbasic}% preloaded by koma-script classes
\LWR@loadafter{tocbibind}
```

```
\LWR@loadafter{tocloft}
\LWR@loadafter{tocstyle}
\LWR@loadafter{todo}
\LWR@loadafter{todonotes}
\LWR@loadafter{transparent}
\LWR@loadafter{trivfloat}
```

```
% \LWR@loadafter{typearea}% preloaded by koma-script classes
\LWR@loadafter{ulem}
\LWR@loadafter{upref}
\LWR@loadafter{varioref}
\LWR@loadafter{verse}
\LWR@loadafter{wallpaper}
\LWR@loadafter{watermark}
\LWR@loadafter{wrapfig}
\LWR@loadafter{xcolor}
\LWR@loadafter{xfrac}
\LWR@loadafter{xltxtra}
\LWR@loadafter{xmpincl}
\LWR@loadafter{xtab}
\LWR@loadafter{zwpagelayout}
```

## 23 Required packages

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

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

**for HTML & PRINT:**

```
\begin{warpall}
```

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

Detects XeTeX and LuaLaTeX:

```
\RequirePackage{iftex}
\newif\ifxetexorluatex
\ifXeTeX
    \xetexorluatextrue
\else
    \ifLuaTeX
        \xetexorluatextrue
    \else
        \xetexorluatexfalse
    \fi
\fi
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

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

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

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
\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 LaTeX dash and quote ligatures, which may fail on older browers but at least won't corrupt written words.

```
\RequirePackage {microtype}

\microtypesetup{
    protrusion=false,
    expansion=false,
    tracking=false,
    kerning=false,
    spacing=false}

\DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
```

```
\fi
```

```
\end{warpHTML}
```

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

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

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

**for HTML output:**

```
\begin{warpHTML}
\RequirePackage[paperheight=190in,paperwidth=20in,%
left=2in,right=12in,%
top=1in,bottom=1in,%
]{geometry}
\@twosidefalse
\@mparswitchfalse
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

Pkg `xparse`

LaTeX3 command argument parsing

```
\RequirePackage{xparse}
```

Pkg `afterpackage`

Used to patch titling to add `\AddSubtitlePublished`. Provided by scrlfile or afterpackage.

```
\@ifundefined{AfterPackage}%
{
\RequirePackage{afterpackage}
}{}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

Pkg `expl3`

LaTeX3 programming

```
\RequirePackage{expl3}
```

Pkg `gettitlestring`

Used to emulate `\nameref`.

```
\RequirePackage{gettitlestring}
```

Pkg `everyhook`

everyhook is used to patch paragraph handling.

```
\RequirePackage{everyhook}
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

Pkg filecontents

Used to write helper files, done in print mode.

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

```
\RequirePackage{filecontents}

\@ifpackagelater{filecontents}{2011/10/09}%
{}
{
\newwrite\fcwrite
\let\LWR@origfilec@ntents\filec@ntents
\def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

Pkg xifthen

```
\RequirePackage{xifthen}
```

Pkg xstring

```
\RequirePackage{xstring}
```

Pkg xstring

```
\RequirePackage{verbatim}
```

Pkg makeidx

```
\RequirePackage{makeidx}
\makeindex
```

Pkg calc

```
\RequirePackage{calc}
```

Pkg refcount

```
\RequirePackage{refcount}
```

Pkg newfloat

```
\RequirePackage{newfloat}
```

Pkg caption

```
\RequirePackage{caption}
```

```
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

Pkg environ

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

```
\RequirePackage{environ}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

Pkg zref

Used for cross-references.

```
\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.

```
\PassOptionsToPackage{leqno}{amsmath}
\RequirePackage{amsmath}
```

Used to convert lengths for image width/height options.

```
\RequirePackage{printlen}
```

```
\end{warpHTML}
```

# 24 Loading packages

**for HTML output:**

```
\begin{warpHTML}
```

Remember the original \RequirePackage:

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

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

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

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

Note that argument 4 is passed directly to \StrBetween.

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

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

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

Find the n'th package name from the list:

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

Remove blanks. The original name with blanks is in LWR@strresult and the final name with no blanks goes into LWR@strresulttwo.

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

See if the package name was found:

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

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

```
\IfFileExists{lwarp-\LWR@strresulttwo.sty}%
{% lwarp-* file found
        \StrSubstitute%
```

```
            {\LWR@requirepackagenames}%
            {\LWR@strresult}%
            {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
}%
{}% no lwarp-* file
}% yes filename
}
```

\RequirePackage [⟨*1: options*⟩] {⟨*2: package names*⟩} [⟨*3: version*⟩]

For each of many package names in a comma-separated list, if an lwarp version of a package exists, select it instead of the LaTeX version.

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

Redirect up to nine names:

```
\renewcommand*{\LWR@requirepackagenames}{#2}
\LWR@lookforpackagename{1}
\LWR@lookforpackagename{2}
\LWR@lookforpackagename{3}
\LWR@lookforpackagename{4}
\LWR@lookforpackagename{5}
\LWR@lookforpackagename{6}
\LWR@lookforpackagename{7}
\LWR@lookforpackagename{8}
\LWR@lookforpackagename{9}
```

\RequirePackage depending on the options and version:

```
\IfValueTF{#1}
{% options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
    {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
}
{% no options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
    {\LWR@origRequirePackage{\LWR@requirepackagenames}}
}
}
\LetLtxMacro{\usepackage}{\RequirePackage}
```

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

Uses the original package, including options.

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

\IfValueTF{#2}
{\LWR@origRequirePackage{#1}[#2]}
{\LWR@origRequirePackage{#1}}
}
```

`\LWR@ProvidesPackageDrop` {⟨*pkgname*⟩} [⟨*version*⟩]

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

```
\NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
\PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
\IfValueTF{#2}
{\ProvidesPackage{lwarp-#1}[#2]}
{\ProvidesPackage{lwarp-#1}}
\DeclareOption*{}
\ProcessOptions\relax
}
```

```
\end{warpHTML}
```

## 25 File handles

Defines file handles for writes.

**for HTML & PRINT:**

```
\begin{warpall}
```

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

```
\newwrite\LWR@quickfile%
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

`\LWR@lateximagesfile` For `lateximages.txt`.

```
\newwrite\LWR@lateximagesfile
```

```
\end{warpHTML}
```

## 26 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. lwarp creates `<filename>_html_inc.tex` whose contents are:

   `\input <filename>.tex`

2. `<filename>_html_inc.tex` is then `\include`d instead of `<filename>.tex`.
3. `<filename>_html_inc.aux` is automatically generated and used by LaTeX.

**for HTML output:**

```
\begin{warpHTML}
```

`\include` {⟨*filename*⟩}

`\@include` {⟨*filename*⟩} Modified to load `_html_inc` files.

```
\def\@include#1 {%
\immediate\openout\LWR@quickfile #1_html_inc.tex% new
\immediate\write\LWR@quickfile{\string\input{#1.tex}}% new
\immediate\closeout\LWR@quickfile% new
\LWR@origclearpage% \changed
\if@filesw
    \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
\fi
\@tempswatrue
\if@partsw
    \@tempswafalse
    \edef\reserved@b{#1}%
    \@for\reserved@a:=\@partlist\do
    {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
\fi
\if@tempswa
    \let\@auxout\@partaux
    \if@filesw
        \immediate\openout\@partaux #1_html_inc.aux % changed
        \immediate\write\@partaux{\relax}%
    \fi
    \@input@{#1_html_inc.tex}% changed
    \LWR@origclearpage% changed
    \@writeckpt{#1}%
    \if@filesw
        \immediate\closeout\@partaux
    \fi
\else
    \deadcycles\z@
    \@nameuse{cp@#1}%
\fi
\let\@auxout\@mainaux%
}
```

```
\end{warpHTML}
```

## 27 Copying a file

**for HTML output:**

```
\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.

```
\newwrite\LWR@copyoutfile  % open the file to write to
\newread\LWR@copyinfile      % open the file to read from

\newcommand*{\LWR@copyfile}[2]{%
\immediate\openout\LWR@copyoutfile=#2
\openin\LWR@copyinfile=#1
\begingroup\endlinechar=-1
\makeatletter
\loop\unless\ifeof\LWR@copyinfile
  \read\LWR@copyinfile to\LWR@fileline  % Read one line and store it into \LWR@fileline
%   \LWR@fileline\par                   % print the content into the pdf
% print the content:
  \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
\repeat
\closeout\LWR@copyoutfile
\endgroup
}
```

```
\end{warpHTML}
```

# 28 Debugging messages

**for HTML & PRINT:**

```
\begin{warpall}
```

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

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

`\tracinglwarp` Turns on the debug tracing messages.

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

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

```
\newcommand{\LWR@traceinfo}[1]{%
\ifbool{LWR@tracinglwarp}%
{%
    \typeout{*** lwarp: #1}%
    % \PackageInfo{lwarp}{#1 : }%
}%
{}%
}
```

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

```
\newbool{HTMLDebugComments}
\boolfalse{HTMLDebugComments}
```

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

```
\AfterEndPreamble{
\LWR@traceinfo{AfterEndPreamble}
}

\AtBeginDocument{
\LWR@traceinfo{AtBeginDocument}
}
```

```
\end{warpall}
```

# 29 Koma-script

Load patches to koma-script.

**for HTML output:**

```
\begin{warpHTML}
```

```
\@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}
```

```
\end{warpHTML}
```

# 30 HTML-conversion output modifications

These booleans modify the HTML output in various ways to improve conversion to EPUB or word processor imports.

**for HTML & PRINT:**

```
\begin{warpall}
```

## 30.1 User-level controls

Bool `FormatEPUB`
Default: `false`

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

```
\newbool{FormatEPUB}
\boolfalse{FormatEPUB}
```

Bool `FormatWP`
Default: `false`

Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.

```
\newbool{FormatWP}
\boolfalse{FormatWP}
```

Bool `WPMarkFloats`
Default: `false`

Adds

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

or

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

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions. [12]

```
\newbool{WPMarkFloats}
\boolfalse{WPMarkFloats}
```

Bool WPMarkMinipages
Default: false

Adds

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

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

```
\newbool{WPMarkMinipages}
\boolfalse{WPMarkMinipages}
```

Bool WPMarkTOC
Default: true

While formatting for word processors, adds

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

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

*If set `false`, the actual TOC is printed instead.*

```
\newbool{WPMarkTOC}
\booltrue{WPMarkTOC}
```

Bool WPMarkLOFT
Default: false

While formatting for word processors, adds

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

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

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

```
\newbool{WPMarkLOFT}
\boolfalse{WPMarkLOFT}
```

Bool WPMarkMath
Default: false

While formatting for word processors, prints math as LaTeX code instead of creat-

[12] Perhaps some day word processors will have HTML import options for identifying `<figure>` and `<figcaption>` tags for figures and tables.

ing SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

```
\newbool{WPMarkMath}
\boolfalse{WPMarkMath}
```

Bool WPTitleHeading
Default: false

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

See table 6 on page 116.

```
\newbool{WPTitleHeading}
\boolfalse{WPTitleHeading}
```

```
\end{warpall}
```

## 30.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If WPTitleHeading is true, adjust so that part is **Heading 1**.

If WPTitleHeading is false, use <h1> for the title, and set part to **Heading 2**.

**for HTML output:**

```
\begin{warpHTML}
```

```
\AtBeginDocument{
\ifbool{FormatWP}{
\@ifundefined{chapter}{
\ifbool{WPTitleHeading}{% part and section starting at h2
\renewcommand*{\LWR@tagtitle}{h1}
\renewcommand*{\LWR@tagtitleend}{/h1}
\renewcommand*{\LWR@tagpart}{h2}
\renewcommand*{\LWR@tagpartend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{}}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% WPTitleHeading
{% not WPTitleHeading, part and section starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagsection}{h2}
\renewcommand*{\LWR@tagsectionend}{/h2}
\renewcommand*{\LWR@tagsubsection}{h3}
\renewcommand*{\LWR@tagsubsectionend}{/h3}
\renewcommand*{\LWR@tagsubsubsection}{h4}
\renewcommand*{\LWR@tagsubsubsectionend}{/h4}
\renewcommand*{\LWR@tagparagraph}{h5}
\renewcommand*{\LWR@tagparagraphend}{/h5}
\renewcommand*{\LWR@tagsubparagraph}{h6}
\renewcommand*{\LWR@tagsubparagraphend}{/h6}
}% not WPTitleHeading
}% chapter undefined
{% chapter defined
\ifbool{WPTitleHeading}{}
{% not WPTitleHeading, part and chapter starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagchapter}{h2}
\renewcommand*{\LWR@tagchapterend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{}}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% not WPTitleHeading
}% chapter defined
}{}% FormatWP
}% AtBeginDocument
```

```
\end{warpHTML}
```

## 31 Remembering original formatting macros

**for HTML output:**

```
\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.

```
\LetLtxMacro{\LWR@origtextrm}{\textrm}
\LetLtxMacro{\LWR@origtextsf}{\textsf}
\LetLtxMacro{\LWR@origtexttt}{\texttt}
\LetLtxMacro{\LWR@origtextnormal}{\textnormal}
\LetLtxMacro{\LWR@origtextbf}{\textbf}
\LetLtxMacro{\LWR@origtextmd}{\textmd}
\LetLtxMacro{\LWR@origtextit}{\textit}
\LetLtxMacro{\LWR@origtextsl}{\textsl}
\LetLtxMacro{\LWR@origtextsc}{\textsc}
\LetLtxMacro{\LWR@origtextup}{\textup}
\LetLtxMacro{\LWR@origemph}{\emph}

\LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
\LetLtxMacro{\LWR@origsffamily}{\sffamily}
\LetLtxMacro{\LWR@origttfamily}{\ttfamily}
\LetLtxMacro{\LWR@origbfseries}{\bfseries}
\LetLtxMacro{\LWR@origmdseries}{\mdseries}
\LetLtxMacro{\LWR@origupshape}{\upshape}
\LetLtxMacro{\LWR@origslshape}{\slshape}
\LetLtxMacro{\LWR@origscshape}{\scshape}
\LetLtxMacro{\LWR@origitshape}{\itshape}
\LetLtxMacro{\LWR@origem}{\em}
\LetLtxMacro{\LWR@orignormalfont}{\normalfont}

\let\LWR@origraggedright\raggedright
\let\LWR@origonecolumn\onecolumn

\let\LWR@origsp\sp
\let\LWR@origsb\sb
\LetLtxMacro\LWR@origtextsuperscript\textsuperscript
\LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript
\AtBeginDocument{
\LetLtxMacro\LWR@origtextsubscript\textsubscript
\LetLtxMacro\LWR@orig@textsubscript\@textsubscript
}
\LetLtxMacro\LWR@origunderline\underline

\let\LWR@origscriptsize\scriptsize

\let\LWR@orignewpage\newpage

\let\LWR@origpagestyle\pagestyle
\let\LWR@origthispagestyle\thispagestyle
\let\LWR@origpagenumbering\pagenumbering

```

```
\LetLtxMacro{\LWR@origminipage}{\minipage}
\let\LWR@origendminipage\endminipage
\LetLtxMacro{\LWR@origparbox}{\parbox}

\let\LWR@orignewline\newline

\LetLtxMacro\LWR@origitem\item


\AtBeginDocument{% in case packages change definition
\LetLtxMacro{\LWR@origitemize}{\itemize}
\LetLtxMacro{\LWR@endorigitemize}{\enditemize}
\LetLtxMacro{\LWR@origenumerate}{\enumerate}
\LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
\LetLtxMacro{\LWR@origdescription}{\description}
\LetLtxMacro{\LWR@endorigdescription}{\enddescription}
}

\let\LWR@origpar\par

\LetLtxMacro{\LWR@origfootnote}{\footnote}
\let\LWR@orig@mpfootnotetext\@mpfootnotetext

\let\LWR@origclearpage\clearpage
```

```
\end{warpHTML}
```

## 32 Accents

Native LaTeX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

**for HTML output:**

```
\begin{warpHTML}
```

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
\AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only `\t` needs to be redefined:

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

For pdfLaTeX, additional work is required:

```
\ifPDFTeX
\LetLtxMacro{\LWR@origequalaccent}{\=}
\LetLtxMacro{\LWR@origdotaccent}{\.}
\LetLtxMacro{\LWR@origu}{\u}
\LetLtxMacro{\LWR@origv}{\v}
\LetLtxMacro{\LWR@origc}{\c}
\LetLtxMacro{\LWR@origd}{\d}
\LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdfLaTeX, Unicode diacritical marks are used:

```
\renewcommand*{\=}[1]{#1\HTMLunicode{0305}}
\renewcommand*{\.}[1]{#1\HTMLunicode{0307}}
\renewcommand*{\u}[1]{#1\HTMLunicode{0306}}
\renewcommand*{\v}[1]{#1\HTMLunicode{030C}}
\renewcommand*{\c}[1]{#1\HTMLunicode{0327}}
\renewcommand*{\d}[1]{#1\HTMLunicode{0323}}
\renewcommand*{\b}[1]{#1\HTMLunicode{0331}}
\fi
```

For all engines, a Unicode diacritical tie is used:

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

\LWR@restoreorigaccents

Called from \restoreoriginalformatting when a lateximage is begun.

```
\ifPDFTeX
\newcommand*{\LWR@restoreorigaccents}{%
\LetLtxMacro{\=}{\LWR@origequalaccent}%
\LetLtxMacro{\.}{\LWR@origdotaccent}%
\LetLtxMacro{\u}{\LWR@origu}%
\LetLtxMacro{\v}{\LWR@origu}%
\LetLtxMacro{\t}{\LWR@origt}%
\LetLtxMacro{\c}{\LWR@origc}%
\LetLtxMacro{\d}{\LWR@origd}%
\LetLtxMacro{\b}{\LWR@origb}%
}
\else% XeLaTeX, LuaLaTeX:
\newcommand*{\LWR@restoreorigaccents}{%
\LetLtxMacro{\t}{\LWR@origt}%
}
\fi
}% AtBeginDocument
```

```
\end{warpHTML}
```

# 33 Configuration Files

```
\begin{warpprint}
\typeout{lwarp: generating configuration files}
\end{warpprint}
```

## 33.1 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.

```
\begin{warpprint}
\immediate\openout\LWR@quickfile=\jobname_html.tex
\immediate\write\LWR@quickfile{%
\detokenize{\PassOptionsToPackage}%
{warpHTML,BaseJobname=\jobname}{lwarp}%
}
\immediate\write\LWR@quickfile{%
\detokenize{\input}\string{\jobname.tex\string }%
}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

## 33.2 lwarpmk.conf

File lwarpmk.conf

lwarpmk.conf is automatically (re-)created by the lwarp package when executing
  pdflatex <project.tex>,
or similar for xelatex or lualatex, in print-document generation mode, which is the default unless the warpHTML option is given. lwarpmk.conf is then used by the utility lwarpmk.

An example lwarpmk.conf:

```
opsystem = "Unix"   -- or "Windows"
latexname = "pdflatex"   -- or "lualatex" or "xelatex"
sourcename = "projectname"   -- your .tex source
homehtmlfilename = "index"   -- or "projectname"
htmlfilename = ""   -- or "projectname" if numbered HTML files
```

**for PRINT output:**

```
\begin{warpprint}
\ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
\immediate\openout\LWR@quickfile=lwarpmk.conf
\ifbool{usingOSWindows}{
```

```
\immediate\write\LWR@quickfile{opsystem = "Windows"}
}{
\immediate\write\LWR@quickfile{opsystem = "Unix"}
}
\ifPDFTeX
\immediate\write\LWR@quickfile{latexname = "pdflatex"}
\fi
\ifXeTeX
\immediate\write\LWR@quickfile{latexname = "xelatex"}
\fi
\ifLuaTeX
\immediate\write\LWR@quickfile{latexname = "lualatex"}
\fi
\immediate\write\LWR@quickfile{sourcename = "\jobname"}
\immediate\write\LWR@quickfile{%
homehtmlfilename = "\HomeHTMLFilename"%
}
\immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
\immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
\immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
\immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

## 33.3 project.lwarpmkconf

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

```
\begin{warpprint}
\ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
\immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
\ifbool{usingOSWindows}{
\immediate\write\LWR@quickfile{opsystem = "Windows"}
}{
\immediate\write\LWR@quickfile{opsystem = "Unix"}
}
\ifPDFTeX
\immediate\write\LWR@quickfile{latexname = "pdflatex"}
\fi
\ifXeTeX
\immediate\write\LWR@quickfile{latexname = "xelatex"}
\fi
\ifLuaTeX
\immediate\write\LWR@quickfile{latexname = "lualatex"}
\fi
\immediate\write\LWR@quickfile{sourcename = "\jobname"}
\immediate\write\LWR@quickfile{%
homehtmlfilename = "\HomeHTMLFilename"%
}
\immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
\immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
\immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
\immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

## 33.4 lwarp.css

File lwarp.css This is the base CSS layer used by lwarp.

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

```
\begin{warpprint}
\begin{filecontents*}{lwarp.css}
/*
 CSS stylesheet for the LaTeX lwarp package
 Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
*/


/* a fix for older browsers: */
header, section, footer, aside, nav, main,
    article, figure { display: block; }


A:link {color:#000080 ; text-decoration: none ; }
A:visited {color:#800000 ; }
A:hover {color:#000080 ; text-decoration: underline ;}
A:active {color:#800000 ; }

a.tocpart {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
a.tocchapter {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
a.tocsection {display: inline-block ; margin-left: 1em ;
    text-indent: -.5em ; font-weight: bold ; }
a.tocsubsection {display: inline-block ; margin-left: 2em ;
    text-indent: -.5em ; }
a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
    text-indent: -.5em ; }
a.tocparagraph {display: inline-block ; margin-left: 4em ;
    text-indent: -.5em ; }
a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
```

```
    text-indent: -.5em ; }
a.tocfigure {margin-left: 0em}
a.tocsubfigure {margin-left: 2em}
a.toctable {margin-left: 0em}
a.tocsubtable {margin-left: 2em}
a.toctheorem {margin-left: 0em}
a.toclstlisting {margin-left: 0em}

body {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
        "Lucida Bright", Georgia, serif;
    background: #FAF7F4 ;
    color: black ;
    margin:0em ;
    padding:0em ;
    font-size: 100% ;
    line-height: 1.2 ;
}

p {margin: 1.5ex 0em 1.5ex 0em ;}

/* Holds a section number to add space between it and the name */
span.sectionnumber { margin-right: 0em }

/* Inserted in front of index lines */
span.indexitem {margin-left: 0em}
span.indexsubitem {margin-left: 2em}
span.indexsubsubitem {margin-left: 4em}

div.hidden, span.hidden { display: none ; }

kbd {
    font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
    font-size: 100% ;
}

pre { padding: 3pt ; }

span.strong { font-weight: bold; }

span.textmd { font-weight: normal; }

span.textsc { font-variant: small-caps; }

span.textsl { font-style: oblique; }

span.textup { font-variant: normal; }

span.textrm {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
    "Lucida Bright", Georgia, serif;
}

span.textsf {
     font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
}

span.textcircled { border: 1px solid black ; border-radius: 1ex ; }

span.underline {
    text-decoration: underline ;
    text-decoration-skip ;
}


/* For realscripts */
.supsubscript {
    display: inline-block;
    text-align:left ;
}

.supsubscript sup,
.supsubscript sub {
    position: relative;
    display: block;
    font-size: .5em;
    line-height: 1;
}

.supsubscript sup {
    top: .5em;
}

.supsubscript sub {
    top: .5em;
}

span.attribution {
  margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
}

span.citetitle {
  margin-left: 1em ; font-size: 80% ; font-style: oblique;
}

span.poemtitle {
  font-size: 120% ; font-weight: bold;
```

```
}

pre.tabbing {
    font-family: "Linux Libertine Mono O", "Lucida Console",
        "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
        "Liberation Mono", "FreeMono", "Andale Mono",
        "Nimbus Mono L", "Courier New", monospace;
}

blockquote {
    margin-left: 0px ;
    margin-right: 0px ;
}

/* quotchap is for the quotchap package */
div.quotchap {
  font-style: oblique ;
  overflow-x: auto ;
  margin-left: 2em ;
  margin-right: 2em ;
}

blockquote p, div.quotchap p {
  line-height: 1.5;
    text-align: left ;
    font-size: .85em ;
    margin-left: 3em ;
    margin-right: 3em ;
}

/* qauthor is for the quotchap package */
div.qauthor {
  display: block ;
  text-align: right ;
  margin-left: auto ;
  margin-right: 2em ;
  font-size: 80% ;
  font-variant: small-caps;
}

div.qauthor p {
  text-align: right ;
}

blockquotation {
    margin-left: 0px ;
    margin-right: 0px ;
}

blockquotation p {
  line-height: 1.5;
    text-align: left ;
    font-size: .85em ;
    margin-left: 3em ;
    margin-right: 3em ;
}

div.epigraph, div.dictum {
  line-height: 1.2;
    text-align: left ;
    padding: 3ex 1em 0ex 1em ;
/*      margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
    margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
/*    margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
    font-size: .85em ;
    max-width: 27em ;
}



div.epigraphsource, div.dictumauthor {
    text-align:right ;
    margin-left:auto ;
/*      max-width: 50% ; */
    border-top: 1px solid #A0A0A0 ;
    padding-bottom: 3ex ;
    line-height: 1.2;
}

div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
div.dictumauthor { font-style:italic }


/* lettrine package: */
span.lettrine { font-size: 3ex ; float: left ; }
span.lettrinetext { font-variant: small-caps ; }

/* ulem and soul packages: */
span.uline {
    text-decoration: underline ;
    text-decoration-skip ;
}

span.uuline {
    text-decoration: underline ;
    text-decoration-skip ;
    text-decoration-style: double ;
}

```

```
span.uwave {
    text-decoration: underline ;
    text-decoration-skip ;
    text-decoration-style: wavy ;
}

span.sout {
    text-decoration: line-through ;
}

span.xout {
    text-decoration: line-through ;
}

span.dashuline {
    text-decoration: underline ;
    text-decoration-skip ;
    text-decoration-style: dashed ;
}

span.dotuline {
    text-decoration: underline ;
    text-decoration-skip ;
    text-decoration-style: dotted ;
}

span.letterspacing { letter-spacing: .2ex ; }

span.capsspacing {
    font-variant: small-caps ;
    letter-spacing: .1ex ;
}

span.highlight { background: #F8E800 ; }




html body {
  margin: 0 ;
  line-height: 1.2;
}


body div {
  margin: 0ex;
}


h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
```

```
{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    font-style: normal ;
    font-weight: bold ;
    text-align: left ;
}

h1 {    /* title of the entire website, used on each page */
    text-align: center ;
    font-size: 2.5em ;
    padding: .4ex 0em 0ex 0em ;
}
h2 { font-size: 2.25em }
h3 { font-size: 2em }
h4 { font-size: 1.75em }
h5 { font-size: 1.5em }
h6 { font-size: 1.25em }
span.paragraph {font-size: 1em ; font-variant: normal ;
    margin-right: 1em ; }
span.subparagraph {font-size: 1em ; font-variant: normal ;
    margin-right: 1em ;  }

div.minisec {
    font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    font-style: normal ;
    font-weight: bold ;
    text-align: left ;
}

/* Title of the file */
h1 {
  margin: 0ex 0em 0ex 0em ;
  line-height: 1.3;
  text-align: center ;
}

/* Part */
h2 {
  margin: 1ex 0em 1ex 0em ;
  line-height: 1.3;
  text-align: center ;
}

/* Chapter */
h3 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Section */
h4 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Section */
h5 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Sub-Section */
h6 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}


div.titlepage {
  text-align: center ;
}

.footnotes {
    font-size: .85em ;
    margin: 3ex 1em 0ex 1em ;
    padding-bottom: 1ex ;
    border-top: 1px solid silver ;
}

.marginpar, .marginparblock {
    max-width:50%;
    float:right;
    text-align:left;
    margin: 1ex 0.5em 1ex 1em ;
    padding: 1ex 0.5em 1ex 0.5em ;
    font-size: 85% ;
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    overflow-x: auto;
}

.marginpar br { margin-bottom: 2ex ; }

div.marginblock, div.marginparblock {
    max-width:50%;
    float:right;
    text-align:left;
    margin: 1ex 0.5em 1ex 1em ;
    padding: 1ex 0.5em 1ex 0.5em ;
    overflow-x: auto;
}

div.marginblock div.minipage,
div.marginparblock div.minipage {
    display: block ;
    margin: 0pt auto 0pt auto ;
}

div.marginblock div.minipage p ,
div.marginparblock div.minipage p
    { font-size: 85%}

div.marginblock br ,
div.marginparblock br
    { margin-bottom: 2ex ; }


section.textbody div.footnotes{
    margin: 3ex 0em 0ex 0em ;
    border-bottom: 2px solid silver ;
}

.footnoteheader {
    border-top: 2px solid silver ;
    margin-top: 3ex ;
    padding-top: 1ex ;
    font-weight: bold ;
}

.mpfootnotes {
    text-align: left ;
    font-size: .85em ;
    margin-left: 1em ;
    border-top: 1px solid silver ;
}

/* Remove footnote top border in the title page. */
div.titlepage div.mpfootnotes {
    border-top: none ;
}



ol {
  margin: 1ex 1em 1ex 0em;
  line-height: 1.2;
```

```
}

ul, body dir, body menu {
  margin: 1ex 1em 1ex 0em;
  line-height: 1.2;
}

li { margin: 0ex 0em 1ex 0em; }

html {
  margin: 0;
  padding: 0;
}

.programlisting {
  font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
  margin: 1ex 0ex 1ex 0ex ;
  padding: .5ex 0pt .5ex 0pt ;
  overflow-x: auto;
}

section.textbody>pre.programlisting {
border-top: 1px solid silver ;
border-bottom: 1px solid silver ;
}


.inlineprogramlisting {
  font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
  overflow-x: auto;
}

span.listinglabel {
    display: inline-block ;
    font-size: 70% ;
    width: 4em ;
    text-align: right ;
    margin-right: 2em ;
}

div.abstract {
  margin: 2em 5% 2em 5% ;
  padding: 1ex 1em 1ex 1em ;
/*  font-weight: bold ; */
  font-size: 90% ;
    text-align: left ;
```

```
}

div.abstract dl {line-height:1.5;}
div.abstract dt {color:#304070;}

div.abstracttitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
    font-size:1.25em;
    text-align: center ;
}

span.abstractrunintitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
}


.verbatim {
    overflow-x: auto ;
}

.alltt {
    overflow-x: auto ;
}


.bverbatim {
    margin: 1ex 0pt 1ex 0pt ;
    padding: .5ex 0pt .5ex 0pt ;
    overflow-x: auto ;
}

.lverbatim {
    margin: 1ex 0pt 1ex 0pt ;
    padding: .5ex 0pt .5ex 0pt ;
    overflow-x: auto ;
}

.fancyvrb {
    font-size:.85em ;
    margin: 3ex 0pt 3ex 0pt
}

.fancyvrblabel {
    font-weight:bold;
    text-align: center ;
}


.verse {
    font-family: "Linux Libertine Mono O", "Lucida Console",
        "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
        "Liberation Mono", "FreeMono", "Andale Mono",
        "Nimbus Mono L", "Courier New", monospace;
    margin-left: 1em ;
}


div.singlespace { line-height: 1.2 ; }
div.onehalfspace { line-height: 1.5 ; }
div.doublespace { line-height: 2 ; }


/* Word processor format output: */
div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}




/* Minipage environments, vertically aligned to top, center, bottom: */
.minipage, .fminipage, .fcolorminipage {
    /* display: inline-block ; */
        /* Mini pages which follow each other will be tiled. */
    margin: .25em .25em .25em .25em;
    padding: .25em .25em .25em .25em;
    display: inline-flex;
    flex-direction: column ;
    overflow: auto;
}

/* Paragraphs in the flexbox did not collapse their margins. */
/* Have not yet researched this. */
.minipage p {margin: .75ex 0em .75ex 0em ;}

.fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
.fcolorbox .minipage, .fcolorboxBlock .minipage
    {border: none ; background: none;}

.fbox, .fboxBlock { border: 1px solid black ; }

.fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
.fminipage, .fcolorminipage
    {display: inline-block}
```

```

.shadowbox, .shabox {
   border: 1px solid black;
    box-shadow: 3px 3px 3px #808080 ;
     border-radius: 0px ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
  display: inline-block ;
}

.doublebox {
   border: 3px double black;
     border-radius: 0px ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
  display: inline-block ;
}

.ovalbox, .Ovalbox {
   border: 1px solid black;
     border-radius: 1ex ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
  display: inline-block ;
}

.Ovalbox { border-width: 2px ; }

.framebox {
   border: 1px solid black;
     border-radius: 0px ;
    padding: .3ex .2em 0ex .2em ;
    margin: 0pt .1ex 0pt .1ex ;
  display: inline-block ;
}


.mdframed {
/*     padding: 0ex ; */
/*    border: 1px solid blafck; */
/*      border-radius: 0px ; */
    padding: 0ex ;
    margin: 3ex 5% 3ex 5% ;
/*   display: inline-block ; */
}

.mdframed p { padding: 0ex .5em 0ex .5em ; }

.mdframed dl { padding: 0ex .5em 0ex .5em ; }

```

```
.mdframedtitle {
    padding: .5em ;
    display: block ;
    font-size: 130%
}

.mdframedsubtitle {
    padding: 0ex .5em 0ex .5em ;
    display: block ;
    font-size: 115% ;
}

.mdframedsubsubtitle {
    padding: 0ex .5em 0ex .5em ;
    display: block ;
}

.mdtheorem {
    padding: 0ex .5em 0ex .5em ;
    margin: 3ex 5% 3ex 5% ;
/*   display: inline-block ; */
}


/* framed package */
.framed, pre.boxedverbatim, fcolorbox {
    margin: 3ex 0em 3ex 0em ;
   border: 1px solid black;
     border-radius: 0px ;
    padding: .3ex 1em 0ex 1em ;
  display: block ;
}

.snugframed {
    margin: 3ex 0em 3ex 0em ;
   border: 1px solid black;
     border-radius: 0px ;
  display: block ;
}

.framedleftbar {
    margin: 3ex 0em 3ex 0em ;
   border-left: 3pt solid black;
     border-radius: 0px ;
    padding: .3ex .2em .3ex 1em ;
  display: block ;
}

.framedtitle {
    margin: 0em ;
    padding: 0em ;
    font-size: 130%
}

.framedtitle p { padding: .3em }



dl {
  margin: 1ex 2em 1ex 0em;
  line-height: 1.3;
}

dl dt {
    margin-top: 1ex;
    margin-left: 1em ;
    font-weight: bold;
}

dl dd p { margin-top: 0em; }


nav {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    margin-bottom: 4ex ;
}

nav p {
    line-height: 1.2 ;
    margin-top:.5ex ;
    margin-bottom:.5ex;
    font-size: .9em ;
}



img, img.hyperimage, img.borderimage {
    max-width: 600px;
    border: 1px solid silver;
    box-shadow: 3px 3px 3px #808080 ;
    padding: .5% ;
    margin: .5% ;
    background: none ;
}

img.inlineimage{
    padding: 0px ;
    box-shadow: none ;
    border: none ;
    background: none ;
    margin: 0px ;
    display: inline-block ;
    border-radius: 0px ;
}

img.logoimage{
    max-width: 300px ;
    box-shadow: 3px 3px 3px #808080 ;
    border: 1px solid black ;
    background:none ;
    padding:0 ;
    margin:.5ex ;
    border-radius: 10px ;
}


.section {
/*
    To have each section float relative to each other:
*/
/*
    display: block ;
    float: left ;
    position: relative ;
    background: white ;
    border: 1px solid silver ;
    padding: .5em ;
*/
    margin: 0ex .5em 0ex .5em ;
    padding: 0 ;
}


figure {
    margin: 3ex auto 3ex auto ;
    padding: 1ex 1em 1ex 1em ;
    overflow-x: auto ;
}


/* To automatically center images in figures: */
/*
figure img.inlineimage {
    margin: 0ex auto 0ex auto ;
    display: block ;
}
*/

```

```
/* To automatically center minipages in figures: */
/*
figure div.minipage, figure div.minipage div.minipage {
    margin: 1ex auto 1ex auto ;
    display: block ;
}
*/

figure div.minipage p { font-size: 85% ; }

figure.subfigure, figure.subtable  {
    display: inline-block ; margin: 3ex 1em 3ex 1em ;
}

figcaption .minipage { margin:0 ; padding: 0 }

div.minipage figure { border: none ; box-shadow: none ; }

div.floatrow { text-align: center; }

div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }

div.floatfoot { font-size: .85em ;
    border-top: 1px solid silver ; line-height: 1.2 ; }

figcaption , .lstlistingtitle {
    font-size: .85em ;
    text-align: center ;
    font-weight: bold ;
    margin-top: 1ex ;
    margin-bottom: 1ex ;
}

figure.subfigure figcaption, figure.subtable figcaption  {
    border-bottom: none ; background: none ;
}

div.nonfloatcaption {
    margin: 1ex auto 1ex auto ;
    font-size: .85em ;
    text-align: center ;
    font-weight: bold ;
}

/* For a \RawCaption inside a minipage inside a figure's floatrow: */
figure div.floatrow div.minipage figcaption {
    border: none ;
    background: none ;
}

```

```

table {
    margin: 1ex auto 1ex auto ;
    border-collapse: separate ;
    border-spacing: 0px ;
    line-height: 1.3 ;
    }

tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \hline */

tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \toprule, \bottomrule */

td {padding: 1ex .5em 1ex .5em ;}

table td.tdl { text-align: left ; vertical-align: middle ; }
table td.tdc { text-align: center ; vertical-align: middle ; }
table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
table td.tdbang { text-align: center ; vertical-align: middle ; }
table td.tdr { text-align: right ; vertical-align: middle ; }
table td.tdp { text-align: left ; vertical-align: bottom ; }
table td.tdm { text-align: left ; vertical-align: middle ; }
table td.tdb { text-align: left ; vertical-align: top ; }
table td.tdP { text-align: center ; vertical-align: bottom ; }
table td.tdM { text-align: center ; vertical-align: middle ; }
table td.tdB { text-align: center ; vertical-align: top ; }

table td.tvertbarl { border-left: 1px solid black }
table td.tvertbarr { border-right: 1px solid black }


/* for cmidrules: */
table td.tdrule {
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulel {
    border-top-left-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdruler {
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulelr {
    border-top-left-radius:.5em ;
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}


/* Margins of paragraphs inside table cells: */
td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }

td.tdp , td.tdprule , td.tdP , td.tdPrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdm , td.tdmrule , td.tdM , td.tdMrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdb , td.tdbrule , td.tdB , td.tdBrule
    { padding: 0ex .5em 0ex .5em ; }


/* table notes: */
.tnotes {
    margin: 0ex 5% 1ex 5% ;
    padding: 0.5ex 1em 0.5ex 1em;
    font-size:.85em;
    text-align: left ;
}

.tnotes dl dt p {margin-bottom:0px;}

.tnoteitemheader {margin-right: 1em;}


/* for bigdelim */
.ldelim, .rdelim { font-size: 200% }


/* center, flushleft, flushright environments */
div.center{text-align:center;}
div.center table {margin-left:auto;margin-right:auto;}
div.flushleft{text-align:left;}
div.flushleft table {margin-left:0em ; margin-right:auto;}
div.flushright{text-align:right;}
div.flushright table {margin-left:auto ; margin-right: 0em ;}


/* Fancybox */
div.Btrivlist table tr td { padding: .2ex 0em ; }


```

```
/* program listing callouts: */
span.callout {
     font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    border-radius: .5em;
    background-color:black;
    color:white;
    padding:0px .25em 0px .25em;
    margin: 0 ;
    font-weight: bold;
    font-size:.72em ;
}

div.programlisting pre.verbatim span.callout{
    font-size: .85em ;
}





div.published
{
    text-align: center ;
    font-variant: normal ;
    font-style: italic ;
    font-size: 1em ;
    margin: 3ex 0em 3ex 0em ;
}

div.subtitle
{
    text-align: center ;
    font-variant: normal ;
    font-style: italic ;
    font-size: 1.25em ;
    margin: 3ex 0em 3ex 0em ;
}

div.subtitle p { margin: 1ex ; }

div.author
{
    font-variant: normal ;
    font-style: normal ;
    font-size: 1em ;
    margin: 3ex 0em 3ex 0em ;
}

div.oneauthor {
    display: inline-block ;
    margin: 3ex 1em 0ex 1em ;
}

/*
div.author table {
    margin: 3ex auto 0ex auto ;
    background: none ;
}

div.author table tbody tr td { padding: .25ex ; }
*/

span.affiliation {font-size: .85em ; font-variant: small-caps; }

div.titledate {
    text-align: center ;
    font-size: .85em ;
    font-style: italic;
    margin: 6ex 0em 6ex 0em ;
}


nav.topnavigation{
    text-align: left ;
    padding: 0.5ex 1em 0.5ex 1em ;
/*     margin: 2ex 0em 3ex 0em ; */
    margin: 0 ;
    border-bottom: 1px solid silver ;
    border-top: 1px solid silver ;
    clear:right ;
}

nav.botnavigation{
    text-align: left ;
    padding: 0.5ex 1em 0.5ex 1em ;
/*     margin: 3ex 0em 2ex 0em ; */
    margin: 0 ;
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    clear:right ;
}


header{
    line-height: 1.2 ;
    font-size: 1em ;
/*     border-bottom: 2px solid silver ; */
    margin: 0px ;
    padding: 0ex 1em 0ex 1em ;
    text-align:center ;
}

header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}


footer{
    font-size: .85em ;
    line-height: 1.2 ;
    margin-top: 1ex ;
    border-top: 2px solid silver ;
    padding: 2ex 1em 2ex 1em ;
    clear:right ;
    text-align:left ;
}


a.linkhome { font-weight:bold ; font-size: 1em ;}


div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }

img.lateximage{
    padding: 0px 0px 0px 0px ;
    box-shadow: none ;
    border: none ;
    background: none ;
    margin: 0px 0px -.15ex 0px ;
        /* pdfcrop leaves a slight margin, adjust to baseline */
    max-width: 100% ;
    border-radius: 0ex ;
    border: none ;
}



nav.sidetoc {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
        "Lucida Bright", Georgia, serif;
    float:right ;
    width: 20%;
    border-left: 1px solid silver;
    border-top: 1px solid silver;
    border-bottom: 1px solid silver;
/*     border-top: 2px solid #808080 ; */
    background: #FAF7F4 ;
    padding: 2ex 0em 2ex 1em ;
    margin: 0ex 0em 2ex 1em ;
    font-size:.9em ;
    border-radius: 20px 0px 0px 20px ;
    }

div.sidetoccontents {
/*     border-top: 1px solid silver ; */
    overflow-y: auto ;
    width: 100% ;
    text-align: left ;
}


nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
    text-indent: 0 ; }

nav.sidetoc p a {color:black ; font-size: .7em ;}

div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
    border-bottom: 1px solid silver ;    }

nav.sidetoc a:hover {text-decoration: underline ; }



section.textbody { margin: 0ex 1em 0ex 1em ;}


div.multicolsheading { -webkit-column-span: all;
    -moz-column-span: all; column-span: all; }
div.multicols { -webkit-columns: 3 380px ;
    -moz-columns: 3 380px ; columns: 3 380px ; }
div.multicols p {margin-top: 0ex}



/* Used to support algorithmicx: */
span.floatright { float: right ; }




/* Native LaTeX theorems: */

.theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
.theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }


/* theorem, amsthm, and ntheorem packages */

span.theoremheader,
span.theoremheaderplain,
span.theoremheaderdefinition,
span.theoremheaderbreak,
span.theoremheadermarginbreak,
span.theoremheaderchangebreak,
span.theoremheaderchange,
span.theoremheadermargin
{
    font-style:normal ; font-weight: bold ; margin-right: 1em ;
}

span.amsthmnameplain,
span.amsthmnamedefinition,
span.amsthmnumberplain,
span.amsthmnumberdefinition
{
    font-style:normal ; font-weight: bold ;
}


span.amsthmnameremark,
span.amsthmnumberremark
{font-style:italic ; font-weight: normal ; }


span.amsthmnoteplain,
span.amsthmnotedefinition
{font-style:normal ;}


span.theoremheaderremark,
span.theoremheaderproof,
span.amsthmproofname
{font-style:italic ; font-weight: normal ; margin-right: 1em ; }

span.theoremheadersc
{
    font-style:normal ;
    font-variant: small-caps ;
    font-weight: normal ;
    margin-right: 1em ;
}

.theoremendmark {float:right}

div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
div.theorembodybreak, div.theorembodynonumberbreak,
div.theorembodymarginbreak,
div.theorembodychangebreak,
div.theorembodychange,
div.theorembodymargin
{
    font-style:italic;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
div.theorembodyplainupright, nonumberplainuprightsc,
div.amsthmbodydefinition, div.amsthmbodyremark,
div.amsthmproof
{
    font-style: normal ;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

span.amsthmnoteremark {}



/*
For CSS LaTeX and related logos:
Based on:
http://edward.oconnor.cx/2007/08/tex-poshlet
http://nitens.org/taraborelli/texlogo
*/

.latexlogofont {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: normal ;
}

.latexlogo {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    letter-spacing: .03em ;
    font-size: 1.1em;
}

.latexlogo sup {
  text-transform: uppercase;
  letter-spacing: .03em ;
  font-size: 0.85em;
  vertical-align: 0.15em;
  margin-left: -0.36em;
  margin-right: -0.15em;
}

.latexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.1667em;
  margin-right: -0.125em;
  font-size: 1em;
}

.xetexlogo {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    letter-spacing: .03em ;
    font-size: 1.1em;
}

/* A smaller gap between Xe and Tex v.s. LaTeX: */
.xetexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.0667em;
  margin-right: -0.2em;
  font-size: 1em;
  letter-spacing: .03em ;
}

/* A large gap between Xe and LaTeX v.s. TeX: */
.xelatexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.0667em;
  margin-right: -.05em;
  font-size: 1em;
  letter-spacing: .03em ;
}

.amslogo {
    font-family: "TeXGyreChorus","URW Chancery L",
        "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
        "Linux Libertine O", "Nimbus Roman No 9 L",  "FreeSerif",
        "Hoefler Text", Times, "Times New Roman", serif;
  font-style: italic;
}

.lyxlogo {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "DejaVu Sans", "Bitstream Vera Sans", Geneva,
        Verdana, sans-serif ;
}




/* Only display top and bottom navigation if a small screen: */
/* Hide the sidetoc if a small screen: */
```

```
nav.topnavigation { display:none; }
nav.botnavigation { display:none; }

@media screen and (max-width: 45em) {
/*     nav.sidetoc {display:none;} */
    nav.sidetoc {
        float: none ;
        width: 100% ;
        margin: 5ex 0px 5ex 0px ;
        padding: 0 ;
        border-radius: 0 ;
        border-bottom: 1px solid black ;
        border-top: 1px solid black ;
        box-shadow: none ;
    }
/*     nav.topnavigation { display:block } */
    nav.botnavigation { display:block }
    .marginpar {
        max-width: 100%;
        float: none;
        display:block ;
        margin: 1ex 1em 1ex 1em ;
    }
}

@media print {
    body {
        font-family: "Linux Libertine O",
        "DejaVu Serif", "Bitstream Vera Serif",
        "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    }
    nav.sidetoc { display:none; }
    nav.topnavigation { display: none; }
    nav.botnavigation { display: none; }
}

@media handheld {
    nav.sidetoc { display:none; }
    nav.topnavigation { display:block }
    nav.botnavigation { display:block }
}

@media projection {
    nav.sidetoc { display:none; }
    nav.topnavigation { display:block }
    nav.botnavigation { display:block }
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
```

```
\end{warpprint}
```

## 33.5 lwarp_sagebrush.css

File lwarp_sagebrush.css

An optional CSS which may be used for a semi-modern appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_sagebrush.css}
@import url("lwarp.css") ;


A:link {color:#105030 ; text-decoration: none ; }
A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}



h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif",
        "Nimbus Roman No 9 L",  "FreeSerif",
        "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: small-caps ;
    font-weight: normal ;
    color: #304070 ;
    text-shadow: 2px 2px 3px #808080;
}

h1 {    /* title of the entire website, used on each page */
    font-variant: small-caps ;
    color: #304070 ;
    text-shadow: 2px 2px 3px #808080;
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
}

h1 {
  border-bottom: 1px solid #304070;
  border-top: 2px solid #304070;
}

h2 {
  border-bottom: 1px solid #304070;
  border-top: 2px solid #304070;
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
}



div.abstract {
    background: #f5f5eb ;
    background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);

  border: 1px solid silver;
    border-radius: 1em ;
}

div.abstract dl {line-height:1.5;}
div.abstract dt {color:#304070;}

div.abstracttitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
    font-variant: small-caps ;
    font-size:1.5em;
    border-bottom: 1px solid silver ;
    color: #304070 ;
    text-align: center ;
    text-shadow: 1px 1px 2px #808080;
}

span.abstractrunintitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
}


div.epigraph, div.dictum {
    background: #f5f5eb ;
    background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);

    border: 1px solid silver ;
    border-radius: 1ex ;
    box-shadow: 3px 3px 3px #808080 ;
}


```

```
.example {
    background-color: #f5f5eb ;
    background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);

}

div.exampletitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
    font-variant: small-caps ;
    border-bottom: 1px solid silver ;
    color: #304070 ;
    text-align: center ;
    text-shadow: 1px 1px 2px #808080;
}


.sidebar {
    background-color: #f5f5eb ;
    background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);

}

div.sidebartitle{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
    font-variant: small-caps ;
    border-bottom: 1px solid silver ;
    color: #304070 ;
    text-align: center ;
    text-shadow: 1px 1px 2px #808080;
}


.fancyvrblabel {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-weight:bold;
    font-variant: small-caps ;
    font-size: 1.5em ;
    color: #304070 ;
    text-align: center ;
    text-shadow: 1px 1px 2px #808080;
}

```

```


div.minipage {
    background-color: #eeeee7 ;
    border: 1px solid silver ;
    border-radius: 1ex ;
}

section.textbody > div.minipage {
    box-shadow: 3px 3px 3px #808080 ;
}

div.fboxBlock div.minipage { box-shadow: none ; }

.framed .minipage , .framedleftbar .minipage {
    border: none ;
    background: none ;
    padding: 0ex ;
    margin: 0ex ;
}

figure.figure .minipage, figcaption .minipage { border: none; }

div.marginblock div.minipage ,
div.marginparblock div.minipage
    { border: none; }

figure , div.marginblock {
    background-color: #eeeee7 ;
    border: 1px solid silver ;
    border-radius: 1ex ;
    box-shadow: 3px 3px 3px #808080 ;
}

figure figure {
    border: 1px solid silver ;
    margin: 0em ;
    box-shadow: none ;
}

/*
figcaption {
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    background-color: #e8e8e8 ;
}
*/


div.table {
    box-shadow: 3px 3px 3px #808080 ;
}

/*
.tnotes {
    background: #e8e8e8;
    border: 1px solid silver;
}
*/


nav.topnavigation{
    background-color: #b0b8b0 ;
    background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
}

nav.botnavigation{
    background-color: #b0b8b0 ;
    background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
}



header{
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
}

footer{
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
}



nav.sidetoc {
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
    box-shadow: 3px 3px 3px #808080 ;
    border-radius: 0px 0px 0px 20px ;
    }

div.sidetoctitle {color: #304070 ; }

nav.sidetoc a:hover {
    color:#006000 ;
    text-decoration: none ;
    text-shadow:0px 0px 2px #a0a0a0;
}

```

```

@media screen and (max-width: 45em) {
    nav.sidetoc { border-radius: 0 ; }
}


\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

## 33.6 lwarp_formal.css

File lwarp_formal.css An optional CSS which may be used for a more formal appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_formal.css}
@import url("lwarp.css") ;



A:link {color:#802020 ; text-decoration:none; }
A:visited {color:#802020 ; text-shadow:none ;}
A:hover {color:#400000 ; text-shadow:none ;}
A:active {color:#C00000 ; text-shadow:none ;}


body {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    background: #fffcf5;
}

span.textrm {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
}

span.textsf {
     font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
```

```
}



h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
}

h1, h2 {
    background-color: #fffcf5 ;
    background-image: none ;
    border-bottom: 1px solid #808080;
    border-top: 2px solid #808080;
}

div.abstracttitle {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: black ;
    text-shadow: none ;
}

span.abstractrunintitle {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: black ;
    text-shadow: none ;
}

div.abstract { font-size: 100% }

.sidebar {
    background: #fffcf5;
    background-image: none ;
  margin: 2em 5% 2em 5%;
  padding: 0.5em 1em;
  border: none ;
  border-top : 1px solid silver;
  border-bottom : 1px solid silver;
  font-size: 90% ;
```

```
}

div.sidebartitle{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}

.example {
    background: #fffcf5;
    background-image: none ;
  margin: 2em 5% 2em 5%;
  padding: 0.5em 1em;
  border: none ;
  border-top : 1px solid silver;
  border-bottom : 1px solid silver;
}

div.exampletitle{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}

div.fancyvrblabel{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}



.verse {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
}
```

```


figure {
    margin: 3ex 5% 3ex 5% ;
    padding: 1ex 1em 1ex 1em ;
    background-color: #fffcf5 ;
    overflow-x: auto ;
    border: none ;
/*     border-top: 1px solid silver; */
/*     border-bottom: 1px solid silver; */
}


figcaption , .lstlisting {
    border: none ;
/*     border-top: 1px solid silver ; */
/*     border-bottom: 1px solid silver ; */
    background-color: #fffcf5 ;
}

.tnotes {
    background: #fffcf5 ;
}

.theorem {
        background: none ;
}

.minipage {
    background-color: #fffcf5 ;
    border: none ;
}

div.floatrow figure { border: none ; }

figure figure { border: none ; }


nav.toc, nav.lof, nav.lot, nav.lol {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
}

nav.sidetoc {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
    border-radius: 0px 0px 0px 20px ;
}

div.sidetoctitle{
    color: #800000 ;
}

header{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
    text-align:center ;
}

footer{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
    padding: 2ex 1em 2ex 1em ;
    clear:right ;
    text-align:left ;
}

nav.botnavigation {
    background: #dedcd5 ;
    border-top: 1px solid black ;
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

## 33.7 sample_project.css

File `sample_project.css`

The project-specific CSS file. Use with `\CSSFilename`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{sample_project.css}
/* ( --- Start of project.css --- ) */
/* ( --- A sample project-specific CSS file for lwarp --- ) */

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

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

/* ( --- End of project.css --- ) */
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

## 33.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for lwarp.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp.xdy}
(require "tex/inputenc/latin.xdy")
(merge-rule "\\PS *" "Postscript")
(require "texindy.xdy")
(require "page-ranges.xdy")
(require "book-order.xdy")
(markup-locref :open "\hyperindexref{" :close "}")
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

## 33.9 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by lwarp when using MathJax.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_mathjax.txt}
<!-- https://groups.google.com/forum/#!topic/
                               mathjax-users/jUtewUcE2bY -->
<script type="text/x-mathjax-config">
MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
    var seteqsectionDefault = {name: "", num: 0};
    var seteqsections = {}, seteqsection = seteqsectionDefault;
    var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
    var AMS = MathJax.Extension["TeX/AMSmath"];
    TEX.Definitions.Add({
    macros: {
        seteqsection: "mySection",
        seteqnumber: "mySetEqNumber"
    }
    });

    PARSE.Augment({
    mySection: function (name) {
        seteqsection.num = AMS.number;
        var n = this.GetArgument(name);
        if (n === "") {
        seteqsection = seteqsectionDefault;
        } else {
        if (!seteqsections["_"+n])
            seteqsections["_"+n] = {name:n, num:0};
        seteqsection = seteqsections["_"+n];
        }
        AMS.number = seteqsection.num;
    },
    mySetEqNumber: function (name) {
        var n = this.GetArgument(name);
        if (!n || !n.match(/^ *[0-9]+ *$/))
            n = ""; else n = parseInt(n)-1;
        <!-- $ syntax highlighting -->
        if (n === "" || n < 1)
            TEX.Error
            ("Argument to "+name+" should be a positive integer");
        AMS.number = n;
    }
    });
    MathJax.Hub.Config({
    TeX: {
        equationNumbers: {
        formatTag: function (n)
            {return "("+(seteqsection.name+"."+n).replace(/^\./,"")+")"},
        formatID: function (n) {
            n = (seteqsection.name+’.’+n).replace
                (/[:"’<>&]/g,"").replace(/^\./,"");
            return ’mjx-eqn-’ + n;
        }
        }
    }
    });
});
</script>

<!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
<script type="text/x-mathjax-config">
  MathJax.Ajax.config.path["Contrib"] =
    "https://cdn.mathjax.org/mathjax/contrib";
</script>
```

```

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

<script type="text/x-mathjax-config">
 MathJax.Hub.Config({
   extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
   jax: ["input/TeX","output/HTML-CSS"],
   tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
   TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
 });
 MathJax.Ajax.config.path['siunitx']  = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
 </script>

<script type="text/x-mathjax-config">
MathJax.Hub.Config({
    TeX: {
    equationNumbers: {
        autoNumber: "AMS"
    }
    }
});
</script>

<!-- Alternative CDN provider: -->
<script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
</script>

<!-- No longer supported after April 30, 2017: -->
<!--
<script
  src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
</script>
-->

\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

## 33.10 lwarpmk option

The following is only generated if the `lwarpmk` option was given to lwarp.

```
\begin{LWR@createlwarpmk}
```

Prog `lwarpmk` Creates a local copy of `lwarpmk`:

```
\begin{filecontents*}{lwarpmk.lua}
#!/usr/bin/env texlua

-- Copyright 2016-2017 Brian Dunn

-- Print the usage of the lwarpmk command:

printversion = "v0.44"

function printhelp ()
print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
end

function printusage ()
print ( [[

lwarpmk print [project]: Compile the print version if necessary.
lwarpmk print1 [project]: Forced single compile of the print version.
lwarpmk printindex [project]: Process the index for the print version.
lwarpmk printglossary [project]: Process the glossary for the print version.
lwarpmk html [project]: Compile the HTML version if necessary.
lwarpmk html1 [project]: Forced single compile of the HTML version.
lwarpmk htmlindex [project]: Process the index for the html version.
lwarpmk htmlglossary [project]: Process the glossary for the html version.
lwarpmk again [project]: Touch the source code to trigger recompiles.
lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
lwarpmk pdftohtml [project]:
    For use with latexmk or a Makefile:
    Convert project_html.pdf to project_html.html and
    individual HTML files.
lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
lwarpmk -h: Print this help message.
lwarpmk --help: Print this help message.

]] )
printconf ()
end

-- Print the format of the configuration file lwarpmk.conf:

function printconf ()
print ( [[
An example lwarpmk.conf or <project>.lwarpmkconf project file:
--
opsystem = "Unix"   (or "Windows")
latexname = "pdflatex"  (or "lualatex", or "xelatex")
sourcename = "projectname"  (the source-code filename w/o .tex)
homehtmlfilename = "index"  (or perhaps the project name)
htmlfilename = ""  (or "projectname" - filename prefix)
latexmk = "false"  (or "true" to use latexmk to build PDFs)
languge = "english"  (use a language supported by xindy)
xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
--
Filenames must contain only letters, numbers, underscore, or dash.
Values must be in "quotes".

]] ) ;
end


-- Split one large sourcefile into a number of files,
-- starting with destfile.
-- The file is split at each occurance of <!--|Start file|newfilename|*

function splitfile (destfile,sourcefile)
print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
local sfile = io.open(sourcefile)
io.output(destfile)
for line in sfile:lines() do
i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
io.output(newfilename) ;
else -- not a splitpoint
io.write (line .. "\n") ;
end
end -- do
io.close(sfile)
end -- function

-- Incorrect value, so print an error and exit.

function cvalueerror ( line, linenum , cvalue )
    print ( linenum .. " : " .. line ) ;
    print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
    printconf () ;
    os.exit(1) ;
end

-- Load settings from the project's "lwarpmk.conf" file:

function loadconf ()
-- Default configuration filename:
local conffile = "lwarpmk.conf"
-- Optional configuration filename:
if arg[2] ~= nil then conffile = arg[2]..".lwarpmkconf" end
-- Default language:
language = "english"
-- Default xdyfile:
xdyfile = "lwarp.xdy"
```

```
-- Verify the file exists:
if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
print("lwarpmk: " .. conffile .." does not exist.")
print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
printhelp () ;
os.exit(1) -- exit the entire lwarpmk script
else -- file exists
-- Read the file:
print ("lwarpmk: Reading " .. conffile ..".")
local cfile = io.open(conffile)
-- Scan each line:
local linenum = 0
for line in cfile:lines() do -- scan lines
linenum = linenum + 1
i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_%.]*)\"") ;
-- Error if incorrect enclosing characters:
if ( i == nil ) then
print ( linenum .. " : " .. line ) ;
print ( "lwarpmk: Incorrect entry in " .. conffile ..".\n" ) ;
printconf () ;
os.exit(1) ;
end
if ( cvarname == "opsystem" ) then
    -- Verify choice of opsystem:
    if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
        opsystem = cvalue
    else
        cvalueerror ( line, linenum , cvalue )
    end
elseif ( cvarname == "latexname" ) then
    -- Verify choice of LaTeX compiler:
    if (
        (cvalue == "pdflatex") or
        (cvalue == "xelatex") or
        (cvalue == "lualatex")
    ) then
        latexname = cvalue
    else
        cvalueerror ( line, linenum , cvalue )
    end
elseif ( cvarname == "sourcename" ) then sourcename = cvalue
elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
elseif ( cvarname == "latexmk" ) then latexmk = cvalue
elseif ( cvarname == "language" ) then language = cvalue
elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
else
print ( linenum .. " : " .. line ) ;
print ("lwarpmk: Incorrect variable name \"" .. cvarname .. "\" in " .. conffile ..".\n" ) ;
printconf () ;
```

```
os.exit(1) ;
end
end -- do scan lines
io.close(cfile)
end -- file exists
-- Select some operating-system commands:
if opsystem=="Unix" then  -- For Unix / Linux / Mac OS:
rmname = "rm"
mvname = "mv"
touchnamepre = "touch"
touchnamepost = ""
dirslash = "/"
opquote= "\’"
elseif opsystem=="Windows" then -- For Windows
rmname = "DEL"
mvname = "MOVE"
touchnamepre = "COPY /b"
touchnamepost = "+,,"
dirslash = "\\"
opquote= "\""
else print ( "lwarpmk: Select Unix or Windows for opsystem" )
end --- for Windows

-- set xindycmd according to pdflatex vs xelatex/lualatex:
if ( latexname == "pdflatex" ) then
xindycmd = "texindy  -C utf8"
glossarycmd = "xindy -C utf8"
else
xindycmd = "xindy  -M texindy  -C utf8"
glossarycmd = "xindy -C utf8"
end

end -- loadconf


function refreshdate ()
os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
end


-- Scan the LaTeX log file for the phrase "Rerun to get",
-- indicating that the file should be compiled again.
-- Return true if found.

function reruntoget (filesource)
local fsource = io.open(filesource)
for line in fsource:lines() do
if ( string.find(line,"Rerun to get") ~= nil ) then
    io.close(fsource)
    return true
```

```
end
end
io.close(fsource)
return false
end


-- Compile one time, return true if should compile again.
-- fsuffix is "" for print, "_html" for HTML output.

function onetime (fsuffix)
print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
err = os.execute(
--    "echo " ..
    latexname .. " " .. sourcename..fsuffix )
if ( err ~= 0 ) then print ( "lwarpmk: Compile error.") ; os.exit(1) ; end
return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
end


-- Compile up to five times.
-- fsuffix is "" for print, "_html" for HTML output

function manytimes (fsuffix)
if onetime(fsuffix) == true then
if onetime(fsuffix) == true then
if onetime(fsuffix) == true then
if onetime(fsuffix) == true then
if onetime(fsuffix) == true then
end end end end end
end

-- Exit if the given file does not exist.

function verifyfileexists (filename)
if (lfs.attributes ( filename , "modification" ) == nil ) then
print ( "lwarpmk: " .. filename .. " not found." ) ;
os.exit (1) ;
end
end


-- Convert <project>_html.pdf into HTML files:

function pdftohtml ()
    -- Convert to text:
    print ("lwarpmk: Converting " .. sourcename
        .."_html.pdf to " .. sourcename .. "_html.html")
    os.execute("pdftotext  -enc UTF-8  -nopgbrk  -layout "
        .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
    -- Split the result into individual HTML files:
    splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
end


-- Remove auxiliary files:

function removeaux ()
    os.execute ( rmname .. " " ..
        sourcename ..".aux " .. sourcename .. "_html.aux " ..
        sourcename ..".toc " .. sourcename .. "_html.toc " ..
        sourcename ..".lof " .. sourcename .. "_html.lof " ..
        sourcename ..".lot " .. sourcename .. "_html.lot " ..
        sourcename ..".idx " .. sourcename .. "_html.idx " ..
        sourcename ..".ind " .. sourcename .. "_html.ind " ..
        sourcename ..".log " .. sourcename .. "_html.log " ..
        sourcename ..".gl* " .. sourcename .. "_html.gl* " ..
        "*_html_inc.*"
        )
end



-- Create lateximages based on lateximages.txt:
function createlateximages ()
print ("lwarpmk: Creating lateximages.")
local limagesfile = io.open("lateximages.txt")
-- Create the lateximages directory, ignore error if already exists
err = os.execute("mkdir lateximages")
-- Scan lateximages.txt
for line in limagesfile:lines() do
-- lwimgpage is the page number in the PDF which has the image
-- lwimgnum is the sequential lateximage number to assign for the image
i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
-- For each entry:
if ( (i~=nil) ) then
-- Separate out the image into its own single-page pdf:
err = os.execute(
"pdfseparate -f " .. lwimgpage .. " -l " ..
 lwimgpage .. " " .. sourcename .."_html.pdf lateximagetemp-%d.pdf")
-- Crop the image:
err = os.execute(
"pdfcrop --hires lateximagetemp-" .. lwimgpage ..".pdf lateximage-" .. lwimgnum ..".pdf")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Convert the image to svg:
err = os.execute(
"pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Move the result into lateximages/:
err = os.execute(
mvname .. " lateximage-" .. lwimgnum ..".svg lateximages" .. dirslash )
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Remove the temporary files:
err = os.execute(
rmname .. " lateximage-" .. lwimgnum ..".pdf lateximagetemp-" .. lwimgpage ..".pdf")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
end
end -- do
io.close(limagesfile)
end -- function


-- Use latexmk to compile source and index:
-- fsuffix is "" for print, or "_html" for HTML
function compilelatexmk ( fsuffix )
    -- The recorder option is required to detect changes in <project>.tex
    -- while we are loading <project>_html.tex.
    err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
        .. "-e "
        .. opquote
        .. "$makeindex = q/" -- $
        .. xindycmd
        .. "  -M " .. xdyfile
        .. "  -L " .. language .. " /"
        .. opquote
        .. " -pdflatex=\"" .. latexname .." %O %S\" "
        .. sourcename..fsuffix ..".tex" ) ;
    if ( err ~= 0 ) then print ( "lwarpmk: Compile error.") ; os.exit(1) ; end
end



-- lwarpmk --version :

if (arg[1] == "--version") then
print ( "lwarpmk: " .. printversion )

else -- not --version

-- print intro:

print ("lwarpmk: " .. printversion .. "  Automated make for the LaTeX lwarp package.")

-- lwarpmk print:

if arg[1] == "print" then
loadconf ()
if ( latexmk == "true" ) then
    compilelatexmk ("")
    print ("lwarpmk: Done.")
```

```
else -- not latexmk
    verifyfileexists (sourcename .. ".tex") ;
    -- See if up to date:
    if (
        ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
        (
            lfs.attributes ( sourcename .. ".tex" , "modification" ) >
            lfs.attributes ( sourcename .. ".pdf" , "modification" )
        )
    ) then
        -- Recompile if not yet up to date:
        manytimes("")
        print ("lwarpmk: Done.") ;
    else
        print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
    end
end -- not latexmk

elseif arg[1] == "print1" then
    loadconf ()
    verifyfileexists (sourcename .. ".tex") ;
    onetime("")
    print ("lwarpmk: Done.") ;

-- lwarp printindex:
-- Compile the index then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "printindex" then
loadconf ()
print ("lwarpmk: Processing the index.")
os.execute(
    xindycmd
    .. "  -M " .. xdyfile
    .. "  -L " .. language
    .. " " .. sourcename .. ".idx")
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarp printglossary:
-- Compile the glossary then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "printglossary" then
loadconf ()
print ("lwarpmk: Processing the glossary.")

os.execute(glossarycmd .. "  -L " .. language .. "  -I xindy -M " .. sourcename ..
    " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
    .. sourcename .. ".glo")
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk html:

elseif arg[1] == "html" then
loadconf ()
if ( latexmk == "true" ) then
    compilelatexmk ("_html")
    pdftohtml ()
    print ("lwarpmk: Done.")
else -- not latexmk
    verifyfileexists ( sourcename .. ".tex" ) ;
    -- See if exists and is up to date:
    if (
        ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
        (
            lfs.attributes ( sourcename .. ".tex" , "modification" ) >
            lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
        )
    ) then
        -- Recompile if not yet up to date:
        manytimes("_html")
        pdftohtml ()
        print ("lwarpmk: Done.")
    else
        print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
    end
end -- not latexmk

elseif arg[1] == "html1" then
    loadconf ()
    verifyfileexists ( sourcename .. ".tex" ) ;
    onetime("_html")
    pdftohtml ()
    print ("lwarpmk: Done.")

elseif arg[1] == "pdftohtml" then
    loadconf ()
    pdftohtml ()

-- lwarpmk htmlindex:
-- Compile the index then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "htmlindex" then
```

```
loadconf ()
print ("lwarpmk: Processing the index.")
os.execute(
    xindycmd
    .. "  -M " .. xdyfile
    .. "  -L " .. language
    .. " " .. sourcename .. "_html.idx"
)
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk htmlglossary:
-- Compile the glossary then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "htmlglossary" then
loadconf ()
print ("lwarpmk: Processing the glossary.")

os.execute(glossarycmd .. "  -L " .. language .. "  -I xindy -M " ..sourcename ..
    "_html -t " .. sourcename .. "_html.glg -o " ..sourcename ..
    "_html.gls " ..sourcename .. "_html.glo")

print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk limages:
-- Scan the lateximages.txt file to create lateximages,
-- then touch the source to trigger a recompile.

elseif arg[1] == "limages" then
loadconf ()
print ("lwarpmk: Processing images.")
createlateximages ()
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk again:
-- Touch the source to trigger a recompile.

elseif arg[1] == "again" then
loadconf ()
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
```

```
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk clean:
-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*

elseif arg[1] == "clean" then
loadconf ()
removeaux ()
print ("lwarpmk: Done.")

-- lwarpmk cleanall
-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
--    and also project.pdf, *.html

elseif arg[1] == "cleanall" then
loadconf ()
removeaux ()
os.execute ( rmname .. " " ..
    sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
    "*.html"
    )
print ("lwarpmk: Done.")

-- lwarpmk with no argument :

elseif (arg[1] == nil) then
printhelp ()

-- lwarpmk -h or lwarpmk --help :

elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
printusage ()

else
print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
printhelp ()
end

end -- not --version
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
```

```
\end{LWR@createlwarpmk}
```

# 34 Stacks

for HTML output:

```
\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.

## 34.1 Assigning depths

initial depths for empty stack entries:

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

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

```
\newcommand*{\LWR@depthfinished}{-4}
\newcommand*{\LWR@depthpart}{-1}
\newcommand*{\LWR@depthchapter}{0}
\newcommand*{\LWR@depthsection}{1}
\newcommand*{\LWR@depthsubsection}{2}
\newcommand*{\LWR@depthsubsubsection}{3}
\newcommand*{\LWR@depthparagraph}{4}
\newcommand*{\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

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

## 34.2 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.

```
\newcommand*{\LWR@closeone}{}% top of the stack
\newcommand*{\LWR@closetwo}{}
\newcommand*{\LWR@closethree}{}
\newcommand*{\LWR@closefour}{}
\newcommand*{\LWR@closefive}{}
\newcommand*{\LWR@closesix}{}
\newcommand*{\LWR@closeseven}{}
\newcommand*{\LWR@closeeight}{}
\newcommand*{\LWR@closenine}{}
\newcommand*{\LWR@closeten}{}
\newcommand*{\LWR@closeeleven}{}
\newcommand*{\LWR@closetwelve}{}
```

## 34.3 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}
```

```
\newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
\newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
\newcommand*{\LWR@closedepththree}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
\newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
\newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
\newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
\newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
\newcommand*{\LWR@closedepthten}{\LWR@depthnone}
```

```
\newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
\newcommand*{\LWR@closedepthtwelve}{\LWR@depthnone}
```

## 34.4 Pushing and popping the stack

\pushclose {⟨*action*⟩} {⟨*depth*⟩}

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

```
\NewDocumentCommand{\pushclose}{m m}
{
\let\LWR@closetwelve\LWR@closeeleven
\let\LWR@closeeleven\LWR@closeten
\let\LWR@closeten\LWR@closenine
\let\LWR@closenine\LWR@closeeight
\let\LWR@closeeight\LWR@closeseven
\let\LWR@closeseven\LWR@closesix
\let\LWR@closesix\LWR@closefive
\let\LWR@closefive\LWR@closefour
\let\LWR@closefour\LWR@closethree
\let\LWR@closethree\LWR@closetwo
\let\LWR@closetwo\LWR@closeone
\let\LWR@closeone#1
\let\LWR@closedepthtwelve\LWR@closedeptheleven
\let\LWR@closedepthelevel\LWR@closedepthten
\let\LWR@closedepthten\LWR@closedepthnine
\let\LWR@closedepthnine\LWR@closedeptheight
\let\LWR@closedeptheight\LWR@closedepthseven
\let\LWR@closedepthseven\LWR@closedepthsix
\let\LWR@closedepthsix\LWR@closedepthfive
\let\LWR@closedepthfive\LWR@closedepthfour
\let\LWR@closedepthfour\LWR@closedepththree
\let\LWR@closedepththree\LWR@closedepthtwo
\let\LWR@closedepthtwo\LWR@closedepthone
\let\LWR@closedepthone#2
}
```

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

```
\newcommand*{\popclose}
{
\let\LWR@closeone\LWR@closetwo
\let\LWR@closetwo\LWR@closethree
\let\LWR@closethree\LWR@closefour
\let\LWR@closefour\LWR@closefive
\let\LWR@closefive\LWR@closesix
\let\LWR@closesix\LWR@closeseven
\let\LWR@closeseven\LWR@closeeight
\let\LWR@closeeight\LWR@closenine
\let\LWR@closenine\LWR@closeten
\let\LWR@closeten\LWR@closeeleven
\let\LWR@closeeleven\LWR@closetwelve
\let\LWR@closedepthone\LWR@closedepthtwo
\let\LWR@closedepthtwo\LWR@closedepththree
\let\LWR@closedepththree\LWR@closedepthfour
\let\LWR@closedepthfour\LWR@closedepthfive
\let\LWR@closedepthfive\LWR@closedepthsix
\let\LWR@closedepthsix\LWR@closedepthseven
\let\LWR@closedepthseven\LWR@closedeptheight
\let\LWR@closedeptheight\LWR@closedepthnine
\let\LWR@closedepthnine\LWR@closedepthten
\let\LWR@closedepthten\LWR@closedeptheleven
\let\LWR@closedeptheleven\LWR@closedepthtwelve
}
```

```
\end{warpHTML}
```

# 35 Data arrays

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

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

To assign an empty value:

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

**for HTML output:**

```
\begin{warpHTML}
```

`\LWR@setexparray` {⟨*name*⟩} {⟨*index*⟩} {⟨*contents*⟩}

```
\NewDocumentCommand{\LWR@setexparray}{m m m}{%
\ifstrempty{#3}%
{\csdef{#1#2}{}}%
{\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
}
```

`\LWR@getexparray` {⟨*name*⟩} {⟨*index*⟩}

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

```
\end{warpHTML}
```

## 36 HTML entities

**for HTML output:**

```
\begin{warpHTML}
```

HTML entites and HTML Unicode entities:

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

\HTMLentity {⟨*entitytag*⟩}

```
\newcommand*{\HTMLentity}[1]{%
% \LWR@traceinfo{HTMLentity \detokenize{#1}}%
\begingroup%
\LWR@FBcancel%
\LWR@origampersand#1;%
\endgroup
% \LWR@traceinfo{HTMLentity done}%
}
```

\HTMLunicode {⟨*hex_unicode*⟩}

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

\&

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

\textless
\textgreater

```
\let\LWR@origtextless\textless
\renewcommand*{\textless}{\HTMLentity{lt}}

\let\LWR@origtextgreater\textgreater
\renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
\end{warpHTML}
```

# 37 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:**

```
\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.

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

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 7.3.1.

```
\providecommand*{\HTMLFilename}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 7.3.1.

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

`\SetHTMLFileNumber` {⟨*number*⟩}

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

```
\newcommand*{\SetHTMLFileNumber}[1]{%
\setcounter{LWR@htmlfilenumber}{#1}%
}
```

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

Defaults to use section names in the filenames.

```
\newbool{FileSectionNames}
\booltrue{FileSectionNames}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

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

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

\LWR@htmlsectionfilename {⟨*htmlfilenumber or name*⟩}

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

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

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

```
\LWR@traceinfo{about to assign temp}%
\edef\LWR@tempone{#1}%
\LWR@traceinfo{about to compare with ??}%
\ifthenelse{\equal{\LWR@tempone}{??}}%
{\LWR@traceinfo{found ??}}%
{\LWR@traceinfo{not found ??}}%
\LWR@traceinfo{about to compare with zero or empty}%
\ifthenelse{%
    \equal{\LWR@tempone}{0}%
    \OR \equal{\LWR@tempone}{}%
    \OR \equal{\LWR@tempone}{??}%
}%
{%
    \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
    \HomeHTMLFilename.html%
}%
```

For a LaTeX section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```
{%
    \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
    \ifthenelse{%
        \equal{\HTMLFilename}{} \AND
        \equal{\LWR@tempone}{Index} \OR
        \equal{\LWR@tempone}{index}%
    }%
    {%
        \LWR@traceinfo{Prefixing the index name with an underscore.}%
        \_#1.html%
    }%
```

Otherwise, create a filename with the chosen prefix:

```
    {\HTMLFilename#1.html}%
}%
\LWR@traceinfo{LWR@htmlsectionfilename Z}%
}
```

`\LWR@htmlrefsectionfilename` {⟨*label*⟩}

Prints the filename for the given label

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

```
\end{warpHTML}
```

# 38 Homepage link

**for HTML output:**

```
\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.

```
\newcommand*{\LinkHome}{%
\LWR@subhyperrefclass{%
\HomeHTMLFilename.html}%
{Home}{linkhome}%
}
```

`\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.

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

`\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.

```
\newcommand*{\LWR@botnavigation}{
```

```
\LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}
}
```

```
\end{warpHTML}
```

## 39 \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:**

```
\begin{warpHTML}
```

`\PrintStack` Prints the closedepth stack.

```
\newcommand*{\PrintStack}{
\LWR@startpars
\LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
\LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
\LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
\LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
}
```

```
\end{warpHTML}
```

## 40 Closing stack levels

**for HTML output:**

```
\begin{warpHTML}
```

Close one nested level:

```
\newcommand*{\LWR@closeoneprevious}{%

\LWR@closeone

\popclose
}
```

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

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

Close any pending paragraph:

```
\LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
\whileboolexpr{not test{\ifnumcomp{\LWR@closedepthone}{<}{#1}}}%
    {\LWR@closeonepreviоus}
}
```

```
\end{warpHTML}
```

# 41 PDF pages and styles

**for HTML output:**

```
\begin{warpHTML}
```

`\LWR@forcenewpage` New PDF page a before major environment.

This is used just before major environments, such as `verse`. Reduces the chance of an environment overflowing the HTML PDF output page.

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

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

`\pagestyle` {⟨*style*⟩}

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

`\thispagestyle` {⟨*style*⟩}

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

`\markboth` {⟨*left*⟩} {⟨*right*⟩}

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

`\markright` {⟨*right*⟩}

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

\raggedbottom

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

\flushbottom

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

\sloppy

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

\fussy

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

\pagenumbering {⟨*commands*⟩}

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

```
\end{warpHTML}
```

# 42 HTML tags, spans, divs, elements

**for HTML output:**

```
\begin{warpHTML}
```

## 42.1 Mapping LaTeX Sections to HTML Sections

```
\newcommand*{\LWR@tagtitle}{h1}
\newcommand*{\LWR@tagtitleend}{/h1}
\newcommand*{\LWR@tagpart}{h2}
\newcommand*{\LWR@tagpartend}{/h2}
\newcommand*{\LWR@tagchapter}{h3}
\newcommand*{\LWR@tagchapterend}{/h3}
\newcommand*{\LWR@tagsection}{h4}
\newcommand*{\LWR@tagsectionend}{/h4}
\newcommand*{\LWR@tagsubsection}{h5}
\newcommand*{\LWR@tagsubsectionend}{/h5}
\newcommand*{\LWR@tagsubsubsection}{h6}
\newcommand*{\LWR@tagsubsubsectionend}{/h6}
\newcommand*{\LWR@tagparagraph}{span class="paragraph"{}}
```

```
\newcommand*{\LWR@tagparagraphend}{/span}
\newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{}}
\newcommand*{\LWR@tagsubparagraphend}{/span}

\newcommand*{\LWR@tagregularparagraph}{p}
```

## 42.2 Babel-French

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

(*Emulates or patches code by* Daniel Flipo.)

```
\providecommand*{\LWR@FBcancel}{}

\AtBeginDocument{%
\@ifundefined{frenchbsetup}%
{}%
{%
    \frenchbsetup{FrenchFootnotes=false}%
%
    \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
    \renewcommand*{\FBcolonspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;%
        \endgroup%
    }%
    \renewcommand*{\FBthinspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\#x202f;% \,
        \endgroup%
    }%
    \renewcommand*{\FBguillspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~,  for \og xyz \fg{}
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBmedkern}{}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\#x202f;% \,
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBthickkern}{}{%
```

```
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~
        \endgroup%
    }%
    \renewcommand*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
    \ifFBunicode%
    \else%
        \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
        \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
    \fi%
}%
}
```

## 42.3 HTML tags

\LWR@htmltagc {⟨*tag*⟩} Break ligatures and use upright apostrophes in HTML tags.

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

```
\newcommand*{\LWR@htmltagc}[1]{%
{%
% \LWR@traceinfo{LWR@htmltagc \detokenize{#1}}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\protect\LWR@origtextless#1\protect\LWR@origtextgreater%
\endgroup%
}%
}
```

Env LWR@nestspan Disable minipage, \parbox, and HTML <div>s inside a <span>.

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

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

```
\newcommand*{\LWR@nestspanitem}{%
\if@newlist\else{\LWR@htmltagc{br /}}\fi%
\LWR@origitem%
}

\newenvironment*{LWR@nestspan}
{%
\LWR@traceinfo{LWR@nestspan}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{%
    \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
}%
{% not in a lateximage
    \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
    \addtocounter{LWR@spandepth}{1}%
    \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
    \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
    \renewcommand{\BlockClassSingle}[2]{##2}%
    \renewcommand{\LWR@forcenewpage}{}%
    \renewcommand{\LWR@liststart}{%
        \let\item\LWR@nestspanitem%
    }%
    \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
}% not in a lateximage
}% starting env
{% ending env
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{\addtocounter{LWR@spandepth}{-1}}%
\LWR@traceinfo{LWR@nestspan: done}%
}

\AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}
```

\LWR@htmlspan {⟨*tag*⟩} {⟨*text*⟩}

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

```
\NewDocumentCommand{\LWR@htmlspan}{m +m}{%
\LWR@ensuredoingapar%
\LWR@htmltagc{#1}%
\begin{LWR@nestspan}%
#2%
\LWR@htmltagc{/#1}%
\end{LWR@nestspan}%
}
```

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

```
\NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
\LWR@traceinfo{LWR@htmlspanclass #2}%
\LWR@ensuredoingapar%
\LWR@subhtmlelementclass{span}[#1]{#2}%
\begin{LWR@nestspan}%
#3%
\LWR@htmltagc{/span}%
\LWR@traceinfo{LWR@htmlspanclass done}%
\end{LWR@nestspan}%
}
```

`\LWR@htmltag` {⟨*tag*⟩}

Print an HTML tag: <tag>

```
\newcommand*{\LWR@htmltagb}[1]{%
\LWR@htmltagc{#1}%
\endgroup%
}

\newcommand*{\LWR@htmltag}{%
\begingroup\catcode`\_=12
\LWR@htmltagb%
}
```

## 42.4 Block tags and comments

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

`\LWR@htmlopencomment`
`\LWR@htmlclosecomment`

```
\newcommand*{\LWR@htmlopencomment}{%
{%
% \LWR@traceinfo{LWR@htmlopencomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\LWR@origtextless{}!{-}{-}%
\endgroup%
}%
}

\newcommand*{\LWR@htmlclosecomment}{%
{%
% \LWR@traceinfo{LWR@htmlclosecomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
{-}{-}\LWR@origtextgreater{}%
\endgroup%
}%
}
```

\LWR@htmlcomment {⟨*comment*⟩}

```
\newcommand{\LWR@htmlcomment}[1]{%
\LWR@htmlopencomment{}%
{%
\LWR@origttfamily% break ligatures
#1%
}%
\LWR@htmlclosecomment{}}
```

\LWR@htmlblockcomment {⟨*comment*⟩}

```
\newcommand{\LWR@htmlblockcommentb}[1]
{\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}

\newcommand{\LWR@htmlblockcomment}
{%
\begingroup\catcode`\_=12
\LWR@htmlblockcommentb%
}
```

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

```
\newcommand*{\LWR@htmlblocktag}[1]{%
\LWR@stoppars%
\LWR@htmltag{#1}%
\LWR@startpars%
}
```

## 42.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
\NewDocumentCommand{\LWR@subhtmlelementclass}{m O{} m}{%
\LWR@traceinfo{LWR@subhtmlelementclass #1 #3}%
\ifblank{#2}%
{\LWR@htmltag{#1 class="#3"}}% empty option
{\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
\LWR@traceinfo{LWR@subhtmlelementclass done}%
}
```

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

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

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

```
\newcommand*{\LWR@htmlelementclassend}[2]{%
\LWR@stoppars%
\LWR@htmltag{/#1}%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{End of #1 ‘‘#2’’}%
}{}%
\LWR@startpars%
}
```

\LWR@htmldivclass [⟨*style*⟩] {⟨*class*⟩}

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

\LWR@htmldivclassend {⟨*class*⟩}

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

## 42.6 Single-line elements

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

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

```
\NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
\LWR@stoppars
\LWR@subhtmlelementclass{#1}[#2]{#3}%
#4%
\LWR@htmltag{/#1}
\LWR@startpars
}
```

## 42.7 HTML5 semantic elements

\LWR@htmlelement {⟨*element*⟩}

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

\LWR@htmlelementend {⟨*element*⟩}

```
\newcommand*{\LWR@htmlelementend}[1]{%
\LWR@stoppars
\LWR@htmltag{/#1}
\LWR@startpars
}

\end{warpHTML}
```

## 42.8 High-level block and inline classes

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

Nullified versions are provided for print mode.

For other direct-formatting commands, see section 77.

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

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

**for HTML output:**

```
\begin{warpHTML}
\NewDocumentEnvironment{BlockClass}{o m}%
{
\LWR@htmldivclass[#1]{#2}
}
{
\LWR@htmldivclassend{#2}
}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\NewDocumentEnvironment{BlockClass}{o m}{}{}%
\end{warpprint}
```

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

**for HTML output:**

```
\begin{warpHTML}
\newcommand{\BlockClassSingle}[2]{%
\LWR@htmlelementclassline{div}{#1}{#2}%
}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand{\BlockClassSingle}[2]{#2}
\end{warpprint}
```

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

**for HTML output:**

```
\begin{warpHTML}
\NewDocumentCommand{\InlineClass}{o m +m}{%
\LWR@htmlspanclass[#1]{#2}{#3}%
}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\NewDocumentCommand{\InlineClass}{o m +m}{#3}%
\end{warpprint}
```

Env `LWR@BlockClassWP` {⟨*WPstyle*⟩} {⟨*HTMLstyle*⟩} {⟨*class*⟩} Low-level interface for `<div>` classes with an automatic float ID. These are often used when `\ifbool{FormatWP}`.

**for HTML output:**

```
\begin{warpHTML}
\NewDocumentEnvironment{LWR@BlockClassWP}{m m m}%
{%
\LWR@stoppars%
\ifbool{FormatWP}%
{%
    \LWR@maybeincthisautoid%
    \LWR@htmltag{%
        div class="#3" %
        id="autoid-\arabic{LWR@thisautoid}"%
        \ifblank{#1}{}{ style="#1"}%
    }%
}% FormatWP
{% not FormatWP
```

```
    \LWR@htmltag{%
        div class="#3"%
        \ifblank{#2}{}{ style="#2"}%
    }%
}% not FormatWP
\LWR@startpars%
}
{\LWR@htmldivclassend{#3}}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{}{}%
\end{warpprint}
```

## 42.9 Closing HTML tags

**for HTML output:**

```
\begin{warpHTML}
```

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

```
\newcommand*{\LWR@printclosepart}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
\newcommand*{\LWR@printclosechapter}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
\newcommand*{\LWR@printclosesection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
\newcommand*{\LWR@printclosesubsection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
\newcommand*{\LWR@printclosesubsubsection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
\newcommand*{\LWR@printcloseparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
\newcommand*{\LWR@printclosesubparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
\newcommand*{\LWR@printcloselistitem}
    {\LWR@htmltag{/li}}
\newcommand*{\LWR@printclosedescitem}
    {\LWR@htmltag{/dd}}
\newcommand*{\LWR@printcloseitemize}
    {\LWR@htmltag{/ul}}
\newcommand*{\LWR@printcloseenumerate}
    {\LWR@htmltag{/ol}}
\newcommand*{\LWR@printclosedescription}
```

```
    {\LWR@htmltag{/dl}}
```

```
\end{warpHTML}
```

# 43 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 44 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

**for HTML output:**

```
\begin{warpHTML}
```

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

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

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

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

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

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

```
\newcommand*{\LWR@ensuredoingapar}{%
\ifbool{LWR@doingstartpars}%
{\global\booltrue{LWR@doingapar}}%
{}%
}
```

`\LWR@openparagraph`

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

See if paragraph handling is enabled:

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

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

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }% 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>`.

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

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

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

Manually indent item list labels to avoid left margin intrustion:

See if are nested inside an item list:

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

If so, leave some horizontal room in the LaTeX PDF output for list labels:

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

Now have started a paragraph.

```
        \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.)

```
        \let\par\LWR@closeparagraph%
    }% end of yes nest par tags
}% end of handling pars
{}% not handling pars
}
```

`\LWR@closeparagraph`

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

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
{% handling pars
```

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

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }%
```

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

```
    {% no nested par tags
        \ifboolexpr{
            test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
            test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
        }%
        {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
        {}%
    }% no nested par tags
```

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

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

Print a closing tag and some extra vertical space:

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer doing a paragraph:

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

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

```
{% not handling pars
    \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
    {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
    {}%
}% not handling pars
```

Finish with regular paragraph processing

```
\LWR@origpar%
}
```

```
\end{warpHTML}
```

# 44 Paragraph start/stop handling

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

Section 43 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:**

```
\begin{warpHTML}
```

\LWR@startpars

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

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

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
    {}%
```

If not currently in paragraph mode:

```
    {%
```

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

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

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

```
        \let\par\LWR@closeparagraph

    }% an intentionally blank line
```

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

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

No <par> tag yet to undo:

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

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

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

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
    {%
```

Print a closing tag:

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer have an open HTML paragraph:

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

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

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

    }% an intentionally blank line
```

If was not in an HTML paragraph:

```
    {}%
```

See if currently allowing HTML paragraphs:

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

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

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

Else: do nothing

```
    {}%
```

no longer in paragraph mode

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

no <p> tag to undo:

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

```
\end{warpHTML}
```

# 45 Page headers and footers

for HTML & PRINT:

```
\begin{warpall}
```

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

```
\newcommand{\LWR@firstpagetop}{} % for the home page alone
\newcommand{\LWR@pagetop}{} % for all other pages
\newcommand{\LWR@pagebottom}{}

\newcommand{\LWR@setfirstpagetopb}[1]{%
\renewcommand{\LWR@firstpagetop}{#1}
\catcode`\_=8
}
```

\HTMLFirstPageTop {⟨*text and logos*⟩}

```
\newcommand{\HTMLFirstPageTop}{%
\catcode`\_=12
\LWR@setfirstpagetopb
}
```

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

\HTMLPageTop {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageTop}{%
\catcode`\_=12
\LWR@setpagetopb
}
```

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

\HTMLPageBottom {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageBottom}{%
\catcode`\_=12
\LWR@setpagebottomb
}
```

```
\end{warpall}
```

# 46 CSS

**for HTML output:**

```
\begin{warpHTML}
```

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

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

\CSSFilename {⟨*new-css-filename.css*⟩} Assigns the CSS file to be used by the following HTML pages.

```
\newcommand*{\LWR@newcssb}[1]{%
\renewcommand*{\LWR@currentcss}{#1}
\catcode`\_=8
}

\newcommand*{\CSSFilename}{
\catcode`\_=12
\LWR@newcssb
}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand*{\CSSFilename}[1]{}
\end{warpprint}
```

## 47 Title, HTML meta author, HTML meta description

**for HTML output:**

```
\begin{warpHTML}
```

`\title` {⟨*title*⟩} Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
\let\LWR@origtitle\title

\renewcommand*{\title}[1]{%
    \LWR@origtitle{#1}%
    \begingroup%
        \renewcommand{\thanks}[1]{}%
        \protected@xdef\thetitle{#1}%
    \endgroup%
}
```

```
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

`\HTMLTitle` {⟨*Titlename*⟩} The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
\providecommand{\thetitle}{}

\newcommand{\theHTMLTitle}{\thetitle}

\newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor` {⟨*authorname*⟩} The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the titling package is used.

```
\providecommand{\theauthor}{}

\newcommand{\theHTMLAuthor}{\theauthor}

\newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

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

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

\HTMLDescription {⟨*New HTML meta description.*⟩} Assigns the HTML file's description meta tag.

```
\newcommand{\LWR@currentHTMLDescription}{}

\newcommand{\HTMLDescription}[1]{%
\renewcommand{\LWR@currentHTMLDescription}{#1}
}

\end{warpall}
```

# 48 Footnotes

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

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

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

## 48.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the LaTeX box \LWR@footnotes. Using this instead of the original \footins box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 48.4 for the implementation.

## 48.2 Minipage footnotes

See section 48.5 for how minipage footnotes are gathered. See section 76.3 for how minipage footnotes are placed into the document.

## 48.3 Titlepage thanks

See section 55.6 for titlepage footnotes.

## 48.4 Regular page footnote implementation

**for HTML & PRINT:**

```
\begin{warpall}
```

Ctr FootnoteDepth
Default: 5

Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`. The default of 5 places footnotes before each `\subparagraph` or higher. See table 7 for a table of LaTeX section headings.

```
\newcounter{FootnoteDepth}
\setcounter{FootnoteDepth}{5}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

Patch LaTeX footnotes to use a new `\box` for lwarp footnotes.

```
\newbox\LWR@footnotes
```

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

`\@makefntext` {⟨*text*⟩}

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

`\@makefnmark`

```
\def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

Footnotes may be in regular text, in which case paragraphs are tagged, or in a table data cell or `lateximage`, in which case paragraph tags must be added manually.

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

`\LWR@footnotetext` {⟨*text*⟩}

```
\long\def\LWR@footnotetext#1{%
\LWR@traceinfo{LWR@footnotetext}%
\global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
    \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
    \protected@edef\@currentlabel{%
        \csname p@footnote\endcsname\@thefnmark%
    }% @currentlabel
```

Open a group:

```
    \color@begingroup%
```

Use HTML superscripts even inside a `lateximage`:

```
    \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
    \@makefntext{#1}%
```

Closing paragraph tag:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {\par}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
```

Close the group:

```
    \color@endgroup%
}% vbox
}%
```

\@footnotetext {⟨*text*⟩}

```
\let\@footnotetext\LWR@footnotetext
```

## 48.5 Minipage footnote implementation

\@mpfootnotetext {⟨*text*⟩}

```
\long\def\@mpfootnotetext#1{%
\global\setbox\@mpfootins\vbox{%
    \unvbox\@mpfootins
    \reset@font\footnotesize
    \hsize\columnwidth
    \@parboxrestore
    \protected@edef\@currentlabel
    {\csname p@mpfootnote\endcsname\@thefnmark}%
    \color@begingroup
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%

    \@makefntext{%
        \ignorespaces#1%
    }%
```

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

```
    \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
    {}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
    \color@endgroup%
}% vbox
}
```

### 48.6 Printing pending footnotes

\LWR@printpendingfootnotes Enclose the footnotes in a class, print, then clear.

```
\newcommand*{\LWR@printpendingfootnotes}{%
\ifvoid\LWR@footnotes\else
    \LWR@forcenewpage
    \begin{BlockClass}{footnotes}
    \LWR@origmedskip
    \unvbox\LWR@footnotes
    \setbox\LWR@footnotes=\vbox{}
    \end{BlockClass}
\fi
}
```

\LWR@maybeprintpendingfootnotes {⟨*depth*⟩} Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
\newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
\ifboolexpr{
    not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
    bool{FormatEPUB} or
    bool{FormatWP}
}%
{\LWR@printpendingfootnotes}%
{}%
}
```

```
\end{warpHTML}
```

## 49 Marginpars

\marginpar [⟨*left*⟩] {⟨*right*⟩} \marginpar may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the \marginpar. Paragraph breaks are converted to <br /> tags.

\marginparBlock [⟨*left*⟩] {⟨*right*⟩} To include block-related macros, use \marginparBlock, which takes the same arguments but creates a <div> instead of a <span>. A line break will occur in the text where the \marginBlock occurs.

**for HTML output:**

```
\begin{warpHTML}
```

\marginpar [⟨*left*⟩] {⟨*right*⟩}

```
\renewcommand{\marginpar}[2][]{%
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
    \LWR@htmlspanclass{marginpar}{#2}%
}%
}
```

\marginparBlock [⟨*left*⟩] {⟨*right*⟩}

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```
\newcommand{\marginparBlock}[2][]{%
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
\begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
#2
\end{BlockClass}
}%
}
```

```
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
```

\marginparBlock [⟨*left*⟩] {⟨*right*⟩}

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

Print version.

```
\LetLtxMacro\marginparBlock\marginpar
```

```
\end{warpprint}
```

# 50 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:**

```
\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.

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

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

```
\newbool{CombineHigherDepths}
\booltrue{CombineHigherDepths}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

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

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

`\LWR@thisnewfilename` The filename being sanitized.

```
\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` LaTeX commands which appear in section names and TOC captions.

```
\newcommand*{\LWR@filenamenoblanks}[1]{%
\begingroup
```

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

```
\LWR@nullfonts
\renewcommand*{\LWR@htmltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
\edef\LWR@thisnewfilename{#1}
\fullexpandarg
```

Convert spaces into hyphens:

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

Convert punctutation into hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{’}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\LWR@origampersand}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textless}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textgreater}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textbackslash}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{~{}}{-}[\LWR@thisnewfilename]
%     "~{}" for babel
\StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{‘}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
\StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
%     emdash
\StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
%     endash
\global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
\endgroup
}
```

\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.

```
\newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```
\LWR@htmlelementclassend{section}{textbody}

\LWR@printpendingfootnotes

```

No footer between files if EPUB:

```
\ifbool{FormatEPUB}
{}
{
    \LWR@htmlelement{footer}

    \LWR@pagebottom

    \LWR@htmlelementend{footer}
}
```

No bottom navigation if are finishing the home page or formatting for EPUB or a word-processor.

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
\LWR@stoppars
\LWR@htmltag{/body}\LWR@orignewline
\LWR@htmltag{/html}\LWR@orignewline
\LWR@orignewpage

\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.

```
\ifbool{FileSectionNames}%
{\LWR@filenamenoblanks{#1}}
{\renewcommand*{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}}
```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
\LWR@htmlblockcomment{%
|Start file|%
\LWR@htmlsectionfilename{\LWR@thisfilename}|%
}
```

At the top of the starting file:

```
\LWR@stoppars
```

```

\LWR@filestart{ -- #1}% there is an EMdash in front of the #1

```

No navigation between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@topnavigation}

```

No header if between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \LWR@htmlelement{header}

    \LWR@pagetop

    \LWR@htmlelementend{header}
}

```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\ifcsvoid{thetitle}{}{\LWR@printthetitle}}

```

No sideTOC if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@sidetoc}

```

Start of the `<textbody>`:

```
\LWR@htmlelementclass{section}{textbody}

```

Keep paragraph tags disabled for now:

```
\LWR@stoppars

```

Track the page numbers:

```
\setcounter{LWR@latestautopage}{\value{page}}
}
```

```
\end{warpHTML}
```

# 51 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing LaTeX 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.

XeLaTeX and LuaLaTeX directly support accented section and file names.

**for HTML output:**

```
\begin{warpHTML}
```

## 51.1 User-level starred section commands

\ForceHTMLPage For HTML output, forces the next section to be on its own HTML page, if FileDepth allows, even if starred. For use with \printindex and others which generate a starred section which should be on its own HTML page. Also see \ForceHTMLTOC.

For print output, no effect.

```
\newbool{LWR@forcinghtmlpage}
\boolfalse{LWR@forcinghtmlpage}

\newcommand*{\ForceHTMLPage}{%
\global\booltrue{LWR@forcinghtmlpage}%
}
```

\ForceHTMLTOC For HTML output, forces the next section to have a TOC entry, even if starred. For use with \printindex and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with tocbibind. Also see \ForceHTMLPage.

For print output, no effect.

```
\newbool{LWR@forcinghtmltoc}
\boolfalse{LWR@forcinghtmltoc}

\newcommand*{\ForceHTMLTOC}{%
\global\booltrue{LWR@forcinghtmltoc}%
}
```

```
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand*{\ForceHTMLPage}{}
\newcommand*{\ForceHTMLTOC}{}
\end{warpprint}
```

**for HTML output:**

```
\begin{warpHTML}
```

## 51.2 Book class commands

`\mainmatter` ⚠ Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
\newbool{LWR@mainmatter}
\DeclareDocumentCommand{\mainmatter}{}{%
\booltrue{LWR@mainmatter}%
}
```

`\frontmatter` Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
\DeclareDocumentCommand{\frontmatter}{}{%
\boolfalse{LWR@mainmatter}%
}
```

`\backmatter` Declare the back matter section of the document. Does not reset the page number.

```
\DeclareDocumentCommand{\backmatter}{}{%
\boolfalse{LWR@mainmatter}
}
```

### 51.3 Sectioning support macros

\LWR@sectionnumber {⟨*section type*⟩}

Typeset a section number and its trailing space with CSS formatting:

```
\newcommand*{\LWR@sectionnumber}[1]{%
\InlineClass{sectionnumber}{#1}%
}
```

autosec A tag used by the TOC and index.

\LWR@createautosec {⟨*section type*⟩}

Create an autosection tag.

```
\newcommand*{\LWR@createautosec}[1]{%
\LWR@htmltag{#1 id="autosec-\thepage"{}}%
}
```

\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.

```
\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.

```
\NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
\LWR@closeprevious{#1}%
```

Push a new section depth:

```
\LWR@pushoneclose{#1}{#2}%
}
```

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.

```
\newcounter{LWR@prevFileDepth}
\setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

\@seccntformat {⟨*sectiontype*⟩}

```
\def\@seccntformat#1{\csname the#1\endcsname\protect\quad}
```

\simplechapterdelim Used by tocbibind and anonchap.

```
\newcommand*{\simplechapterdelim}{}
```

\@chapcntformat {⟨*sectiontype*⟩}

`\let` to `\@seccntformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from tocbibind or anonchap.

```
\let\@chapcntformat\@seccntformat
```

Ctr LWR@currentautosec Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following autosec label is created.

```
\newcounter{LWR@currentautosec}
```

\LWR@section * [⟨*TOC name*⟩] {⟨*name*⟩} {⟨*sectiontype*⟩}

The common actions for the high-level sectioning commands.

```
\DeclareDocumentCommand{\LWR@section}{m m m m}{%
\LWR@traceinfo{LWR@section |#2| |#3|}%
\LWR@traceinfo{LWR@section: not an empty section}%
\LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
\global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for `\part*`, for appendix.

Generate a new LaTeX page so that TOC and index page number points to the section:

```
\LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
\IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{}%
\ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
\ifthenelse{%
    \(%
        \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
        \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
        \(\boolean{LWR@forcinghtmlpage}\)%
    \)%
    \AND%
    \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
    \AND%
    \(%
        \NOT\boolean{CombineHigherDepths}\OR%
        \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
    \)%
    \AND%


    \NOT\isempty{#3}% phantomsection
}%
```

If so: start a new HTML file:

```
{% new file
    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
    \IfNoValueTF{#2}%
```

If no optional entry

```
    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
    {\LWR@newhtmlfile{#2}}%
}% new file
```

Else: No new HTML file:

```
{% not new file
```

Generate a new LaTeX page so that TOC and index page number points to the section:

```
    \LWR@traceinfo{LWR@section: not a new HTML file}%
    \LWR@orignewpage%

}% not new file
```

Remember this section’s name for `\nameref`:

```
\IfValueT{#3}{%
\LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
\IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
}%
```

Print an opening comment with the level and the name; ex: “section” “Introduction”

```

\ifbool{HTMLDebugComments}{%
    \begingroup%
    \LWR@nullfonts%
    \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}%
    \endgroup%
}{}%

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
\ifthenelse{%
    \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

Create the opening tag with an autosec:

```
\LWR@createautosec{\csuse{LWR@tag#4}}%
```

```
\setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
\IfBooleanTF{#1}%
{%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
\ifbool{LWR@forcinghtmltoc}%
{\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
{}%
}% starred
```

Not starred, so step counter and add to TOC:

```
{% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
    \ifthenelse{%
        \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
    }%
    {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For `article` class, lwarp assumes that all is mainmatter.

```
        \LWR@traceinfo{LWR@section: about to test main matter}%
        \ifbool{LWR@mainmatter}%
        {%
            \LWR@traceinfo{LWR@section: yes mainmatter}%
            \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
            \LWR@traceinfo{LWR@section: about to addcontentsline}%
            \addcontentsline{toc}{#4}%
            {%
                 \protect\numberline{\csuse{the#4}}%
                {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
            }%
            \LWR@traceinfo{LWR@section: finished addcontentsline}%
        }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
        {% not main matter
            \LWR@traceinfo{LWR@section: no main matter}%
            \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
        }% end of not main matter
    }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
    {%
        \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
    }%
```

For part, print the section type:

```
    \ifbool{LWR@mainmatter}%
    {%
        \ifthenelse{%
            \(\cnttest{\csuse{LWR@depth#4}}{<=}%
                {\value{secnumdepth}}\) \AND%
            \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)%
        }%
        {\csuse{#4name}~{}}%
        {}%
```

Print the section number:

```
        \LWR@traceinfo{LWR@section: about to print section number}%
        \ifthenelse{%
            \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
        }%
        {%
            \ifstrequal{#4}{chapter}%
            {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
            {\protect\LWR@sectionnumber{\@seccntformat{#4}}}%
        }%
        {}%
        \LWR@traceinfo{LWR@section: finished print section number}%
    }{}%
}% end of not starred
```

Print the section name:

```
#3%
```

Close the heading tag, such as /H2:

```
\LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a LaTeX label:

```
\label{autopage-\theLWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
\ifthenelse{%
    \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

If not starred, remember the previous depth to possibly trigger a new HTML page.

HOWEVER, allow a `\part*` to start a new HTML page. This is used by appendix.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a `\listoftables` was tested in the middle of the document. The `\chapter*` for the list was not allowing a new HTML page for the section following it while `CombineHigherDepths` was true.

```
\ifthenelse{%
    \NOT\equal{#1}{\BooleanTrue}\OR%
    \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
}%
{% not starred
    \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
}% not starred
{}%
```

Reset to defaults if not a phantomsection:

```
\ifstrempty{#3}%
{}%
{%
\global\boolfalse{LWR@forcinghtmlpage}%
\global\boolfalse{LWR@forcinghtmltoc}%
}%
%
\LWR@traceinfo{LWR@section: done}%
}
```

## 51.4 \section and friends

`\part` * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\newcommand{\part@preamble}{}% for koma-script

\DeclareDocumentCommand{\part}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%

\LWR@section{#1}{#2}{#3}{part}%

\part@preamble% for koma-script
\renewcommand{\part@preamble}{}%
}
```

\chapter * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\let\@printcites\relax% for quotchap package

\newcommand{\chapter@preamble}{}% for koma-script

\@ifundefined{chapter}
{}
{%
\DeclareDocumentCommand{\chapter}{s o m}{%
\LWR@traceinfo{chapter #3}%
\LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%

\LWR@section{#1}{#2}{#3}{chapter}%

\@printcites% for quotchap package

\chapter@preamble% for koma-script
\renewcommand{\chapter@preamble}{}%
}
}
```

\section * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\section}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%

\LWR@section{#1}{#2}{#3}{section}%
}
```

\subsection * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\subsection}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%

\LWR@section{#1}{#2}{#3}{subsection}%
}
```

`\subsubsection` * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\subsubsection}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsubsubsection}%
{\LWR@printclosesubsubsection}%

\LWR@section{#1}{#2}{#3}{subsubsection}%
}
```

`\paragraph` * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\paragraph}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%

\LWR@section{#1}{#2}{#3}{paragraph}%
}
```

`\subparagraph` * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\subparagraph}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%

\LWR@section{#1}{#2}{#3}{subparagraph}%
}
```

```
\end{warpHTML}
```

## 52 Starting a new file

**for HTML & PRINT:**

```
\begin{warpall}
```

`\HTMLLanguage` Default language for the HTML `lang` tag.

```
\newcommand*{\LWR@currentHTMLLanguage}{en-US}

```

```
\newcommand*{\HTMLLanguage}[1]{%
\renewcommand*{\LWR@currentHTMLLanguage}{#1}%
}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

`\LWR@filestart` {⟨*title_suffix*⟩}

Creates the opening HTML tags.

```
\newcommand*{\LWR@filestart}[1]{
\LWR@traceinfo{LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
\begingroup
\LWR@traceinfo{LWR@filestart: A}
\LWR@nullfonts
\LWR@traceinfo{LWR@filestart: B}
```

Create the page's HTML header:

```
\LWR@htmltag{!DOCTYPE html}\LWR@orignewline
\LWR@traceinfo{LWR@filestart: C}
```

The language is user-adjustable:

```
\LWR@htmltag{html lang="\LWR@currentHTMLLanguage"{}}\LWR@orignewline
```

Start of the meta data:

```
\LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
\LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
\ifthenelse{\equal{\theHTMLAuthor}{}}%
{}%
{\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline}%
```

lwarp is the generator:

```
\LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
    \LWR@orignewline
```

If there is a description, add it now:

```
\ifdefempty{\LWR@currentHTMLDescription}{}{%
\LWR@htmltag{%
meta name="description" content="\LWR@currentHTMLDescription" /}%
    \LWR@orignewline
}%
```

Mobile-friendly viewport:

```
\LWR@htmltag{meta name="viewport" %
content="width=device-width, initial-scale=1.0" /}%
    \LWR@orignewline
```

IE patch:

```
\LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
\LWR@htmltag{%
script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}}%
\LWR@htmltag{/script}\LWR@orignewline
\LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
\ifthenelse{\equal{\theHTMLTitle}{}}%
{}%
{\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%
```

The page's stylesheet:

```
\LWR@htmltag{%
link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
\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.

```
\ifbool{mathjax}%
{%
    \begingroup%
    \LWR@restoreoriglists%
    \boolfalse{LWR@verbtags}
    \verbatiminput{lwarp_mathjax.txt}%
    \booltrue{LWR@verbtags}
    \endgroup%
    \LWR@stoppars
}% end of mathjax
{}%
```

End of the header:

```
\LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
\LWR@htmltag{body}\LWR@orignewline
\endgroup
\LWR@traceinfo{LWR@filestart: done}
}
```

```
\end{warpHTML}
```

## 53 Starting HTML output

**for HTML output:**

```
\begin{warpHTML}
```

\LWR@LwarpStart

Executed at the beginning of the entire document.

```
\catcode`\$=\active
\newcommand*{\LWR@LwarpStart}
{%
\LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
\ifbool{FormatWP}{%
    \setcounter{FileDepth}{-5}%
    \boolfalse{HTMLDebugComments}%
}{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
\edef\LWR@strresult{\HomeHTMLFilename}
\edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
\edef\LWR@strresult{\HTMLFilename}
\edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
\LWR@origonecolumn%
\LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
\LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
\LWR@origraggedright%
\LetLtxMacro{\\}{\LWR@endofline}%
```

Spread the lines for `pdftotext` to read them well:

```
\linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
\immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
\setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
\renewcommand{\ps@plain}{}
```

\centering
\raggedleft
\raggedright

Not used in the HTML environment:

```
\renewcommand*{\centering}{}
\renewcommand*{\raggedleft}{}
\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:

```
\LetLtxMacro{\LWR@origtabular}{\tabular}
\LetLtxMacro{\LWR@origendtabular}{\endtabular}
```

```
\LetLtxMacro{\tabular}{\LWR@tabular}
\LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
\let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in amsmath environments and is also patched by cleveref.

Label in HTML

```
\let\LWR@origltx@label\ltx@label
\let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
\let\TeX\LWR@TeX
\let\LaTeX\LWR@LaTeX
\let\LuaTeX\LWR@LuaTeX
\let\LuaLaTeX\LWR@LuaLaTeX
\let\XeTeX\LWR@XeTeX
\let\XeLaTeX\LWR@XeLaTeX
\let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
\global\boolfalse{LWR@doingapar}
\global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
\LWR@traceinfo{LWR@lwarpStart: Starting new file.}
\LWR@filestart{}
\LWR@traceinfo{LWR@lwarpStart: Generating first header.}
\LWR@htmltag{header}\LWR@orignewline
\LWR@startpars
\LWR@firstpagetop
\LWR@stoppars
\LWR@htmltag{/header}\LWR@orignewline
\LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
\LWR@htmltag{section class="textbody"{}}
```

Document and page settings:

```
\mainmatter
\LWR@origpagenumbering{arabic}
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native LaTeX environments, as well as those provided by enumitem, enumerate, and paralist.

```
\LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
\catcode`\$=\active
```

Allow HTML paragraphs to begin:

```
\LWR@startpars
\LWR@traceinfo{LWR@lwarpStart: done}
}
\catcode`\$=3% math shift until lwarp starts
```

```
\end{warpHTML}
```

## 54 Ending HTML output

**for HTML output:**

```
\begin{warpHTML}
```

`\LWR@requesttoc` {⟨*boolean*⟩} {⟨*suffix*⟩} Requests that a toc, lof, or lot be generated.

```
\newcommand*{\LWR@requesttoc}[2]{%
\ifbool{#1}
{
    \expandafter\newwrite\csuse{tf@#2}
    \immediate\openout \csuse{tf@#2} \jobname.#2\relax
}{}
}
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
\newcommand*{\LWR@LwarpEnd}
{
\LWR@stoppars
\LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
\LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
\LWR@printpendingfootnotes
```

Create the footer:

```
\LWR@htmlelement{footer}

\LWR@pagebottom

\LWR@htmlelementend{footer}
```

No bottom navigation if are finishing the home page, or if formatting for an EPUB or word processor.

Presumably has a table-of-contents.

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
}
```

```
\LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
\LWR@htmltag{/body}\LWR@orignewline
\LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
\LWR@orignewpage
```

For `lateximage` commands:

```
\immediate\closeout\LWR@lateximagesfile
}
```

```
\end{warpHTML}
```

## 55 Title page

package support

lwarp supports the native LaTeX titling commands, and also supports the packages

⚠ load order

authblk and titling. If both are used, authblk should be loaded before titling.

\published and \subtitle

If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 55.7.

affiliation

lwarp provides for the \author macro an additional \affiliation macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using titlingpage's \theauthor in the main text.

reusing titlepage information

⚠ \theauthor, authblk

The titling package maintains the definitions of \thetitle, \theauthor, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the \thanks and \affiliation removed, and for \author the \and is replaced to generate a simple inline list of authors separated by commas. Note: \theauthor does not work well with authblk unless the traditional LaTeX syntax is used.

custom titlepages

\printtitle, \printauthor, etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the \thanks and \affiliation.

\printthanks

\printthanks has been added to force the printing of thanks inside a titlingpage environment when \maketitle is not used.

⚠

Inside a \titlepage or \titlingpage environment, use \thanks instead of \footnote for acknowledgements, etc.

## 55.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

\author
\and

{⟨*author*⟩} While using \maketitle and print mode, the author is treated as a single-column tabular and the \and feature finishes the current tabular then starts a new one for the next author. Each author thus is placed into its own tabular, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a <div> of class author, and each individual author is inside a <div> of class oneauthor.

\@title
\@author
\@date

\@title, \@author, etc. store the values as originally assigned, including any \thanks, \and, or \affiliation. These are low-level macros intended to be used by other macros only inside a titlepage or titlingpage, and are used by \maketitle. The author is printed inside a single-column tabular, which becomes multiple single-column tabulars if multiples authors are included. For HTML these tabulars become side-by-side <div>s of class oneauthor, all of which are combined into one <div> of class author.

\printtitle
\printauthor
\printdate

\printtitle, etc. are user-level macros intended to be used in custom titlepage or titlingpage environments in cases where \maketitle is not desired. These commands preserve the \thanks, etc., and should not be used in the main text.

\thetitle
\theauthor
\thedate

\thetitle, \theauthor, and \thedate are available if titling has been loaded, and are sanitized user-level versions from which have been removed the \thanks and \affiliation, and \and is changed for inline text usage. The author is printed inline without \affiliation or \thanks, with \and placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

\HTMLPageBottom {⟨*text*⟩}

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

⚠ \theauthor — \theauthor does not work well if authblk is used. If \theauthor is important, it is recommended to use the standard LaTeX syntax for \author, optionally with lwarp's \affiliation macro as well.

⚠ affiliations — After \maketitle has completed, \theauthor retains the definition of the author, but \and is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

\affiliation — A solution, provide here, is to define a macro \affiliation which, during \maketitle, starts a new row and adds the affiliation, but after \maketitle is finished \affiliation is re-defined to discard its argument, thus printing only the author names when \author is later used inline.

## 55.2 Changes for \affiliation

\affiliation {⟨*text*⟩}

Adds the affiliation to the author for use in \maketitle.

Inside titlepage, this macro prints its argument. Outside, it is null.

**for HTML & PRINT:**

```
\begin{warpall}
\newrobustcmd{\affiliation}[1]{}
\end{warpall}
```

**for PRINT output:**

```
\begin{warpprint}
```

```
\AtBeginEnvironment{titlepage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}

\AtBeginDocument{
\@ifpackageloaded{titling}{
\AtBeginEnvironment{titlingpage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}
}{}% titling loaded
}% AtBeginDocument
```

```
\end{warpprint}
```

**for HTML output:**

```
\begin{warpHTML}
```

Env `titlepage` Sets up a `<div>` of class `titlepage`.

```
\renewenvironment*{titlepage}
{
\renewrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{##1}}
\LWR@printpendingfootnotes
\LWR@forcenewpage
\BlockClass{titlepage}
}
{
\endBlockClass
\LWR@printpendingfootnotes
}
```

```
\end{warpHTML}
```

## 55.3 Printing the thanks

**for HTML & PRINT:**

```
\begin{warpall}
```

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
\newcommand*{\printthanks}{\@thanks}
```

```
\end{warpall}
```

## 55.4 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

**for HTML output:**

```
\begin{warpHTML}
```

`\printtitle`

```
\newcommand*{\printtitle}
{
\LWR@stoppars
\LWR@htmltag{\LWR@tagtitle}%
\@title%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
\newcommand*{\LWR@printthetitle}
{
\LWR@stoppars
\LWR@htmltag{\LWR@tagtitle}%
\thetitle%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

`\printauthor` HTML version.

```
\newcommand*{\printauthor}{
```

The entire author block is contained in a `<div>` named `author`:

```
\begin{BlockClass}{author}
```

`\and` finishes one author and starts the next:

```
\renewcommand{\and}{%
\end{BlockClass}
\begin{BlockClass}{oneauthor}
}
```

Individual authors are contained in a `<div>` named `oneauthor`:

```
\begin{BlockClass}{oneauthor}
\@author
\end{BlockClass}
\end{BlockClass}
}
```

\printdate

```
\newcommand*{\printdate}{%
\begin{BlockClass}{titledate}
\@date
\end{BlockClass}
}
```

```
\end{warpHTML}
```

## 55.5 Printing the title, etc. in print form

The following are for printing the title, etc. in a titlepage or a titlingpage in print form:

**for PRINT output:**

```
\begin{warpprint}
```

\printtitle

```
\newcommand*{\printtitle}{{\Huge\@title}}
```

\printauthor Print mode.

```
\newcommand*{\printauthor}
    {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

\printdate

```
\newcommand*{\printdate}{{\small\textit{\@date}}}
```

```
\end{warpprint}
```

## 55.6 \maketitle for HTML output

An HTML <div> of class titlepage is used.

\thanks are a form of footnotes used in the title page. See section 48 for other kinds of footnotes.

See \thanksmarkseries{series}, below, to set the style of the footnote marks.

**for HTML output:**

```
\begin{warpHTML}
```

```
\if@titlepage
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@arabic\c@footnote}%
}
\else
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
}
\fi
```

\LWR@maketitlesetup Patches \thanks macros.

```
\newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\LWR@setfootnoteseries%
\def\@makefnmark{\textsuperscript{\thefootnote}}
```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\textsuperscript{\@thefnmark} \LWR@orighspace{1in}
```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

Print the text:

```
##1%
}%
}
```

\@fnsymbol {⟨*counter*⟩}

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being seen by `pdftotext`.

```
\def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
   \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
   **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
   \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}
```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
\begin{titlepage}
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any `\thanks` footnotes:

```
\@thanks
```

Close the HTML titlepage div and cleanup:

```
\end{titlepage}
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@maketitle\relax
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@date\@empty
\global\let\@title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}
```

\@maketitle HTML mode. Typesets the title, etc.:

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
    \@title
    \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    \begin{BlockClass}{author}
        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
            \@author
        \end{BlockClass}
    \end{BlockClass}
    \begin{BlockClass}{titledate}
    \@date
    \end{BlockClass}
}
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
\newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
\@thanks
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
}
```

```
\end{warpHTML}
```

### 55.7 \published and \subtitle

\subtitle and \published To add \subtitle and \published to the titlepage, load the titling package and use \AddSubtitlePublished in the preamble.

The default lwarp.css has definitions for the published and subtitle classes.

⚠ titling hooks

After titling is loaded, \AddSubtitlePublished is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks. Do not use \AddSubtitlePublished if the user has patched the titling hooks for some other reason. Portions are marked \warpprintonly to reduce extra tags in HTML. Similarly, BlockClass has no effect in print mode. Thus, the following may be marked warpall.

**for HTML & PRINT:**

```
\begin{warpall}
```

\AddSubtitlePublished Adds \published and \subtitle, and related.

```
\AfterPackage{titling}{
\newcommand*{\AddSubtitlePublished}{%

\newcommand{\@published}{}

\newcommand{\published}[1]{\gdef\@published{##1}}

\renewcommand*{\maketitlehooka}{\printpublished}

\newcommand*{\printpublished}{%
\warpprintonly{\begin{center}\unskip}%
\begin{BlockClass}{published}%
\warpprintonly{\large\itshape}%
\@published%
\end{BlockClass}%
\warpprintonly{\end{center}}%
}

\newcommand{\@subtitle}{}

\newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}

\renewcommand*{\maketitlehookb}{\printsubtitle}

\newcommand*{\printsubtitle}{%
\warpprintonly{\begin{center}\unskip}%
\begin{BlockClass}{subtitle}%
\warpprintonly{\Large\itshape}%
\@subtitle%
\end{BlockClass}%
\warpprintonly{\end{center}}%
}

}% \AddSubtitlePublished
}% AfterPackage
```

```
\end{warpall}
```

# 56 Abstract

The following code replaces the LaTeX default, and will itself be replaced later if the abstract package is loaded.

**for HTML output:**

```
\begin{warpHTML}
```

`\abstractname` User-redefinable title for the abstract.

Also over-written by the babel package.

```
\providecommand*{\abstractname}{Abstract}
```

Env `abstract`

```
\DeclareDocumentEnvironment{abstract}{}
{
\LWR@forcenewpage
\BlockClass{abstract}
\BlockClassSingle{abstracttitle}{\abstractname}
}
{
\endBlockClass
}
```

```
\end{warpHTML}
```

# 57 Quote and verse

## 57.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

**for HTML output:**

```
\begin{warpHTML}
\newcommand{\attribution}[1]{%
\InlineClass{attribution}{--\,#1}}% emdash
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand{\attribution}[1]{\textsc{---\,#1}}
\end{warpprint}
```

`\citetitle` for use inside quote, quotation, verse:

**for HTML output:**

```
\begin{warpHTML}
\newcommand{\citetitle}[1]{%
\InlineClass{citetitle}{--\,#1}}% emdash
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand{\citetitle}[1]{\textsl{---\,#1}}
\end{warpprint}
```

## 57.2 Quotes, quotations

**for HTML output:**

```
\begin{warpHTML}
```

Env `quote`

```
\renewenvironment*{quote}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquote}
}
{\LWR@htmlblocktag{/blockquote}}

\renewenvironment*{quotation}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquotation}
}
{\LWR@htmlblocktag{/blockquotation}}
```

```
\end{warpHTML}
```

## 57.3 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
Len \leftmargini
Len \TMLvleftskip
Len \TMLleftmargini

These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths \HTMLvleftskip and \HTMLleftmargini are provided to control the margins in HTML output. These new lengths may be set by the user before any verse environment, and persist until they are manually changed again. One reason to change \HTMLleftmargini is if there is a wide \flagverse in use, such as the word "Chorus", in which case the value of \HTMLleftmargini should be set to a wide enough length to contain "Chorus". The default is wide enough for a stanza number.

Horizontal spacing relies on pdftotext's ability to discern the layout (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargini or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

**for HTML & PRINT:**

```
\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.

```
\newlength{\HTMLvleftskip}
\setlength{\HTMLvleftskip}{1em}
```

Len \TMLleftmargini Sets \leftmargini inside a verse environment in HTML.

```
\newlength{\HTMLleftmargini}
\setlength{\HTMLleftmargini}{4.5em}
```

```
\end{warpall}
```

# 58 Verbatim and tabbing

**for HTML & PRINT:**

```
\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.

```
\newlength{\VerbatimHTMLWidth}
\setlength{\VerbatimHTMLWidth}{4in}
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

Bool LWR@verbtags

Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
\newbool{LWR@verbtags}
\booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [⟨*style*⟩] {⟨*class*⟩}

Encloses a verbatim environment with the given CSS class.

```
\newcommand*{\LWR@atbeginverbatim}[2][]
{%
```

Avoid excessive space between lines:

```
\setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
\LWR@stoppars%
```

Create a new `pre` of the given class. The tags may temporarily be turned off for internal use, such as loading the MathJax script.

```
\ifbool{LWR@verbtags}{%
    \LWR@htmltag{pre class="#2"
    \ifthenelse{\equal{#1}{}}{}{style="#1"}%
}% pre
}{}%
```

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.

```
\begingroup%
\LWR@origttfamily%
```

Since inside a <pre>, restore the original list processing:

```
\LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
\LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain LaTeX \hspace functionality:

```
\LetLtxMacro{\hspace}{\LWR@orighspace}%
}
```

\LWR@afterendverbatim Finishes enclosing a verbatim environment.

```
\newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
\endgroup%
\unskip%
```

At the end of the environment, close the pre:

```
\ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}

}{}%
```

Resume regular paragraph handling:

```
\LWR@startpars%
}
```

\verbatiminput {⟨*filename*⟩}

Patch \verbatiminput to add HTML tags:

```
\let\LWRV@origverbatim@input\verbatim@input

```

```
\renewcommand{\verbatim@input}[2]{%
\ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
\LWRV@origverbatim@input{#1}{#2}%
\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
```

Env `verbatim`

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verbatim.}
\AtBeginEnvironment{verbatim}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{verbatim}{%
    \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
}
```

Env `tabbing`

```
\AfterEndPreamble{
\LWR@traceinfo{Patching tabbing.}
\AtBeginEnvironment{tabbing}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{tabbing}{%
    \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
}
```

```
\end{warpHTML}
```

## 59 Theorems

`\newtheorem` {⟨*text*⟩} [⟨*counter*⟩] -or- [⟨*oldname*⟩] {⟨*text*⟩}

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.

- The contents are placed inside a `<div>` of class theoremcontents.

**for HTML output:**

```
\begin{warpHTML}
```

`\@begintheorem` {⟨*name*⟩} {⟨*number*⟩}

```
\renewcommand{\@begintheorem}[2]{%
\LWR@forcenewpage
\BlockClass{theoremcontents}
\InlineClass{theoremlabel}{#1\ #2\ }
}
```

`\@opargbegintheorem` {⟨*name*⟩} {⟨*number*⟩} {⟨*oparg*⟩}

```
\renewcommand{\@opargbegintheorem}[3]{%
\LWR@forcenewpage
\BlockClass{theoremcontents}
\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
}
```

`\@endtheorem`

```
\renewcommand*{\@endtheorem}{%
\endBlockClass% theoremcontents
}
```

```
\end{warpHTML}
```

# 60 Lists

The environments `itemize`, `enumerate`, and `description` are patched when lwarp is started. These patches support the standard LaTeX environments, as well as those of enumerate, enumitem, and paralist, and at least the French version of babel. Additional patches are done on a package-specific basis.

The LaTeX source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.

empty item To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

```
begin{itemize}
item \mbox{}
    \begin{itemize}
...
    \end{itemize}
item \
    \begin{itemize}
  ...
    \end{itemize}
```

## 60.1 List environment

**for HTML output:**

```
\begin{warpHTML}
```

The following may be locally redefined by `enumerate` or `description`.

```
\newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\newcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
```

`\LWR@listitem` [⟨*label*⟩]

Handles `\item` inside a `list`, `itemize`, or `enumerate`.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@listitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

Env `list` {⟨*label*⟩} {⟨*commands*⟩}

```
\newcommand*{\LWR@liststart}{%
\LWR@stoppars%
\LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
\LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
\LWR@startpars%
\setlength{\topsep}{0pt}%
\setlength{\partopsep}{0pt}%
\setlength{\itemsep}{0pt}%
\setlength{\parsep}{0pt}%
\setlength{\leftmargin}{0pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\listparindent}{0pt}%
\setlength{\itemindent}{0pt}%
\setlength{\labelsep}{1em}%
}

\newcommand*{\LWR@listend}{%
\LWR@stoppars%
\LWR@closeprevious{\LWR@depthlistitem}%
\LWR@closeoneprevious%
\LWR@startpars%
}
```

## 60.2 Itemize

\LWR@itemizeitem [⟨*label*⟩]

Handles \item inside an itemize or enumerate.

See \LWR@openparagraph where extra \hspace is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@itemizeitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

Env itemize [⟨*options*⟩]

```
\newcommand*{\LWR@itemizestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
\let\item\LWR@itemizeitem%
}
```

## 60.3 Enumerate

An HTML unordered list is used with customized LaTeX-generated labels.

Env enumerate [⟨*options*⟩]

```
\newcommand*{\LWR@enumeratestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
\let\item\LWR@itemizeitem%
}
```

## 60.4 Description

\LWR@descitem [⟨*label*⟩] Handles an \item inside a description.

```
\newcommand*{\LWR@descitem}[1][]%
{%
\LWR@stoppars%
\LWR@setlatestname{#1}%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem}%
```

Temporarily disable \hspace, which article.cls, etc. use per \item for descriptions only. This causes lwarp to mistakenly place an empty span between HTML list tags.

```
\LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original \item code:

```
\LWR@origitem[]%
```

Restore \hspace for use in the item text:

```
\LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
\LWR@orighspace{1in}
\LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
\LWR@orignewline%
\LWR@htmltag{dd}%
\LWR@startpars%
}
```

Env description [⟨*options*⟩]

```
\newcommand*{\LWR@descriptionstart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
\renewcommand*{\LWR@printopenlist}{dl}
\let\item\LWR@descitem%
}
```

## 60.5 Patching the lists

```
\let\LWR@orig@trivlist\@trivlist
\let\LWR@origtrivlist\trivlist
\let\LWR@origendtrivlist\endtrivlist
```

`\LWR@patchlists` Patches list environments.

`\LWr@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native LaTeX environments, as well as those provided by enumitem, enumerate, and paralist.

```
\newcommand*{\LWR@patchlists}{%
    \let\item\LWR@listitem%
    \renewcommand*{\@trivlist}{%
        \LWR@liststart%
        \LWR@orig@trivlist%
    }%
    \renewcommand*{\endtrivlist}{\LWR@origendtrivlist\LWR@listend}%
    \renewcommand*{\trivlist}{%
        \LWR@origtrivlist%
        \def\makelabel####1{\LWR@orighspace{3em}####1}%
    }%
    \renewcommand*{\itemize}{\LWR@itemizestart\LWR@origitemize}%
    \renewcommand*{\enumerate}{\LWR@enumeratestart\LWR@origenumerate}%
    \renewcommand*{\description}{\LWR@descriptionstart\LWR@origdescription}%
%     \AtBeginEnvironment{itemize}{\LWR@itemizestart}%
%     \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}%
%     \AtBeginEnvironment{description}{\LWR@descriptionstart}%
}
```

`\LWR@restoreoriglists` Restores the original `trivlist` environment.

```
\newcommand*{\LWR@restoreoriglists}{%
\let\@trivlist\LWR@orig@trivlist%
\let\trivlist\LWR@origtrivlist%
\let\endtrivlist\LWR@origendtrivlist%
\LetLtxMacro\item\LWR@origitem%
\LetLtxMacro\itemize\LWR@origitemize%
\LetLtxMacro\enditemize\LWR@endorigitemize%
\LetLtxMacro\enumerate\LWR@origenumerate%
\LetLtxMacro\endenumerate\LWR@endorigenumerate%
\LetLtxMacro\description\LWR@origdescription%
\LetLtxMacro\enddescription\LWR@endorigdescription%
}
```

```
\end{warpHTML}
```

# 61 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

## 61.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, `*` column specifiers, siunitx `S` columns, or the packages multirow, longtable, supertabular, or xtab.

**Defining environments:**

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include `tabular` and instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands. These are are ignored in print mode.

  ```
  \StartDefiningTabulars
  <define macros or environments using tabular and &
  here>
  \EndDefiningTabulars
  ```

⚠ `tabular` inside another
environment

- When creating a new environment which contains a `tabular` environment, lwarp's emulation of the tabular does not automatically resume when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

**Cell contents:**

⚠ paragraphs

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.

⚠ \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
    ... &   \multirow{2}{.5in}{text}   & ...
    ... &   \mrowcell                  & ...
```

vposn

  Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 192.2 for \multicolumrow.

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
    ... &   \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}   & ...
    ... &   \mcolrowcell          &           \mcolrowcell   & ...
    ... &   \mcolrowcell          &           \mcolrowcell   & ...
```

vposn

Note that recent versions of multirow include a new optional `vposn` argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

  ```
  \TabularMacro\somemacro & more row contents \\
  ```

**Column specifiers:**

⚠ * column specification

- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of `@` and `!` is used at each column, and they are used in that order.

\multirow

- In `\multirow` cells, the print version may have extra instances of <, >, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ \newcolumntype

- `\newcolumntype` is ignored; unknown column types are set to `l`.

**Rules:**

vertical rules

- Vertical rules next to either side of an `@` or `!` column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with `@` or `!` columns, and full-width rules ignore trim.

full-width rules

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:

  ```
  \cmidrule{2-3} \\[-2ex]
  \bottomrule
  ```

  The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a "misplaced `\noalign`" error occurs, change

  ```
  This & That \endhead
  ```

  to

  ```
  \warpprintonly{This & That \endhead}
  ```

  and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevent to HTML output.

**Other:**

- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.
- longtable headings — For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}.
- ⚠ S columns — For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarp's tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

## 61.2 Token lookahead

Used by \LWR@futurenonspacelet to look at the next token.

**for HTML output:**

```
\begin{warpHTML}
```

\LWR@mynexttoken

```
\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

```
\def\LWR@futurenonspacelet#1{\def\LWR@cs{#1}%
\afterassignment\LWR@fnslone\let\nexttoken= }
\def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
\def\LWR@fnsltwo{%
\expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
\else\let\next=\nexttoken\fi\next}
\def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
\newcommand*{\LWR@getmynexttoken}{%
\LWR@traceinfo{LWR@getmynexttoken}%
% nothing must follow this next line
\LWR@futurenonspacelet\LWR@mynexttoken\LWR@tabledatacolumntag
}
```

## 61.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```
\newbool{LWR@startedrow}
\boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
\newbool{LWR@tabularcelladded}
\boolfalse{LWR@tabularcelladded}
```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for `\midrule`.

```
\newbool{LWR@doinghline}
\boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
\newbool{LWR@doingtbrule}
\boolfalse{LWR@doingtbrule}
```

Bool LWR@doingcmidrule True if the next row will have a cmidrule above it.

This is used by `\LWR@tabularfinishrow` to force a final empty row to create the border for the `\cmidrule`.

```
\newbool{LWR@doingcmidrule}
\boolfalse{LWR@doingcmidrule}
```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
\newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty `\multirow` cell, and thus there is no data tag to close.

```
\newbool{LWR@skippingmrowcell}
```

Bool LWR@skippingmcolrowcell True if are doing an empty `\multicolumnrow` cell, and thus there is no data tag to close, and do not print @ and ! columns.

```
\newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
\newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
\newbool{LWR@emptyatbang}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
\newbool{LWR@intabularmetadata}
\boolfalse{LWR@intabularmetadata}
```

## 61.4 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:
http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860

\LWR@instertatbangcols

```
\newcommand*{\LWR@insertatbangcols}{%
\ifbool{LWR@skipatbang}%
{}%
{%
    \LWR@printatbang{at}{\arabic{LWR@tablecolindex}}%
    \LWR@printatbang{bang}{\arabic{LWR@tablecolindex}}%
}%
}
```

\LWR@closetabledatacell If LWR@skippingmrowcell or LWR@skippingmcolrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
\newcommand*{\LWR@closetabledatacell}{%
\global\booltrue{LWR@intabularmetadata}%
\ifbool{LWR@exitingtabular}{}%
{% not exiting tabular
    \ifboolexpr{bool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
    {%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this nonexistant column.

```
        \ifbool{LWR@skippingmcolrowcell}%
        {}%
        {\LWR@insertatbangcols}%
    }%
    {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```
        \unskip%
        \ifboolexpr{%
            bool{LWR@tabularmutemods} or
            bool{LWR@skipatbang} or
            bool{LWR@emptyatbang}
        }%
        {}%
        {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tablecolindex}}}%
```

Close paragraphs:

```
        \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
        \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
        \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
        \global\booltrue{LWR@tabularcelladded}%
        \LWR@insertatbangcols%
    }% not skipping mrowcell
}% not exiting tabular
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
\global\boolfalse{LWR@skipatbang}%
}
```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```
\newcounter{LWR@tabulardepth}
\setcounter{LWR@tabulardepth}{0}

```

When not used inside a tabular, & performs its original function as recorded here ( with catcode 4 ).

```
\let\LWR@origampmacro&
```

```
\end{warpHTML}
```

### 61.4.1 Localizing & catcodes

**for HTML & PRINT:**

```
\begin{warpall}
```

⚠ misplaced alignment tab character &

Place \StartDefiningTabulars and \EndDefiningTabulars before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of & must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
        misplaced alignment tab character &
```

\StartDefiningTabulars

Place before defining something with & in it.

```
\newcommand{\StartDefiningTabulars}{%
\LWR@traceinfo{StartDefiningTabulars}
\warpHTMLonly{\catcode`\&=\active}%
}
```

\EndDefiningTabulars

Place after defining something with & in it.

```
\newcommand{\EndDefiningTabulars}{%
\LWR@traceinfo{EndDefiningTabulars}
\warpHTMLonly{\catcode`\&=4}%
}
```

```
\end{warpall}
```

### 61.4.2 Handling &

**for HTML output:**

```
\begin{warpHTML}
```

&

Will behave depending on whether it is being used inside tabular.

& is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

```
\newcommand*{\LWR@tabularampersand}{%
\LWR@traceinfo{LWR@tabularampersand}%
\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
{%
```

If not skipping a multirow cell, close the current data cell.

```
    \unskip%
    \LWR@closetabledatacell%
```

Move to the next column.

```
    \addtocounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
    \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
    \LWR@getmynexttoken%
}%
```

If not inside a tabular, performs the original action:

```
{\LWR@origampmacro}%
}
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded. Also, cleveref uses the ampersand in one of its options.

& is made active inside a tabular.

& is left alone when in math alignments.

### 61.4.3 Filling an unfinished row

\LWR@tabularfinishrow Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
\newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
\ifboolexpr{%
    not bool {LWR@exitingtabular} or%
    bool{LWR@doingtbrule} or%
    bool{LWR@doingcmidrule} or%
    bool{LWR@doinghline} or%
    bool{LWR@startedrow}%
}{%
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
\begingroup%
```

If generating a final row for the `\bottomrule` borders, turn off the `@`, `!`, `<`, and `>` column output:

```
\ifbool{LWR@exitingtabular}{%
    \booltrue{LWR@tabularmutemods}%
}{}%
```

Reenable the table data tags until finished with the final row:

```
\global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
\whileboolexpr{%
    test {
        \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {
            \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}
        }%
    )%
}%
{%
    \LWR@tabledatasinglecolumntag%
```

The following is essentially `\LWR@tabularampersand` with `LWR@emptyatbang` added to empty the following cells:

```
    \LWR@closetabledatacell%
    \addtocounter{LWR@tablecolindex}{1}%
    \boolfalse{LWR@tabularcelladded}%
    \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
    \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
    {\LWR@getmynexttoken}%
    {}%
}%
```

Reenable the original LWR@exitingtabular to close the entire table:

```
\endgroup%
\global\boolfalse{LWR@emptyatbang}%
}{}% ifboolexpr
}
```

## 61.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or \\*

\LWR@tabularendofline

```
\NewDocumentCommand{\LWR@tabularendofline}{s o}{%
\ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
{\LWR@tabularfinishrow}%
{\LWR@closetabledatacell}%
%    \begin{macrocode}
\LWR@htmltag{/tr}\LWR@orignewline
\global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}
```

Additional setup:

```
\global\boolfalse{LWR@doinghline}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
\LWR@clearmidrules%
```

Start at first column:

```
\setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
\boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide between single column data tag or a special case:

```
\LWR@getmynexttoken%
}
```

## 61.6 Variables

```
\newcommand*{\LWR@colsresult}{}%temp storage for column format results
\newcommand*{\LWR@pposition}{}
\newcommand*{\LWR@pleft}{}
\newcommand*{\LWR@pright}{}
```

\LWR@tablecolspec Holds the parsed column specification, of total width LWR@tabletotalcols, not counting @ and ! columns.

Will contain a string such as llrrccpc, exactly one letter per LaTeX table column, without @, !, >, <, or the vertical bar.

```
\newcommand*{\LWR@tablecolspec}{}
```

\LWR@strresult Holds the result of Str functions.

```
\providecommand*{\LWR@strresult}{}
\providecommand*{\LWR@strresulttwo}{}
```

\LWR@origcolspec Holds the original column specs given to tabular.

```
\newcommand*{\LWR@origcolspec}{}
```

Ctr LWR@tablecolspecwidth Holds the number of tokens in the table columns specification.

This is includes one for each @, !, <, > column, and also one for each of the parameters of p, @, !, <, > columns, and three for each D column.

(This is not the total # of LaTeX columns in the table.)

```
\newcounter{LWR@tablecolspecwidth}
```

Ctr LWR@tablecolspecindex While parsing the LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
\newcounter{LWR@tablecolspecindex}
```

Ctr LWR@tablecolindex While parsing the LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
\newcounter{LWR@tablecolindex}
```

Ctr LWR@tabletotalcols While parsing a table column specification, begins at 0 and increments by 1 per LaTeX table column. Eventually holds the final number of LaTeX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalcols.)

```
\newcounter{LWR@tabletotalcols}
```

Ctr LWR@tabletotalcolsnext Holds the next LaTeX table column index while parsing, equal to one more than LWR@tabletotalcols.

```
\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 35.

LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 35.

LWR@colbeforespec A data array of specifications for > columns.

LWR@colafterspec A data array of specifications for < columns.

LWR@colbarspec A data array of specifications for vertical rules.

## 61.7 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
\newcommand*{\LWR@colparameter}{}
```

\LWR@parseatcolumn Handles @{text} columns.

```
\newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
\LWR@traceinfo{at column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
    \LWR@traceinfo{at the left edge: %
    \LWR@getexparray{LWR@colatspec}{leftedge}}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
    \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
    \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

\LWR@parsebangcolumn

Handles !{text} columns.

```
\newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
\LWR@traceinfo{bang column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
    \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

\LWR@parsebeforecolumn

Handles >{text} columns.

```
\newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

\LWR@parseaftercolumn

Handles <{text} columns.

```
\newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

\LWR@parsebarcolumn

Handles vertical rules.

```
\newcommand*{\LWR@parsebarcolumn}{%
\LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
}%
{% not at the left edge:
    \LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcols}}{tvertbarr}%
}%
\booltrue{LWR@validtablecol}%
}
```

## 61.8 Parsing 'l', 'c', or 'r' columns

\LWR@parsenormalcolumn {⟨*thiscolumn*⟩}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
\newcommand*{\LWR@parsenormalcolumn}[1]{%
\appto\LWR@tablecolspec{#1}%
\addtocounter{LWR@tabletotalcols}{1}%
\addtocounter{LWR@tabletotalcolsnext}{1}%
\LWR@traceinfo{normal column \arabic{LWR@tabletotalcols}: #1}%
\LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\booltrue{LWR@validtablecol}%
}
```

## 61.9 Parsing ‘p’, ‘m’, or ‘b’ columns

\LWR@parsepcolumn {⟨*thiscolumn*⟩} The width will be ignored.

```
\newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
\addtocounter{LWR@tablecolspecindex}{1}%
}
```

## 61.10 Parsing ‘D’ columns

From the dcolumn package.

\LWR@parseDcolumn {⟨*thiscolumn*⟩} The three parameters will be ignored.

```
\newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
\addtocounter{LWR@tablecolspecindex}{3}%
}
```

## 61.11 Parsing the column specifications

⚠ HTML CSS cannot exactly match the LaTeX concept of a baseline for a table row. Table 8 shows the LaTeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool LWR@validtablecol True if found a valid table column type.

Table 8: Tabular baseline

| l | p | m | b | r |
|---|---|---|---|---|
|  |  |  | bot |  |
|  |  | mid | bot |  |
| l | par | mid | bot | r |
|  | par | mid |  |  |
|  | par |  |  |  |

Table 9: 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 8, LaTeX 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 LaTeX baselines.

**m:** With paragraph tags, CSS `vertical-align:middle`.

**b:** With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

**P, M, B:** Horizontally-centered versions.

**S:** Converted to 'r'. From the siunitx package.

**D:** Converted to 'c'. From the dcolumn package.

**@, !, >, <:** One each, in that order.

**|:** Vertical rule.

**Unknown:** Converted to 'l'.

**`\newcolumn`:** Currently treated as unknown.

```
\newbool{LWR@validtablecol}
```

\LWR@parsetablecols {⟨*colspecs*⟩}

Scans the column specification left to right.

Builds \LWR@tablecolspec with the final specification, one column per entry. The final number of cells in each row is stored in LWR@tabletotalcols.

```
\newcommand*{\LWR@parsetablecols}[1]{%
\LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
\renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
\expandarg%
\StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
\renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
\setcounter{LWR@tabletotalcols}{0}%
\setcounter{LWR@tabletotalcolsnext}{1}%
\LWR@setexparray{LWR@colatspec}{leftedge}{}%
\LWR@setexparray{LWR@colatspec}{1}{}%
\LWR@setexparray{LWR@colatspec}{2}{}%
\LWR@setexparray{LWR@colatspec}{3}{}%
\LWR@setexparray{LWR@colbangspec}{leftedge}{}%
\LWR@setexparray{LWR@colbangspec}{1}{}%
\LWR@setexparray{LWR@colbangspec}{2}{}%
\LWR@setexparray{LWR@colbangspec}{3}{}%
\LWR@setexparray{LWR@colbeforespec}{1}{}%
\LWR@setexparray{LWR@colbeforespec}{2}{}%
\LWR@setexparray{LWR@colbeforespec}{3}{}%
\LWR@setexparray{LWR@colafterspec}{1}{}%
\LWR@setexparray{LWR@colafterspec}{2}{}%
\LWR@setexparray{LWR@colafterspec}{3}{}%
\LWR@setexparray{LWR@colbarspec}{leftedge}{}%
\LWR@setexparray{LWR@colbarspec}{1}{}%
\LWR@setexparray{LWR@colbarspec}{2}{}%
\LWR@setexparray{LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
\setcounter{LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
\LWR@traceinfo{about to StrLen}%
\expandarg%
\StrLen{\LWR@origcolspec}[\LWR@strresult]%
\fullexpandarg%
\LWR@traceinfo{finished StrLen}%
\setcounter{LWR@tablecolspecwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@tablecolspecindex}}{>}{\value{LWR@tablecolspecwidth}}%
    }%
}%
{%
```

Place the next single-character column type into `\LWR@strresult`:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@strresult]%
\LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: \LWR@strresult}%
\fullexpandarg%
```

Not yet found a valid column type

```
\boolfalse{LWR@validtablecol}%
```

⚠ Note that the parameter for a `p{spec}` column is a token list which will NOT match l,c,r,p.

```
\IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
\IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
\IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
```

```
\IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
\IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the dcolumn package:

```
\IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
\IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions "P", "M", and "B":

```
\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

```
\IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an `l` column:

```
\ifbool{LWR@validtablecol}{}{%
    \LWR@traceinfo{invalid column type: \LWR@strresult}%
    \LWR@parsenormalcolumn{l}%
}%
\addtocounter{LWR@tablecolspecindex}{1}%
}% whiledo
}%
```

## 61.12 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for `hline` or `tbrule` if necessary.

```
\newcommand*{\LWR@maybenewtablerow}
{%
\ifbool{LWR@startedrow}%
{}% started the row
{% not started the row
```

Remember that now have started the row:

```
    \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
    \global\booltrue{LWR@intabularmetadata}%
    \ifbool{LWR@doinghline}%
    {%
        \LWR@htmltag{tr class="hline" }%
        \LWR@orignewline%
    }%
    {% not doing hline
        \ifbool{LWR@doingtbrule}%
        {%
            \LWR@htmltag{tr class="tbrule"}%
            \LWR@orignewline%
        }%
        {\LWR@htmltag{tr}\LWR@orignewline}%
    }% end of not doing hline
}% end of not started the row
}
```

## 61.13 Printing vertical bar tags

\LWR@printbartag {⟨*index*⟩}

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```
\newcommand*{\LWR@printbartag}[1]{%
\ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
{}% muting or empty
{% not muting
    \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{#1}}%
    \ifdefempty{\LWR@tempone}{}{ \LWR@tempone}%
}% not muting
}
```

## 61.14 Printing at or bang tags

\LWR@printatbang {⟨*at -or- bang*⟩} {⟨*index*⟩}

```
\newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```
\edef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}%
\LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```
\ifdefempty{\LWR@atbangspec}%
{}%
{% not empty
    \LWR@htmltag{%
        td class="td#1%
        \LWR@subaddcmidruletrim{}{}%
        \LWR@printbartag{#2}%
        "%
        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \LWR@tdendstyles%
    }%
```

Create an empty cell if muting for the `\bottomrule`:

```
    \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
    {}%
    {\LWR@atbangspec}%
%
    \LWR@htmltag{/td}\LWR@orignewline%
    \global\booltrue{LWR@tabularcelladded}%
}% not empty
}%
```

`\LWR@addleftmostbartag`

```
\newcommand*{\LWR@addleftmostbartag}{%
\ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
    \LWR@printbartag{leftedge}%
}{}%
}
```

`\LWR@tabularleftedge`

```
\newcommand*{\LWR@tabularleftedge}{%
\ifnumcomp{\value{LWR@tablecolindex}}{=}{1}%
{%
    \LWR@printatbang{at}{leftedge}%
    \LWR@printatbang{bang}{leftedge}%
}% left edge
{}% not left edge
}
```

## 61.15 Data opening tag

\LWR@thiscolspec Temporary storage.

```
\newcommand*{\LWR@thiscolspec}{}
```

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment

```
\newcommand*{\LWR@tabledatasinglecolumntag}%
{%
\LWR@maybenewtablerow%
```

Don't start a new paragraph tag if have already started one:

```
\ifbool{LWR@intabularmetadata}%
{%
```

If have found the end of tabular command, do not create the next data cell:

```
    \ifbool{LWR@exitingtabular}{}%
    {% not exiting tabular
```

Print the @ and ! contents before first column:

```
        \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
        \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
        \LWR@htmltag{td class="td%
```

append this column's spec:

```
        \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
        \LWR@addcmidruletrim%
        \LWR@addleftmostbartag%
        \LWR@printbartag{\arabic{LWR@tablecolindex}}%
        "%
```

```
        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@thiscolspec]%
        \LWR@addformatwpalignment{\LWR@thiscolspec}%
        \LWR@tdendstyles%
        }%
```

If this is a p, m, b, or X column, allow paragraphs:

```
        \ifboolexpr{%
            test{ \ifdefstring{\LWR@strresult}{p} } or
            test{ \ifdefstring{\LWR@strresult}{m} } or
            test{ \ifdefstring{\LWR@strresult}{b} } or
            test{ \ifdefstring{\LWR@strresult}{P} } or
            test{ \ifdefstring{\LWR@strresult}{M} } or
            test{ \ifdefstring{\LWR@strresult}{B} } or
            test{ \ifdefstring{\LWR@strresult}{X} }
        }%
        {% allow pars
            \LWR@startpars%
            \global\booltrue{LWR@tableparcell}%
        }% allow pars
        {}% no pars
```

Print the > contents unless muted for the `\bottomrule`:

```
        \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
        {}%
        {%
            \LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tablecolindex}}%
        }%
        \global\boolfalse{LWR@intabularmetadata}%
    }% not exiting tabular
}{}% in tabular metadata
}%
```

## 61.16 Midrules

`LWR@midrules` `LWR@midrules` is a data array (section 35) of columns each containing a non-zero width if a midrule should be created for this column.

`LWR@trimlrules` `LWR@trimlrules` is a data array (section 35) of columns containing `l` if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 35) of columns containing r if a midrule should be right trimmed for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
\newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
\newlength{\LWR@heavyrulewidth}
\setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
\newlength{\LWR@lightrulewidth}
\setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth The default width of the rule.

```
\newlength{\LWR@cmidrulewidth}
\setlength{\LWR@cmidrulewidth}{.03em}
```

Len \LWR@thiscmidrulewidth The width of the next rule, defaulting to \LWR@cmidrulewidth.

If not \LWR@cmidrulewidth, a style will be used to generate the custom width.

Assigned from the LWR@midrules array.

```
\newlength{\LWR@thiscmidrulewidth}
\setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \\.

Clears all LWR@midrules and LWR@trimruels markers for this line.

```
\newcommand*{\LWR@clearmidrules}
{%
\setcounter{LWR@midrulecounter}{1}%
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{\value{LWR@tablecolspecwidth}}%
    }%
}%
{%
\LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
\setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
\LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
\LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
\addtocounter{LWR@midrulecounter}{1}%
}%
}
```

\LWR@subcmidrule {⟨*width*⟩} {⟨*trim*⟩} {⟨*leftcolumn*⟩} {⟨*rightcolumn*⟩}

Marks LWR@midrules data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```
\newcommand*{\LWR@subcmidrule}[4]{%
\setcounter{LWR@midrulecounter}{#3}%
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
    }%
}%
{%
    \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
    \addtocounter{LWR@midrulecounter}{1}%
}% whiledo
\IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
\IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
\booltrue{LWR@doingcmidrule}%
}
```

\LWR@docmidrule [⟨*width*⟩] {⟨*trim*⟩} {⟨*leftcolumn-rightcolumn*⟩}

Marks LWR@midrules array elements to be “Y” from left to right columns. Also marks trimming for the L and/or R columns.

```
\NewDocumentCommand{\LWR@docmidrule}{O{\LWR@cmidrulewidth} D(){} >{\SplitArgument{1}{-}}m}%
{\LWR@subcmidrule{#1}{#2}#3}
```

Used to compute margins, tabular trims:

```
\newlength{\LWR@templengthone}%
\newlength{\LWR@templengthtwo}%
```

Used to add a style to a table data cell:

```
\newboolean{LWR@tdhavecellstyle}
```

\LWR@tdstartstyles Begins possibly adding a table data cell style.

```
\newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

\LWR@tdaddstyle Starts adding a table data cell style.

```
\newcommand*{\LWR@tdaddstyle}{%
\ifbool{LWR@tdhavecellstyle}%
{; }%
{ style="}%
\booltrue{LWR@tdhavecellstyle}%
}
```

\LWR@tdendstyles Finishes possibly adding a table data cell style. Prints the closing quote.

```
\newcommand*{\LWR@tdendstyles}{%
\ifbool{LWR@tdhavecellstyle}{%
    "%
    \global\boolfalse{LWR@tdhavecellstyle}%
}{}%
}
```

\LWR@subaddcmidruletrim {⟨*lefttrim*⟩} {⟨*righttrim*⟩} Adds a \cmidrule with optional trim.

```
\newcommand*{\LWR@subaddcmidruletrim}[2]{%
\setlength{\LWR@templengthone}{\LWR@getexparray{LWR@midrules}{\arabic{LWR@tablecolindex}}}%
\ifdimcomp{\LWR@templengthone}{>}{0pt}%
{%
```

Print the class without left and right trim letters appended:

```
    \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
    \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
}%
{%
    \setlength{\LWR@thiscmidrulewidth}{0pt}%
}%
}
```

\LWR@addcmidruletrim Adds left or right trim to a \cmidrule.

```
\newcommand*{\LWR@addcmidruletrim}{%
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@tablecolindex}}}%
}
```

\LWR@addrulewidth {⟨*thiswidth*⟩} {⟨*defaultwidth*⟩}

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`:

```
\ifboolexpr{%
    test{\ifdimcomp{#1}{=}{0pt}}%
    or ( test{\ifdimcomp{#1}{=}{#2}} and not bool{FormatWP} )%
}%
{}% default width
{% custom width
```

Ensure that the width is wide enough to display in the browser:

```
    \LWR@forceminwidth{#1}%
```

Begin adding a style:

```
    \LWR@tdaddstyle%
```

The style itself:

```
    \uselengthunit{PT}%
    border-top:\rndprintlength{\LWR@atleastonept} solid %
```

The darkness of the color depends on the thickness of the rule:

```
    \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
    {\#A0A0A0}%
    {% lightrule or heaver
        \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
        {\#808080}%
        {black}%
    }% lightrule or heavier
}% custom width
}
```

\LWR@addcmidrulewidth Adds a `style` for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand{\LWR@addcmidrulewidth}{%
\LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
}
```

`\LWR@WPcell` {⟨*text-align*⟩} {⟨*vertical-align*⟩}

```
\newcommand*{\LWR@WPcell}[2]{%
\LWR@tdaddstyle%
text-align:#1; vertical-align:#2%
}
```

`\LWR@addformatwpalignment` If `FormatWP`, adds a `style` for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand*{\LWR@addformatwpalignment}[1]{%
\ifbool{FormatWP}{%
\IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
\IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
\IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
\IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
\IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
}{}%
}
```

## 61.17 Multicolumns

### 61.17.1 Parsing multicolumns

```
\newcounter{LWR@tablemulticolswidth}
```

Indexes into the multicolumn specification:

```
\newcounter{LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
\newbool{LWR@mcolvertbarl}
\newbool{LWR@mcolvertbarr}
```

\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.

```
\newcommand*{\LWR@printmccoltype}[1]{%
\LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
\IfStrEq{\LWR@strresult}{l}{l}{}%
\IfStrEq{\LWR@strresult}{c}{c}{}%
\IfStrEq{\LWR@strresult}{r}{r}{}%
\IfStrEq{\LWR@strresult}{p}{p}{}%
\IfStrEq{\LWR@strresult}{m}{m}{}%
\IfStrEq{\LWR@strresult}{b}{b}{}%
\IfStrEq{\LWR@strresult}{P}{P}{}%
\IfStrEq{\LWR@strresult}{M}{M}{}%
\IfStrEq{\LWR@strresult}{B}{B}{}%
\IfStrEq{\LWR@strresult}{S}{r}{}%
\IfStrEq{\LWR@strresult}{X}{p}{}%
```

```
\IfStrEq{\LWR@strresult}{|}{%
    \ifnumcomp{\value{LWR@tablemulticolspos}}{=}{1}%
        {\booltrue{LWR@mcolvertbarl}}%
        {\booltrue{LWR@mcolvertbarr}}%
}{}%
\LWR@traceinfo{lwr@printmccoltype done}%
}
```

\LWR@multicolpartext Print the data with paragraph tags:

```
\newcommand*{\LWR@multicolpartext}{%
\LWR@startpars%
\LWR@multicoltext%
\LWR@stoppars%
}
```

\LWR@multicolother {⟨*colspec*⟩} For @, !, >, <, print the next token without paragraph tags:

```
\newcommand*{\LWR@multicolother}[1]{%
\addtocounter{LWR@tablemulticolspos}{1}%
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
\LWR@strresult%
```

A valid column data type was found:

```
\booltrue{LWR@validtablecol}%
}
```

\LWR@multicolskip Nothing to print for this column type.

```
\newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
\booltrue{LWR@validtablecol}%
}
```

\LWR@printmccoldata {⟨*colspec*⟩} Print the data for any valid column type found.

```
\newcommand*{\LWR@printmccoldata}[1]{%
\LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
\boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles `@`, `>`, `<` as it comes to them.

```
\IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{D}{%
\addtocounter{LWR@tablemulticolspos}{3}% skip parameters
\LWR@multicoltext%
}{}%
\IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
```

```
\IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
\ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```
\LWR@traceinfo{lwr@printmccoldata done}%
}
```

`\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`.

```
\newcommand*{\LWR@parsemulticolumnalignment}[2]{%
\setcounter{LWR@tablemulticolspos}{1}%
\StrLen{#1}[\LWR@strresult]%
\setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@tablemulticolspos}}{>}{\value{LWR@tablemulticolswidth}}%
    }
}%
{%
```

Execute the assigned print function for each token in the column spec:

```
#2{#1}%
```

Move to the next token in the column spec:

```
\addtocounter{LWR@tablemulticolspos}{1}%
}%
}
```

### 61.17.2 Multicolumn factored code

```
\newcommand{\LWR@multicoltext}{}
```

To find multicolumn right trim:

```
\newcounter{LWR@lastmulticolumn}
```

\LWR@domulticolumn [⟨*1: vpos*⟩] [⟨*2: #rows*⟩] {⟨*3: numLaTeXcols*⟩} {⟨*4: numHTMLcols*⟩} {⟨*5: colspec*⟩} {⟨*6: text*⟩}

```
\NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%
\LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
\renewcommand{\LWR@multicoltext}{%
#6%
\booltrue{LWR@validtablecol}%
}%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
\setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
\addtocounter{LWR@lastmulticolumn}{#3}%
\addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
\LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
\LWR@htmltag{td colspan="#4" %
```

```
\IfValueT{#2}{ % rows?
rowspan="#2" %
```

```
\IfValueT{#1}{% vpos?
\ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
\ifstrequal{#1}{t}{style="vertical-align:top" }{}%
}% vpos?
}% rows?
```

```
 class="td%
```

Print the column type and vertical bars:

```
\boolfalse{LWR@mcolvertbarl}%
```

```
\boolfalse{LWR@mcolvertbarr}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag.

If this position had a "Y" then add "rule" for a horizontal rule:

```
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%
```

Also add vertical bar class.

```
\ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
\ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%
```

Close the class tag's opening quote:

```
"%
```

```
\LWR@tdstartstyles%
\LWR@addcmidrulewidth%
\LWR@addformatwpalignment{#5}%
\LWR@tdendstyles%
}% end of the opening table data tag
\global\boolfalse{LWR@intabularmetadata}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
}
```

### 61.17.3 Multicolumn

`\LWR@htmlmulticolumn` {⟨*numcols*⟩} {⟨*alignment*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
{%
```

Figure out how many extra HTML columns to add for `@` and `!` columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tablecolindex}{#1}%
\addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
\booltrue{LWR@skipatbang}%
}
```

### 61.17.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable

Per the caption pacakge, step the counter if longtable*.

```
\newbool{LWR@starredlongtable}
\boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
\providecommand*{\LTcaptype}{table}
```

\LWR@longtabledatacaptiontag

* [⟨*toc entry*⟩] {⟨*caption*⟩}

```
\NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
{%
```

Remember the latest name for \nameref:

```
\IfValueTF{#2}{% optional given?
    \ifblank{#2}% optional empty?
    {\LWR@setlatestname{#3}}% empty
    {\LWR@setlatestname{#2}}% given and non-empty
}% optional given
{\LWR@setlatestname{#3}}% no optional
```

Create a multicolumn across all the columns:

Figure out how many extra HTML columns to add for @ and ! columns found between the first and the last column:

```
\LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
```

```
{% \LWR@domulticolumn
\IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
{% yes star
    \LWR@figcaption%
    #3%
    \endLWR@figcaption%
}%
{% No star:
```

Not the star version:

Don't step the counter if `\caption[]{A caption.}`

```
    \ifbool{LWR@starredlongtable}%
    {%
        \ifblank{#2}% TOC entry
        {}%
        {%
            \refstepcounter{\LTcaptype}%
            \protected@edef\@currentlabel{%
            \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
        }%
    }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
    \LWR@figcaption%
    \csuse{fnum@\LTcaptype}\CaptionSeparator#3%
    \endLWR@figcaption%
```

See if an optional caption was given:

```
    \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
    {}%
```

If the optional caption was given, but might only be `[]`:

```
    {% TOC entry not empty
        \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is `[]`:

```
        {% No TOC entry
            \addcontentsline%
            {\csuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
            {\ignorespaces #3\protect\relax}%
            }%
        }% end of No TOC entry
```

The optional caption has text enclosed:

```
        {% yes TOC entry
            \addcontentsline%
            {\csuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
            {\ignorespaces #2\protect\relax}%
            }%
        }% end of yes TOC entry
    }% end of TOC entry not empty
}% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
\booltrue{LWR@skipatbang}%
}% end of \LWR@domulticolumn

\addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
\addtocounter{LWR@tablecolindex}{-1}

}
```

### 61.17.5 Counting HTML tabular columns

The LaTeX specification for a table includes a number of columns separated by the & character. These columns differ in content from line to line. Additional virtual columns may be specified by the special @ and ! columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, @ and ! columns are placed into their own tabular columns. Thus, a LaTeX `\multicolumn` command may span several additional @ and ! columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```
\newcounter{LWR@tabhtmlcolindex}
\newcounter{LWR@tabhtmlcolend}
\newcounter{LWR@tabhtmlcoltotal}
```

`\LWR@subtabularhtmlcolumns` {⟨*index*⟩}

Factored from `\LWr@tabularhtmlcolumns`, which follows.

```
\newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the `@` specification for this column:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the `@` specification is not empty, add to the count:

```
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the `!` columns:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
}
```

`\LWR@tabularhtmlcolumns` {⟨*starting LaTeX column*⟩} {⟨*number LaTeX columns*⟩}

Compute the total number of HTML columns being spanned, considering the starting LaTeX table column and the number of LaTeX tabular columns being spanned. Any `@` and `!` columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
\newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the LaTeX span, to which additional `@` and `!` columns may be added:

```
\setcounter{LWR@tabhtmlcolindex}{#1}%
\setcounter{LWR@tabhtmlcoltotal}{#2}%
\setcounter{LWR@tabhtmlcolend}{#1}%
\addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
\ifnumcomp{\value{LWR@tabhtmlcolindex}}{=}{1}{%
    \LWR@subtabularhtmlcolumns{leftedge}%
}{}%
```

Walk across the LaTeX columns looking for `@` and `!` columns:

```
\whileboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}%
    }%
}%
{%
    \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
    \addtocounter{LWR@tabhtmlcolindex}{1}%
}% whiledo
}
```

```
\end{warpHTML}
```

## 61.18 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 192 for the HTML versions.

**for HTML & PRINT:**

```
\begin{warpall}
```

`\multicolumnrow` {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

For discussion of the use of `\DeclareExpandableDocumentCommand`, see: https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex

After the user may have

```
\AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

```
\@ifundefined{@xmultirow}
{}% no version of multirow was loaded
{% \@xmultirow defined, so some version of multirow was loaded
```

\@ifpackageloaded{multirow} determines if v2.0 or later of multirow was used, which included the \ProvidesPackage macro.

```
\@ifpackageloaded{multirow}{% v2.0 or newer
\@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
{% v2.0+:
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
}
{% loaded but older, probably not executed:
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}
}% packageloaded{multirow}
```

If not \@ifpackageloaded{multirow} but \@xmultirow is defined, then this must be v1.6 or earlier, which did not \ProvidesPackage{multirow}, and did not have the vposn option.

```
{% v1.6 or older did not \ProvidePackage
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}
```

The user-level interface. This is provided if the HTML version was not already given.

```
\providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
}% \@xmultirow defined, so multirow was loaded
}% AtBeginDocument
```

```
\end{warpall}
```

## 61.19 Utility macros inside a table

**for HTML output:**

```
\begin{warpHTML}
```

```
\newcommand*{\LWR@donothing}{}
```

In case bigdelim is not loaded:

```
\newcommand*{\ldelim}{}
\newcommand*{\rdelim}{}
```

```
\end{warpHTML}
```

## 61.20 Special-case tabular markers

**for HTML & PRINT:**

```
\begin{warpall}
```

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatcally start a new HTML table data cell yet. See section 8.7.

```
\newcommand*{\TabularMacro}{}
```

```
\end{warpall}
```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

⚠ `tabular` inside another environment

When creating a new environment which contains a `tabular` environment, lwarp's emulation of the tabular does not automatically resume when the containing viroment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

**for HTML output:**

```
\begin{warpHTML}
```

```
\newcommand*{\ResumeTabular}{%
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
\LWR@getmynexttoken%
}
```

```
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
```

```
\newcommand*{\ResumeTabular}{}
```

```
\end{warpprint}
```

## 61.21 Checking for a new table cell

**for HTML output:**

```
\begin{warpHTML}
```

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```
\newbool{LWR@exitingtabular}
```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```
\newbool{LWR@tabularmutemods}
```

\LWR@tabledatacolumntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as \hline, \toprule, etc:

```
\newcommand*{\LWR@tabledatacolumntag}%
{%
\LWR@traceinfo{LWR@tabledatacolumntag}%
```

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

```
\let\mynext\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```
\ifdefequal{\LWR@mynexttoken}{\end}%
    {\global\booltrue{LWR@exitingtabular}}{}%
```

longtable can have a caption in a cell

```
\ifdefequal{\LWR@mynexttoken}{\caption}%
    {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
\ifdefequal{\LWR@mynexttoken}{\multicolumn}%
    {\let\mynext\LWR@donothing}{}%
\ifdefequal{\LWR@mynexttoken}{\multirow}%
    {\let\mynext\LWR@donothing}{}%
\ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
    {\let\mynext\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mrowcell}%
    {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\hline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\toprule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\midrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\cmidrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\specialrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\cline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\bottomrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\ldelim}%
    {\let\mynext\LWR@donothing}{}%
```

```
%
\ifdefequal{\LWR@mynexttoken}{\rdelim}%
    {\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:

```
\mynext%
}
```

```
\end{warpHTML}
```

## 61.22 \mrowcell

**for HTML & PRINT:**

```
\begin{warpall}
```

`\mrowcell` ⚠

The user must insert `\mrowcell` into any `\multirow` cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mrowcell}{}
```

```
\end{warpall}
```

## 61.23 \mcolrowcell

**for HTML & PRINT:**

```
\begin{warpall}
```

`\mcolrowcell` ⚠

The user must insert `\mcolrowcell` into any `\multicolumnrow` cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mcolrowcell}{}
```

```
\end{warpall}
```

## 61.24 New \tabular definition

for HTML output:

```
\begin{warpHTML}
```

These are default defininitions in case booktabs is not loaded, and are not expected to used, but must exist as placeholders.

```
\newcommand*{\LWR@origtoprule}[1]{\hline}
\newcommand*{\LWR@origmidrule}[1]{\hline}
\LetLtxMacro\LWR@origcmidrule\cline
\newcommand*{\LWR@origbottomrule}[1]{\hline}
\newcommand*{\LWR@origaddlinespace}[1][]{}
\newcommand*{\LWR@origmorecmidrules}{}
\newcommand*{\LWR@origspecialrule}[3]{\hline}
```

\LWR@hline The definition of \hline depends on whether tabls has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
\AtBeginDocument{
\@ifpackageloaded{lwarp-tabls}
{
\newcommand*{\LWR@hline}[1][]{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
    {\booltrue{LWR@doinghline}}%
    \LWR@getmynexttoken}%
}
{
\newcommand*{\LWR@hline}{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
    {\booltrue{LWR@doinghline}}%
    \LWR@getmynexttoken}%
}
}% AtBeginDocument
```

\LWR@nullifyNoAutoSpacing For babel-french, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses xstring.

```
\AtBeginDocument{
\@ifundefined{frenchbsetup}%
{% no babel-french
   \newcommand*{\LWR@nullifyNoAutoSpacing}{}
}% no babel-french
{% yes babel-french
    \newcommand*{\LWR@nullifyNoAutoSpacing}{%
```

```
        \NoAutoSpacing%
        \renewcommand*{\NoAutoSpacing}{}%
        \renewcommand*{\LWR@FBcancel}{}%
    }
}% yes babel-french
}% AtBeginDocument
```

Env `LWR@tabular` [⟨*vertposition*⟩] {⟨*colspecs*⟩}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since siunitx might redefine tabular in the user's document.

```
\StartDefiningTabulars

\newenvironment*{LWR@tabular}[2][]
{%
\LWR@traceinfo{LWR@tabular started}%
\addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
\global\boolfalse{LWR@doinghline}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
```

For babel-french, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```
\LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the `@` and `!` columns.

```
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
```

Create the `table` tag:

```
\global\booltrue{LWR@intabularmetadata}%
\LWR@forcenewpage
\LWR@htmlblocktag{table}%
```

Parse the table columns:

```
\LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
\LWR@stoppars%
```

Track column #:

```
\setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
\boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
\LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
\LetLtxMacro{\\}{\LWR@tabularendofline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

The optional parameter for \hline supports the tabls package.

```
\LWR@traceinfo{LWR@tabular: redefining macros}%
\renewcommand*{\hline}{\LWR@hline}%
%
\RenewDocumentCommand{\cline}{m}%
{\LWR@docmidrule{##1}\LWR@getmynexttoken}%
```

```
\DeclareDocumentCommand{\toprule}{o d()}
    {%
        \IfValueTF{##1}%
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doingtbrule}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\midrule}{o d()}%
    {%
        \IfValueTF{##1}%
```

```
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doinghline}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\cmidrule}{O{\LWR@cmidrulewidth} d() m}%
{\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\bottomrule}{o d()}
    {%
        \IfValueTF{##1}%
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doingtbrule}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\addlinespace}{o}{}%
\DeclareDocumentCommand{\morecmidrules}{}{}%
\DeclareDocumentCommand{\specialrule}{m m m d()}%
    {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}\LWR@getmynexttoken}%
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use \LWR@getmynexttoken.

```
\renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
\renewcommand*{\mrowcell}{%
    \LWR@maybenewtablerow%
    \LWR@tabularleftedge%
    \global\booltrue{LWR@skippingmrowcell}%
}%
\renewcommand*{\mcolrowcell}{%
    \LWR@maybenewtablerow%
    \global\booltrue{LWR@skippingmcolrowcell}%
}%
\LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%
```

Reset for new processing:

```
\global\boolfalse{LWR@tableparcell}%
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
\global\boolfalse{LWR@skipatbang}%
\global\boolfalse{LWR@emptyatbang}%
```

Set & for its special meaning inside the tabular:

```
\StartDefiningTabulars%
\protected\gdef&{\LWR@tabularampersand}%
```

Look ahead for a possible table data cell:

```
\LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
\LWR@getmynexttoken%
}%
```

Ending the environment:

```
{%
\LWR@traceinfo{LWR@tabular ending}%
\ifboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {%
            \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
        }%
    )%
}%
{%
    \LWR@tabularfinishrow%
}%
{%
    \LWR@closetabledatacell%
}%
\LWR@htmlblocktag{/tr}%
\LWR@htmlblocktag{/table}%
\global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
\addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
\protected\gdef&{\LWR@origampmacro}%
\EndDefiningTabulars%
\LWR@traceinfo{LWR@tabular finished ending}%
}

\EndDefiningTabulars
```

```
\end{warpHTML}
```

### 61.25 Array

Pkg array

array is also automatically loaded by siunitx.

## 62 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

**for HTML output:**

```
\begin{warpHTML}
```

### 62.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
\providecommand*{\@currentlabelname}{}
```

\LWR@stripperiod {⟨*text*⟩} [⟨.⟩]

Removes a trailing period.

```
\def\LWR@stripperiod#1.\ltx@empty#2\@nil{#1}%
```

\LWR@setlatestname {⟨*object name*⟩}

Removes \label, strips any final period, and remembers the result.

```
\newcommand*{\LWR@setlatestname}[1]{%
```

Table 10: Cross-referencing data structures

**Original LaTeX:** (print and HTML)

**\refstepcounter:** Steps the couunter and sets \@currentlabel.

**\@currentlabel:** \p@<ctr>\the<ctr> Updated by \refstepcounter.

**\label:** Writes to the .aux file:
\newlabel{<label>}{{\@currentlabel}{\thepage}}

**\newlabel:** When the .aux file is read, sets \r@<label>.

**\r@<label>:** Set to: {{\@currentlabel}{\thepage}}

**\ref:** Returns the first part of \r@<label>.

**\pageref:** Returns the second part of \r@<label>.

**Added by lwarp:** (HTML only)

**\label:** Adds HTML tags (section 62.3), plus \splabel data (section 62.2):

**zLWR@name:** The section name for this label.
**zLWR@htmlfilenumer:** The filenumber or name for this label.
**zLWR@lateximagedepth:** The lateximagedepth for this label.
**zLWR@lateximagenumber:** The lateximagenumber for this label.

**\nameref:** Emualted from hyperref for lwarp. See section 62.4.

**\ref and \nameref:** Adds HTML tags. See section 62.4.

**Added by amsmath:** (print and HTML)

**\label:** Execution is delayed until the math environment is completed.

**\ltx@label:** LaTeX \label, (HTML: patched by lwarp,) later patched by cleveref.

**Added by cleveref:** (print and HTML)

\refstepcounter**:** Added: sets \cref@currentlabel.

**\cref@currentlabel:** (<type>=<ctr> unless an alias is used):
[<type>] [\arabic{<ctr>}] [<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 48.4 for use with footnotes.

**\label:** Writes to the .aux file:
\newlabel{<label>@cref}{{\cref@currentlabel}{\thepage}}

**\newlabel:** (Unchanged.) When the .aux file is read, sets \r@<label>@cref.

**\r@<label>@cref:** Set to: {{\cref@currentlabel}{\thepage}}

**Utility functions:** See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.

**Cross-referencing names:** \crefname and \Crefname assign human-readable names for references to this counter type.

**Additionally patched by lwarp:** (HTML only)

**\cref, etc.:** Modified for lwarp. See section 74.

**\label inside math:** See section 67.5.1.

**Footnotes:** See \noteentry in section 48.4.

Remove `\label` and other commands from the name, the strip any final period. See zref-titleref and gettitlestring.

```
\GetTitleStringExpand{#1}%
\edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
\edef\@currentlabelname{%
\expandafter\LWR@stripperiod\@currentlabelname%
\ltx@empty.\ltx@empty\@nil%
}%
}
```

## 62.2 Zref setup

See:
http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference

Create a new property list called special:

```
\zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
\zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
\zref@newprop{zLWR@htmlfilenumber}{%
\ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
}%
```

Additional properties for lateximages:

```
\zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
\zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

`zLWR@htmlfilenumber` property holds the file number or name

Add a `LWR@htmlfilenumber` property, and lateximage properties to special:

```
\zref@addprop{special}{zLWR@name}
\zref@addprop{special}{zLWR@htmlfilenumber}
\zref@addprop{special}{zLWR@lateximagedepth}
\zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
\newcommand*{\LWR@spref}[2]{%
\zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` {⟨*label*⟩} Returns the section name for this label:

```
\newcommand*{\LWR@nameref}[1]{%
\LWR@spref{#1}{zLWR@name}%
}
```

`\LWR@htmlfileref` {⟨*label*⟩} Returns the file number for this label:

```
\newcommand*{\LWR@htmlfileref}[1]{%
% DO NOT USE \LWR@traceinfo HERE!  Will be expanded.
\LWR@spref{#1}{zLWR@htmlfilenumber}%
}
```

`\LWR@lateximagedepthref` {⟨*label*⟩} Returns the `lateximagedepth` for this label:

```
\newcommand*{\LWR@lateximagedepthref}[1]{%
\LWR@spref{#1}{zLWR@lateximagedepth}%
}
```

`\LWR@lateximagenumberref` {⟨*label*⟩} Returns the `lateximagenumber` for this label:

```
\newcommand*{\LWR@lateximagenumberref}[1]{%
\LWR@spref{#1}{zLWR@lateximagenumber}%
}
```

`\LWR@splabel` {⟨*label*⟩} Sanitize the name and then creates the label:

```
\newcommand*{\LWR@splabel}[1]{%
\LWR@setlatestname{\@currentlabelname}%
\zref@labelbylist{#1}{special}}
```

## 62.3 Labels

`\LWR@subsublabel` {⟨*label*⟩} Creates an HTML id tag.

```
\newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
    \ifbool{LWR@doingstartpars}%
    {% pars allowed
        \ifbool{LWR@doingapar}
        {% par started
            \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
        }% par started
        {% par not started
            \LWR@stoppars%
            \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
            \LWR@startpars%
        }% par not started
    }% pars allowed
    {% pars not allowed
        \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
    }% pars not allowed
}% not lateximage
}
```

\LWR@newlabel {⟨*label*⟩} [⟨*type*⟩]

\label during HTML output when not in math mode, removing extra spaces around the label, as done by regular LaTeX \label.

clevereref later encases this to add its own cross-referencing.

The optional ⟨*type*⟩ is per the ntheorem package, and is ignored.

```
\NewDocumentCommand{\LWR@newlabel}{m o}{%
\LWR@traceinfo{LWR@newlabel: starting}%
\LWR@traceinfo{LWR@newlabel: !#1!}%
% \@bsphack%
```

Create a traditional LaTeX label, as modified by cleveref:

```
\LWR@origlabel{#1}%
```

Create a special label which holds the section number, LWR@htmlfilenumber, LWR@lateximagedepth, and LWR@lateximagenumber:

```
\LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
\LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
\LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
\LWR@splabel{#1}%
\LWR@subsublabel{#1}%
% \@esphack%
\LWR@traceinfo{LWR@newlabel: done}%
}
```

## 62.4 References

\LWR@startref {⟨*label*⟩} (Common code for \ref and \nameref.)

Open an HTML tag reference to a filename, # character, and a label.

```
\newcommand*{\LWR@startref}[1]
{%
\edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
\LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
\LWR@htmltag{a href="%
\LWR@traceinfo{LWR@startref B}%
\LWR@htmlrefsectionfilename{#1}%
\LWR@traceinfo{LWR@startref C}%
\#%
```

Create the destination id:

See if LWR@lateximagedepth is unknown:

```
\LWR@traceinfo{LWR@startref D: !#1!}%
\ifthenelse{\equal{\LWR@lidref}{??}}%
```

"??" if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```
{%
    \LWR@traceinfo{LWR@startref D0: ??}%
    ??%
}%
```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
{%
    \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
    \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
    {%
        \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
        lateximage\LWR@lateximagenumberref{#1}%
    }%
    {%
        \LWR@traceinfo{LWR@startref D3}%
        #1%
    }%
}%
\LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
"{}}%
\LWR@traceinfo{LWR@startref F}%
}
```

`\LWR@subnewref` {⟨*label*⟩} {⟨*label or sub@label*⟩}

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as "`1(b)`".

```
\NewDocumentCommand{\LWR@subnewref}{m m}{%
\LWR@traceinfo{LWR@subnewref #1 #2}%
\LWR@startref{#1}%
\LWR@origref{#2}%
\LWR@htmltag{/a}%
}
```

`\ref` * {⟨*label*⟩} `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * {⟨*label*⟩} Create an internal document reference link, or without a link if starred per hyperref.

```
\NewDocumentCommand{\LWR@newref}{s m}{%
\LWR@traceinfo{LWR@newref #2}%
\IfBooleanTF{#1}%
{\LWR@origref{#2}}%
{\LWR@subnewref{#2}{#2}}%
}
```

`\pagerefPageFor` Text for page references.

```
\newcommand*{\pagerefPageFor}{see }
```

`\pageref` * {⟨*label*⟩} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
\NewDocumentCommand{\LWR@newpageref}{s m}{%
\IfBooleanTF{#1}%
{(\pagerefPageFor\LWR@origref{#2})}%
{(\cpageref{#2})}%
}
```

`\nameref` {⟨*label*⟩}

```
\DeclareRobustCommand*{\nameref}[1]{%
\LWR@traceinfo{nameref A}%
\LWR@startref{#1}%
\LWR@traceinfo{nameref B}%
\LWR@nameref{#1}%
\LWR@traceinfo{nameref C}%
\LWR@htmltag{/a}%
\LWR@traceinfo{nameref D}%
}
```

`\Nameref` {⟨*label*⟩} In print, adds the page number. In HTML, does not.

```
\let\Nameref\nameref
```

## 62.5 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.

```
% DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
% \EmulatesPackage{hyperref}[2015/08/01]% Disabled.  Do not do this.
```

Create a link with a text name:

\LWR@subhyperref {⟨*URL*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@subhyperref}{m +m}{%
\LWR@htmltag{a href="{#1}" target="\_{}blank"\LWR@orignewline}#2\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\LWR@subhyperrefclass {⟨*URL*⟩} {⟨*text*⟩} {⟨*htmlclass*⟩}

```
\NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
\LWR@htmltag{a href="{#1}"
class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\href [⟨*options*⟩] {⟨*URL*⟩} {⟨*text*⟩}

Create a link with accompanying text:

```
\NewDocumentCommand{\LWR@hrefb}{O{} m +m}{%
\LWR@subhyperref{#2}{#3}%
\endgroup%
\LWR@ensuredoingapar%
}

\DeclareRobustCommand*{\href}{%
\LWR@ensuredoingapar%
\begingroup%
\catcode‘\_=12
\LWR@hrefb%
}
```

\nolinkurl {⟨*URL*⟩}

Print the name of the link without creating the link:

```
\newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}

\DeclareRobustCommand*{\nolinkurl}{%
\LWR@ensuredoingapar%
\begingroup\catcode‘\_=12
\LWR@nolinkurlb%
}
```

\url {⟨*URL*⟩}

Create a link whose text name is the address of the link. The url package may redefine `\url`, so it is `\let` to `\LWR@urla`here and also redefined by lwarp-url.

```
\NewDocumentCommand{\LWR@urlb}{m}{%
\LWR@hrefb{#1}{#1}%
}

\DeclareRobustCommand*{\url}{%
\LWR@ensuredoingapar%
\begingroup\catcode`\_=12
\LWR@urlb%
}

% \LetLtxMacro\url\LWR@urla
```

`\LWR@subinlineimage` [⟨*alttag*⟩] {⟨*class*⟩} {⟨*filename*⟩} {⟨*extension*⟩} {⟨*style*⟩}

```
\newcommand*{\LWR@subinlineimage}[5][]{%
\ifblank{#1}%
{\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
{\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
}
```

```
\end{warpHTML}
```

Table 11: Float data structures

For each <type> of float (figure, table, etc.) there exists the following:

**counter <type>:** A counter called <type>, such as `figure`, `table`.

**`\<type>name`:** Name. `\figurename` prints “Figure”, etc.

**`\ext@<type>`:** File extension. `\ext@figure` prints “lof”, etc.

**`\fps@<type>`:** Placement.

**`\the<type>`:** Number. `\thetable` prints the number of the table, etc.

**`\p@<type>`:** Parent’s number. Prints the number of the [within] figure, etc.

**`\fnum@<type>`:** Prints the figure number for the caption.
`\<type>name \the<type>`, “Figure 123”.

**`\<type>`:** Starts the float environment. `\figure` or `\begin{figure}`

**`\end<type>`:** Ends the float environment. `\endfigure` or `\end{figure}`

**`\tf@<ext>`:** The LaTeX 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.

## 63 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each <type> of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the babel package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

## 63.1 Float captions

for HTML output:

```
\begin{warpHTML}
```

\LWR@floatbegin {⟨*type*⟩} [⟨*placement*⟩]

Begins a \newfloat environment.

```
\NewDocumentCommand{\LWR@floatbegin}{m o}{%
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
\addtocounter{LWR@thisautoid}{1}%
\booltrue{LWR@freezethisautoid}%
```

```
\begingroup%
```

Settings while inside the environment:

```
\LWR@origraggedright%
```

Open an HTML figure tag:

```
\LWR@htmltag{figure id="autoid-\arabic{LWR@thisautoid}" class="#1"}%
\ifbool{FormatWP}{%
    \LWR@orignewline%
    \LWR@BlockClassWP{}{}{wp#1}%
}{}%
```

```
\renewcommand*{\@captype}{#1}%
\caption@settype{#1}%
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== begin #1 ===

}{}%
}
```

\@float
\@dlbfloat

Support packages which create floats directly.

```
\let\@float\LWR@floatbegin
\let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
\newcommand*{\LWR@floatend}{%
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== end ===

}{}%
\LWR@stoppars%
```

Close an HTML figure tag:

```
\ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
\LWR@htmlelementend{figure}%
\endgroup%
\boolfalse{LWR@freezethisautoid}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
}
```

\end@float
\end@dlbfloat Support packages which create floats directly.

```
\let\end@float\LWR@floatend
\let\end@dblfloat\LWR@floatend
```

Ctr LWR@thisautoid A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```
\newcounter{LWR@thisautoid}
```

Bool LWR@freezethisautoid Prevents multiple increments of \LWR@thisautoid inside a float.

```
\newbool{LWR@freezethisautoid}
\boolfalse{LWR@freezethisautoid}
```

\LWR@maybeincthisautoid Possibly increments the autoid counter.

```
\newcommand*{\LWR@maybeincthisautoid}{%
\ifbool{LWR@freezethisautoid}{}{\addtocounter{LWR@thisautoid}{1}}%
}
```

\LWR@newautoidanchor Adds a new <autoid> anchor.

```
\newcommand*{\LWR@newautoidanchor}{%
\LWR@maybeincthisautoid%
\LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"{}}\LWR@htmltag{/a}%
}
```

\@captype Remembers which float type is in use.

```
\newcommand*{\@captype}{}
```

### 63.1.1 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
\AtBeginDocument{\providecommand*{\CaptionSeparator}{:~}}
```

\@makecaption {⟨*name and num*⟩} {⟨*text*⟩}

Prints the float type and number, the caption separator, and the caption text.

```
\AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

### 63.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the LaTeX PDF file, the LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special LaTeX labels). This page number is used to generate an `autopage` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autoid` `<id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr LWR@nextautoid Tracks autoid for floats. Tracks autopage for floats.

Ctr LWR@nextautopage These are updated per float as the `.lof` file is read.

```
\newcounter{LWR@nextautoid}
\newcounter{LWR@nextautopage}
```

\LWRsetnextfloat {⟨*autopage*⟩} {⟨*autoid*⟩}

This is written to the `.lof` file just before each float's usual entry. The autopage and autoid are remembered for `\l@figure` to use when creating the HTML links.

```
\newcommand*{\LWRsetnextfloat}[2]{%
\setcounter{LWR@nextautopage}{#1}%
\setcounter{LWR@nextautoid}{#2}%
}
```

Ctr LWR@latestautopage

Updated each time a new HTML file is begun. \LWRsetnextfloat is written with this and the autoid by the modified \addcontentsline just before each float's entry.

```
\newcounter{LWR@latestautopage}
\setcounter{LWR@latestautopage}{1}
```

Env LWR@figcaption

Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```
\newenvironment*{LWR@figcaption}
{%
\LWR@htmlblocktag{figcaption}
\ifbool{FormatWP}{%
\begin{BlockClass}[font-style:italic]{italic}
\LWR@origvspace*{\baselineskip}
}{}%
}
{%
\ifbool{FormatWP}{\end{BlockClass}}{}%
\LWR@htmlblocktag{/figcaption}%
}
```

```
\let\LWR@origcaption@begin\caption@begin
\let\LWR@origcaption@end\caption@end
\let\LWR@orig@@par\@@par
```

\LWR@caption@begin

Low-level patches to create HTML tags for captions.

```
\newcommand{\LWR@caption@begin}
{
\LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
\begingroup%
```

The caption code was not allowing the closing par tag:

```
\renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
\RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
\RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
\LWR@figcaption%
\LWR@origcaption@begin%
}
```

\LWR@caption@end

Low-level patches to create HTML tags for captions.

```
\newcommand{\LWR@caption@end}
{%
\LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
\LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
\endLWR@figcaption%
\endgroup%
% \leavevmode% avoid bad space factor (0) error
\LWR@traceinfo{LWR@caption@end: done}%
}
```

\caption@begin
\caption@end

Low-level patches to create HTML tags for captions.

```
\AtBeginDocument{
\let\caption@begin\LWR@caption@begin
\let\caption@end\LWR@caption@end
}
```

\captionlistentry

Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
\let\LWR@origcaptionlistentry\captionlistentry

\renewcommand*{\captionlistentry}{%
\LWR@ensuredoingapar%
\LWR@origcaptionlistentry%
}

\def\LWR@LTcaptionlistentry{%
\LWR@ensuredoingapar%
\LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"{}}\LWR@htmltag{/a}%
  \bgroup
  \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
          {\egroup\LWR@LT@captionlistentry}}%
\def\LWR@LT@captionlistentry#1{%
  \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%
```

\addcontentsline Patched to write the autopage and autoid before each float's entry. No changes if writing .toc For a theorem, automatically defines \ext@<type> as needed, to mimic and reuse the float mechanism.

```
\let\LWR@origaddcontentsline\addcontentsline

\renewcommand*{\addcontentsline}[3]{%
\ifstrequal{#1}{toc}{}{% not TOC
    \LWR@newautoidanchor%
    \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}%
    \addtocontents{\@nameuse{ext@#2}}{%
        \protect\LWRsetnextfloat%
        {\arabic{LWR@latestautopage}}%
        {\arabic{LWR@thisautoid}}%
    }%
}% not TOC
\LWR@origaddcontentsline{#1}{#2}{#3}%
}
```

\captionof Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```
\AtBeginDocument{
\let\LWR@origcaptionof\captionof

\renewcommand*{\captionof}{%
\LWR@stoppars
\LWR@origcaptionof%
}
}
```

```
\end{warpHTML}
```

# 64 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The .toc, .lof, and .lot files are named by the source code \jobname.

In HTML, the printed tables are placed inside a <div> of class toc, lof, or lot.

A "sidetoc" is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular LaTeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output:

```
\begin{warpHTML}
```

## 64.1 Reading and printing the TOC

`\LWR@myshorttoc` {⟨*toc/lof/lot*⟩}

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular LaTeX, the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideTOC.

```
\newcommand*{\LWR@myshorttoc}[1]{
\LWR@ensuredoingapar
```

Only if the file exists:

```
\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`.

```
\begingroup
% \ifxetexorluatex%
% \else
% \inputencoding{latin1}% currently disabled
% \fi
\makeatletter
```

Read in the TOC file:

```
\@input{\jobname.#1}
% \makeatother
\endgroup
}%
{}%
}
```

`\LWR@subtableofcontents` {⟨*toc/lof/lot*⟩} {⟨*sectionstarname*⟩}

Places a TOC/LOF/LOT at the current position.

```
\NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
\@ifundefined{chapter}
{\LWR@closeprevious{\LWR@depthsection}}
{\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
\LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
\@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML `nav` containing the TOC/LOF/LOT:

```
\LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
\LWR@myshorttoc{#1}
```

Close the `nav`:

```
\LWR@htmlelementclassend{nav}{#1}
}
```

`\@starttoc` {⟨*ext*⟩}

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
\let\LWR@orig@starttoc\@starttoc

\renewcommand{\@starttoc}[1]{
\LWR@htmlelementclass{nav}{#1}
\LWR@orig@starttoc{#1}
\LWR@htmlelementclassend{nav}{#1}
}
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. newfloat uses `\listoffigures` for all future float types.

```
\let\LWR@origtableofcontents\tableofcontents
```

```

\renewcommand*{\tableofcontents}{%
```

Do not print the table of contents if formatting for a word processor, which will presumably auto-generate its own updated table of contents:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{

=== table of contents ===

}
{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
    \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
    \LWR@printpendingfootnotes
    \LWR@origtableofcontents
}
}
```

\listoffigures

```
\let\LWR@origlistoffigures\listoffigures

\renewcommand*{\listoffigures}{
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{

=== list of figures ===

}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoffigures
}
}
```

\listoftables

```
\let\LWR@origlistoftables\listoftables

\renewcommand*{\listoftables}{
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{

=== list of tables ===

}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoftables
}
}
```

## 64.2 High-level TOC commands

`\listof` {⟨*type*⟩} {⟨*title*⟩}

Emulate the `\listof` command from the float package (section 141). Used to create lists of custom float types. Also used to redefine the standard LaTeX `\listoffigures` and `\listoftables` commands.

```
\NewDocumentCommand{\listof}{m +m}{%
\LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
\expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
\immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
    \jobname.\csuse{ext@#1}\relax
}
```

## 64.3 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.

```
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

Ctr `SideTOCDepth` Controls how deep the side-TOC gets. Use a standard LaTeX section level similar to tocdepth.

```
\newcounter{SideTOCDepth}
\setcounter{SideTOCDepth}{1}
```

\sidetocname Holds the default name for the sideTOC.

```
\newcommand{\sidetocname}{Contents}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

\LWR@sidetoc Creates the actual side-TOC.

```
\newcommand*{\LWR@sidetoc}{
\LWR@forcenewpage
\LWR@stoppars

```

The entire sideTOC is placed into a nav of class sidetoc.

```
\LWR@htmlelementclass{nav}{sidetoc}

\setcounter{tocdepth}{\value{SideTOCDepth}}

```

The title is placed into a <div> of class sidetoctitle, and may contain paragraphs.

```
\begin{BlockClass}{sidetoctitle}
\sidetocname
\end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoccontents.

```
\begin{BlockClass}{sidetoccontents}
\LinkHome

\LWR@myshorttoc{sidetoc}
\end{BlockClass}
\LWR@htmlelementclassend{nav}{sidetoc}
}
```

## 64.4 Low-level TOC line formatting

\numberline {⟨*number*⟩}

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
\newcommand*{\LWR@numberline}[1]{%
\LWR@sectionnumber{#1}\quad%
}

\LetLtxMacro\numberline\LWR@numberline
```

`\hypertoc` {⟨*1: depth*⟩} {⟨*2: type*⟩} {⟨*3: name*⟩} {⟨*4: page*⟩}

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

**#1** is depth

**#2** is `section`, `subsection`, etc.

**#3** the text of the caption

**#4** page number

```
\NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
\ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
    \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
    \LWR@stoppars%
}
{}
}
```

Ctr `lofdepth` TOC depth for figures.

```
\newcounter{lofdepth}
\setcounter{lofdepth}{1}
```

Ctr `lotdepth` TOC depth for tables.

```
\newcounter{lotdepth}
\setcounter{lotdepth}{1}
```

`\hypertocfloat` {⟨*1: depth*⟩} {⟨*2: type*⟩} {⟨*3: ext of parent*⟩} {⟨*4: caption*⟩} {⟨*5: page*⟩}

**#1** is depth

**#2** is `figure`, `table`, etc.

**#3** is `lof`, `lot`, of the parent.

**#4** the text of the caption

**#5** page number

```
\newcommand{\hypertocfloat}[5]{%
\LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
\@ifundefined{c@#3depth}{%
\newcounter{#3depth}%
\setcounter{#3depth}{1}%
}{}%
```

Respond to `lofdepth`, etc.:

```
\LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
\ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
    \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
    \#autoid-\arabic{LWR@nextautoid}}%
    {#4}{toc#2}%
    \LWR@stoppars%
}{}%
}
```

Automatically called by `\contentsline`:

```
\renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
\DeclareDocumentCommand{\l@chapter}{m m}
    {\hypertoc{0}{chapter}{#1}{#2}}
\renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
\renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
\renewcommand{\l@subsubsection}[2]
    {\hypertoc{3}{subsubsection}{#1}{#2}}
\renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
\renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
\renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
\renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

```
\end{warpHTML}
```

# 65 Index and glossary

See:
http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx

Index links are tracked by the counter LWR@autoindex. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

**for HTML output:**

```
\begin{warpHTML}
```

```
\newcounter{LWR@autoindex}
\setcounter{LWR@autoindex}{0}

\newcounter{LWR@autoglossary}
\setcounter{LWR@autoglossary}{0}
```

\printindex

```
\let\LWR@origprintindex\printindex

\renewcommand*{\printindex}
{
\LWR@startpars
\LWR@origprintindex
}
```

Env theindex

```
\@ifundefined{chapter}
{\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
{\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}

\renewenvironment*{theindex}{%
\LWR@indexsection{\indexname}%
\let\item\LWR@indexitem%
\let\subitem\LWR@indexsubitem%
\let\subsubitem\LWR@indexsubsubitem%
}{}
```

`\LWR@indexitem`

```
\newcommand{\LWR@indexitem}{

\InlineClass{indexitem}{}
}
```

`\LWR@indexitem`

```
\newcommand{\LWR@indexsubitem}{

\InlineClass{indexsubitem}{}
}
```

`\LWR@indexitem`

```
\newcommand{\LWR@indexsubsubitem}{

\InlineClass{indexsubsubitem}{}
}
```

`\@wrindex` {⟨*term*⟩} Redefined to write the `LWR@latestautopage` counter instead of `page`

```
\def\LWR@wrindex#1{%
\addtocounter{LWR@autoindex}{1}%
\LWR@newlabel{LWRindex-\theLWR@autoindex}%
\protected@write\@indexfile{}%
{\string\indexentry{#1}{\theLWR@autoindex}}%
\endgroup
\@esphack}

\let\@wrindex\LWR@wrindex
```

`\@wrglossary` {⟨*term*⟩} Redefined to write the `LWR@latestautopage` counter instead of `page`

```
\def\@wrglossary#1{%
\addtocounter{LWR@autoglossary}{1}%
\LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
\protected@write\@glossaryfile{}%
{\string\glossaryentry{#1}{\theLWR@autoglossary}}%
\endgroup
\@esphack}
```

`\hyperindexref` {⟨*autosecnumber*⟩}

`\hyperindexref{web address}` is inserted into `*.ind` by the xindy style file `lwarp.xdy`

```
\newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}
```

```
\end{warpHTML}
```

**for PRINT output:** A null command for print mode, in case hyperref was not used:

```
\begin{warpprint}
\newcommand{\hyperindexref}[1]{#1}
\end{warpprint}
```

**for HTML & PRINT:** For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
\begin{warpall}
\providecommand{\glo@name}{}
\end{warpall}
```

## 66 Restoring original formatting

`\LWR@restoreorigformatting` Used to temporarily restore the print-mode meaing of a number of formatting, graphics, and symbols-related macros while generating SVG math or a `lateximage`. A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

```
\newcommand*{\LWR@restoreorigformatting}{%
\LWR@traceinfo{LWR@restoreorigformatting}%
\linespread{1}%
```

```
\LetLtxMacro\ref\LWR@origref%{} syntax highlighting
\RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
\RenewDocumentEnvironment{BlockClass}{o m}{}{}%
\renewcommand{\BlockClassSingle}[2]{##2}%
```

```
\LetLtxMacro{\hspace}{\LWR@orighspace}%
\LetLtxMacro\rule\LWR@origrule%
\let\,\LWR@origcomma% disable HTML short unbreakable space
\let\textellipsis\LWR@origtextellipsis%
\let\textless\LWR@origtextless%
\let\textgreater\LWR@origtextgreater%
\LetLtxMacro{\textrm}{\LWR@origtextrm}%
\LetLtxMacro{\textsf}{\LWR@origtextsf}%
\LetLtxMacro{\texttt}{\LWR@origtexttt}%
\LetLtxMacro{\textbf}{\LWR@origtextbf}%
\LetLtxMacro{\textmd}{\LWR@origtextmd}%
\LetLtxMacro{\textit}{\LWR@origtextit}%
\LetLtxMacro{\textsl}{\LWR@origtextsl}%
\LetLtxMacro{\textsc}{\LWR@origtextsc}%
\LetLtxMacro{\textup}{\LWR@origtextup}%
\LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
\LetLtxMacro{\emph}{\LWR@origemph}%
\LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
\LetLtxMacro{\sffamily}{\LWR@origsffamily}%
\LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
\LetLtxMacro{\bfseries}{\LWR@origbfseries}%
\LetLtxMacro{\mdseries}{\LWR@origmdseries}%
\LetLtxMacro{\upshape}{\LWR@origupshape}%
\LetLtxMacro{\slshape}{\LWR@origslshape}%
\LetLtxMacro{\scshape}{\LWR@origscshape}%
\LetLtxMacro{\itshape}{\LWR@origitshape}%
\LetLtxMacro{\em}{\LWR@origem}%
\LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
\let\sp\LWR@origsp%
\let\sb\LWR@origsb%
\LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
\LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
\LetLtxMacro\textsubscript\LWR@origtextsubscript%
\LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
\LetLtxMacro\underline\LWR@origunderline%
\let~\LWR@origtilde%
\let\enskip\LWR@origenskip%
\let\quad\LWR@origquad%
\let\qquad\LWR@origqquad%
\LetLtxMacro{\tabular}{\LWR@origtabular}%
\LetLtxMacro{\endtabular}{\LWR@origendtabular}%
\LetLtxMacro\toprule\LWR@origtoprule%
\LetLtxMacro\midrule\LWR@origmidrule%
\LetLtxMacro\cmidrule\LWR@origcmidrule%
\LetLtxMacro\bottomrule\LWR@origbottomrule%
\LetLtxMacro\addlinespace\LWR@origaddlinespace%
\LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
\LetLtxMacro\specialrule\LWR@origspecialrule%
\let\newline\LWR@orignewline%
\LetLtxMacro{\raisebox}{\LWR@origraisebox}%
```

```
\LetLtxMacro\includegraphics\LWR@origincludegraphics%
\LetLtxMacro{\scalebox}{\LWR@origscalebox}%
\LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
\let\reflectbox\LWR@origreflectbox%
\LetLtxMacro\resizebox\LWR@origresizebox%
\let\framebox\LWR@origframebox%
\let\makebox\LWR@origmakebox%
\let\fbox\LWRprint@fbox%
\let\fboxBlock\LWRprint@fbox%
\LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
\LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
\LetLtxMacro{\minipage}{\LWR@origminipage}%
\let\endminipage\LWR@endminipage%
\LetLtxMacro{\parbox}{\LWR@origparbox}%
\let\TeX\LWR@origTeX%
\let\LaTeX\LWR@origLaTeX%
\let\LaTeXe\LWR@origLaTeXe%
\renewcommand*{\Xe}{X\textsubscript{E}}%
```

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath%
```

```
\csletcs{equation*}{LWR@origequationstar}%
\csletcs{endequation*}{LWR@origendequationstar}%
%
\LWR@restoreorigaccents%
\LWR@restoreoriglists%
%
\LWR@FBcancel%
}
```

# 67 Math

## 67.1 Limitations

### 67.1.1 Rendering tradeoffs

Math rendering

Math may be rendered as SVG graphics or using the MathJax JavaScript display engine.

SVG files

In its current implementation, rendering math as images creates a new SVG file for each expression. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and check-summing may be used to remove the need for duplicate files.

SVG inline

Another approach could be to in-line the SVG files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files

Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML

Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

### 67.1.2 SVG option

SVG math option

For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[13], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size

The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

SVG math copy/paste

For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the LaTeX code for the math expression. For AMS environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred AMS environment, although the `\tag` macro will still appear inside the LaTeX math expression.

### 67.1.3 MathJax option

MathJax math option

Prog `MathJax`

The popular MathJax alternative (`mathjax.org`) may be used to display math.

When MathJax is enabled, math is rendered twice:

1. As regular LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of LaTeX, and

[13] See section 273 regarding fonts and fractions.

2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

MathJax limitations
Prog `MathJax`

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, `\tag`ged equations have the chapter number prepended in HTML output, unlike LaTeX. `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

  `https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY`

subequations

- MathJax itself does not support subequations. This may be improved by parsing the LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a lateximage, and therefore also inside a Ti*k*z or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

⚠ siunitx inside an equation

- Usage of siunitx inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

  `https://github.com/burnpanck/MathJax-siunitx`

  Also see section 8.5.5.

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, bigdelim, units, and nicefrac, along with occasionally-used macros such as `\footnote` and `\relax`.

custom MathJax macros

- MathJax does not automatically support custom LaTeX macros, but they may be created by the user inside a math expression:

```
\begin{document}
(...)
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
    % New macros for MathJax are
    % placed inside a math expression:
    \(
        \newcommand{\expval}[1]{\langle#1\rangle}
        \newcommand{\abs}[1]{\lvert#1\rvert}
    \)
}{}
\end{warpHTML}
```

## 67.2 Inline and display math

**for HTML output:**

```
\begin{warpHTML}
```

\$ Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar \$, print it inside a span to avoid it being interpreted by MathJax, unless are inside a lateximage, in which case it will not be seen by MathJax.

```
\let\LWR@origtextdollar\$

\renewcommand*{\$}{%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{\LWR@origtextdollar}%
{\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
}
```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
\newcounter{LWR@externalfilecnt}
```

```
\LetLtxMacro\LWR@origdollar$
\LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting
```

```
\LetLtxMacro\LWR@origopenparen\(
\LetLtxMacro\LWR@origcloseparen\)
\LetLtxMacro\LWR@origopenbracket\[
\LetLtxMacro\LWR@origclosebracket\]
```

$ $$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:

```
\begingroup
\catcode`\$=\active%
\protected\gdef${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
\protected\gdef\LWR@doubledollar$#1$${
```

If MathJax or formatting for a word processor, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between \[ and \]:

```
{

        \textbackslash[%
        \LWR@HTMLsanitize{#1}%
        \textbackslash]

}% mathjax
```

For SVG, print the math inside a lateximage, with an <alt> tag of the LaTeX code:

```
{% not mathjax

        \begin{lateximage}%
        [\textbackslash{[} \LWR@HTMLsanitize{#1} \textbackslash{]}]%
        \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
        \end{lateximage}%

}% not mathjax
}%
```

\LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
\protected\gdef\LWR@subsingledollar#1{%
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between \( and \):

```
{%
        {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}%
}% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the LaTeX code:

```
{% not mathjax
        \begin{lateximage}%
        [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
        \LWR@origdollar#1\LWR@origdollar%
        \end{lateximage}%
}%
}

\protected\gdef\LWR@singledollar#1${%
\LWR@subsingledollar{#1}%
}
```

`\(` `\)` Redefine to the above dollar macros.

```
\protected\gdef\(#1\){%
$#1$}
\protected\gdef\[#1\]{$$#1$$}

\endgroup



```

`\@ensuredmath` {⟨*expression*⟩} Not yet adapted to lwarp.

```
\LetLtxMacro\LWR@origensuredmath\@ensuredmath

\renewcommand{\@ensuredmath}[1]{%
\ifmmode%
    \LWR@origensuredmath{#1}%
\else%
    \LWR@subsingledollar{\relax#1}%
\fi%
}
```

Remove the old `math` and `displaymath` environments:

```
\let\math\relax
\let\endmath\relax
\let\displaymath\relax
\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`.

```
\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.

```
\NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

## 67.3 MathJax support

Ctr LWR@nextequation Used to add one to compute the next equation number.

```
\newcounter{LWR@nextequation}
```

\LWR@syncmathjax Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside "\(" and "\)" characters. They are printed to HTML output, not interpreted by LaTeX.

```
\newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
\ifcsdef{thechapter}{
\InlineClass{hidden}{
\textbackslash(
\textbackslash{}seteqsection \{\thechapter\}
\textbackslash)
}
}
{}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
\ifthenelse{\cnttest{\value{equation}}>0}
{
```

Tell MathJax that the next set of equations begins with the current LaTeX equation number, plus one.

```
    \setcounter{LWR@nextequation}{\value{equation}}
    \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “`\(`” and “`\)`” characters, to be printed to HTML, not interpreted by LaTeX.

```
    \InlineClass{hidden}{
        \textbackslash(
        \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
        \textbackslash)
    }
}{}% not eq > 0
}
```

`\LWR@hidelatexequation` {⟨*environment*⟩} {⟨*contents*⟩}

Creates the LaTeX version of the equation inside an HTML comment.

```
\NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
\LWR@stoppars
\LWR@htmlopencomment

```

Start the LaTeX math environment inside the HTML comment:

```
\begingroup
\csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their LaTeX meanings.

```
\LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 67.5.1.

Print the contents of the equation:

```
#2
```

End the LaTeX math environment inside the HTML comment:

```
\csuse{LWR@origend#1}
\endgroup

```

Close the HTML comment and resume HTML paragraph handling:

```
\LWR@htmlclosecomment
\LWR@startpars
```

```
}
```

\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 LaTeX.

```
\NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed "`\(`" and "`\)`" characters.

```
\LWR@origtilde\LWR@orignewline
\textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
\textbackslash{}end\{#1\}
\LWR@orignewline
}
```

## 67.4 Equation environment

Remember existing `equation` environment:

```
\let\LWR@origequation\equation
\let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
\let\equation\relax
\let\endequation\relax
```

Env equation The new `equation` environment is created with `\NewEnviron` (from the environ package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the LaTeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\label`s. The contents are also printed to the HTML output for processing by the MathJax script.

```
\NewEnviron{equation}{%

```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

MathJax output:

```
{
```

Print commands to syncronize MathJax's equation number and format to the current LaTeX chapter/section and equation number:

```
    \LWR@syncmathjax
```

Print the LaTeX math inside an HTML comment:

```
    \LWR@hidelatexequation{equation}{\BODY}
}
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the LaTeX equation environment commmands, and the contents of the environment's `\BODY`.

```
{% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
    \begin{lateximage}[(\LWR@equationtag) \textbackslash{begin\{equation\}} %
    \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %
    \textbackslash{end\{equation\}}]% alt tag
```

Create the actual LaTeX-formatted equation inside the `lateximage` using the contents of the environment.

```
    \LWR@origequation
    \BODY% contents collected by NewEnviron
    \LWR@origendequation
    \end{lateximage}%
}% not mathjax

```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
}[%
    \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
    {%
        \LWR@addmathjax{equation}{\BODY}%
    }{}%

]
```

Env equation*

```
\csletcs{LWR@origequationstar}{equation*}
\csletcs{LWR@origendequationstar}{endequation*}
\renewenvironment*{equation*}
{\displaymath}
{\enddisplaymath}
```

## 67.5 AMS Math environments

### 67.5.1 Support macros

Bool LWR@amsmultline True if processing a multline environment.

To compensate for multline-spefific code, LWR@amsmultline is used to add extra horizontal space in \LWR@htmlmathlabel if is used in an amsmath environment which is not a multline environment and not an equation.

```
\newbool{LWR@amsmultline}
\boolfalse{LWR@amsmultline}
```

\LWR@htmlmathlabel {⟨*label*⟩}

lwarp points \ltx@label here. This is used by \label when inside a LaTeX AMS math environment's math display environment.

\LWR@origltx@label points to the LaTeX original, modified by lwarp, then by amsmath, then by cleveref.

```
\newcommand*{\LWR@htmlmathlabel}[1]{%
\LWR@traceinfo{LWR@htmlmathlabel #1}%
```

If mathjax or FormatWP, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

The combined LaTeX & HTML label is printed in a `\text` field:

```
    \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
    \ifbool{LWR@amsmultline}{}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the amsmath package.)

```
    \LWR@htmlclosecomment%
    \LWR@origltx@label{#1}%
    \LWR@htmlopencomment%
    }% text
}% mathjax
{%
    \LWR@origltx@label{#1}%
}%
}
```

`\LWR@beginhideamsmath` Starts hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@beginhideamsmath}{
\LWR@stoppars
\LWR@origtilde\LWR@orignewline
\LWR@htmlopencomment

\begingroup
\LWR@restoreorigformatting
}
```

`\LWR@endhideamsmath` Ends hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@endhideamsmath}{
\endgroup

\LWR@htmlclosecomment
\LWR@orignewline
\LWR@startpars
}
```

### 67.5.2 Environment patches

The following amsmath environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an AMS package, and thus requires special handling.

For SVG math: Each envrionment is encapsulated inside a `lateximage` environment, along with a special optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the AMS environment.

For MathJax: Each environment is syched with LaTeX's equation numbers, typeset with LaTeX inside an HTML comment, then printed to HTML output for MathJax to process.

Env `eqnarray` This environmnet is not an AMS environment and thus its body is not automatically captured, so the environ package is used to capture the environment into `\BODY`.

```
\let\LWR@origeqnarray\eqnarray
\let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
\newbool{LWR@numbereqnarray}
\booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from fancybox):

```
\newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

If MathJax, the environment contents (the `\BODY`) are executed in a HTML comment to trigger the correct equation number increment (if not starred), then are included verbatim in the output for MathJax to interpret:

```
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \ifbool{LWR@numbereqnarray}
    {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
        \LWR@beginhideamsmath
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
        \LWR@addmathjax{eqnarray}{\BODY}
    }%
    {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
    \LWR@addmathjax{eqnarray*}{\BODY}
    }% LWR@numbereqnarray
}% mathjax
{% not mathjax
    \ifbool{LWR@numbereqnarray}
    {
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
        \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
            \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \end{lateximage}
    }%
    {% not LWR@numbereqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```
        \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
        \LWR@origeqnarray
        \BODY
        \nonumber
        \LWR@origendeqnarray
        \end{lateximage}
    }% LWR@numbereqnarray
}% not mathjax
```

Default to number equations in the future:

```
\booltrue{LWR@numbereqnarray}
}
```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```
\RenewEnviron{eqnarray}
{%

\LWR@eqnarrayfactor

}
```

The starred version is patched to turn off the numbering:

```
\csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env multline

```
\BeforeBeginEnvironment{multline}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
}
}

\AfterEndEnvironment{multline}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env multline*

```
\BeforeBeginEnvironment{multline*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{multline*}]
}
}

\AfterEndEnvironment{multline*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline*}{\the\@envbody}
}
{\end{lateximage}}

}

```

Env gather

```
\BeforeBeginEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
}
}

\AfterEndEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{gather}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env gather*

```
\BeforeBeginEnvironment{gather*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{gather*}]
}
}

\AfterEndEnvironment{gather*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{gather*}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env align

```
\BeforeBeginEnvironment{align}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
}
}

\AfterEndEnvironment{align}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{align}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env align*

```
\BeforeBeginEnvironment{align*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{align*}]
}
}

\AfterEndEnvironment{align*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{align*}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env flalign

```
\BeforeBeginEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
}
}

```

```
\AfterEndEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{flalign}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env flalign*

```
\BeforeBeginEnvironment{flalign*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{flalign*}]
}
}

\AfterEndEnvironment{flalign*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{flalign*}{\the\@envbody}
}
{\end{lateximage}}

}
```

```
\end{warpHTML}
```

# 68 Lateximages

## 68.1 Description

Env lateximage A `lateximage` is a piece of the document which is typeset in LaTeX then included in the HTML output as an image. This is used for math if SVG math is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 66.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `<span>` is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

SVG image font size The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

## 68.2 Support counters and macros

**for HTML output:**

```
\begin{warpHTML}
```

Ctr `LWR@lateximagenumber` Sequence the images.

```
\newcounter{LWR@lateximagenumber}
\setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
\newcounter{LWR@lateximagedepth}
\setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
\edef\LWR@hashmark{\string#} % for use in \write
\edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@LIpage` Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.

```
\newcounter{LWR@LIpage}
```

```
\end{warpHTML}
```

## 68.3 Font size

for HTML & PRINT:

```
\begin{warpall}
```

\LateximageFontSizeName Declares how large to write text in the `\lateximage`. The `.svg` file text size should blend well with the surrounding HTML text size.

⚠ no backslash *Do not include the leading backslash in the name.*

```
\newcommand*{\LateximageFontSizeName}{large}
```

```
\end{warpall}
```

## 68.4 Sanitizing math expressions for HTML

for HTML output:

```
\begin{warpHTML}
```

\LWR@HTMLsanitize {⟨*text*⟩}

Math expressions are converted to `lateximages`, and some math environments may contain "&", "<", or ">", which should not be allowed inside an HTML `<alt>` tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way. . .

```
\newcommand{\LWR@HTMLsanitize}[1]{%
\begingroup%
\LWR@FBcancel%
\protect\StrSubstitute{\detokenize{#1}}%
{\detokenize{&}}%
{\detokenize{&amp;}}[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}%
{\detokenize{&lt;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}%
{\detokenize{&gt;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{##}}%
{\#}%
[\LWR@strresult]%
\LWR@strresult%
\endgroup%
}
```

\LWR@HTMLsanitizeexpand {⟨*text*⟩}

This version expands the argument before sanitizing it.

```
\newcommand{\LWR@HTMLsanitizeexpand}[1]{%
\begingroup%
\LWR@FBcancel%
\protect\StrSubstitute{\detokenize\expandafter{#1}}%
{\detokenize{&}}%
{\detokenize{&amp;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}%
{\detokenize{&lt;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}%
{\detokenize{&gt;}}%
[\LWR@strresult]%
\LWR@strresult%
\endgroup%
}
```

## 68.5 Equation numbers

Ctr LWR@startingequation For use with `lateximage` and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```
\newcounter{LWR@startingequation}

\@ifundefined{chapter}
{
\renewcommand{\theLWR@startingequation}{%
\arabic{LWR@startingequation}%
}
}
{% chapter defined
\renewcommand{\theLWR@startingequation}{%
\ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}%
\arabic{LWR@startingequation}%
}
}
```

Bool LWR@isstartingequation

True for the first equation tag, false for later tags in the same environment.

```
\newbool{LWR@isstartingequation}
```

\LWR@startingequationtag

Prints the starting equation number or tag.

```
\let\LWR@startingequationtag\theLWR@startingequation
```

\LWR@equationtag

Prints the ending equation number or tag.

```
\let\LWR@equationtag\theequation
```

Only if SVG math, patch \tag after packages have loaded, in case someone else modified \tag.

```
\AtBeginDocument{

\ifbool{mathjax}{}{% not mathjax
```

\LWR@remembertag {⟨*tag*⟩}

For use inside the math environments while using SVG math. Sets \theLWR@startingequation and \theequation to the given tag.

```
\NewDocumentCommand{\LWR@remembertag}{m}{%
\ifbool{LWR@isstartingequation}%
{%
    \global\boolfalse{LWR@isstartingequation}%
    \xdef\LWR@startingequationtag{#1}%
}%
{}%
\xdef\LWR@equationtag{#1}%
}%
```

Patches for $\mathcal{AMS}$ math \tag macro to remember the first tag:

```
\LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
\LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@

\renewcommand*{\make@df@tag@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@{#1}%
}

\renewcommand*{\make@df@tag@@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@@{#1}%
}

}% not mathjax
}% AtBeginDocument
```

## 68.6 HTML <alt> tags

\LWR@amsmathbody {⟨*envname*⟩} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag.

```
\newcommand*{\LWR@amsmathbody}[1]
{%
\textbackslash\{begin\}\{#1\} %
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
\textbackslash\{end\}\{#1\}%
}
```

\LWR@amsmathbodynumbered {⟨*envname*⟩} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag, prefixed by the equation numbers.

```
\newcommand*{\LWR@amsmathbodynumbered}[1]
{%
\ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
{(\LWR@equationtag)}%
{(\LWR@startingequationtag--\LWR@equationtag)} %
\LWR@amsmathbody{#1} %
}
```

## 68.7 lateximage

Env lateximage [⟨*<alt> tag*⟩]

```
\catcode`\$=\active%


\NewDocumentEnvironment{lateximage}{O{image}}
{%
\LWR@traceinfo{lateximage: starting on page \arabic{page}}%
```

Nested `lateximage`s remain one large `lateximage`:

```
\ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
{%
    \addtocounter{LWR@lateximagedepth}{1}%
}%
```

Otherwise, this is the outer-most lateximage:

```
{% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
    \setcounter{LWR@startingequation}{\value{equation}}%
    \addtocounter{LWR@startingequation}{1}%
    \booltrue{LWR@isstartingequation}%
    \let\LWR@startingequationtag\theLWR@startingequation%
    \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
    \addtocounter{LWR@lateximagenumber}{1}%
    \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
    \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Next file:

```
    \addtocounter{LWR@externalfilecnt}{1}%
    \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
    \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
    \LWR@traceinfo{lateximage: LWR@LIpage is \arabic{LWR@LIpage}}%
```

Create an HTML span which will hold the comment which contains the pdftotext translation of the image's page, and also will hold the link to the .svg file:

```
\LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
class="lateximagesource"{}} \LWR@orignewline
```

Write instructions to the lateximages.txt file:

```
\LWR@traceinfo{lateximage: about to write to lateximages.txt}%
\immediate\write\LWR@lateximagesfile{|\theLWR@LIpage|\theLWR@externalfilecnt|}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```
\LWR@traceinfo{lateximage: about to create open comment}%
\LWR@htmlopencomment%
```

One level deeper:

```
\addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
\LWR@traceinfo{lateximage: about to create a new page}%
\LWR@orignewpage%
```

Typeset the image in a "standard" width page and font size:

```
\LWR@traceinfo{lateximage: about to create minipage}%
\LWR@origminipage{6in}%
\csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
\LWR@traceinfo{lateximage: about to temporarily restore formatting}%
\LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
\def\@mpfn{footnote}%
\def\thempfn{\thefootnote}%
\let\@footnotetext\LWR@footnotetext%
```

Create the LWRlateximage<number> label:

```
    \LWR@traceinfo{lateximage: about to create label}%
    \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
    \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
    \LetLtxMacro$\LWR@origdollar%
    \catcode`\$=3% math shift
    \LetLtxMacro\(\LWR@origopenparen%
    \LetLtxMacro\)\LWR@origcloseparen%
    \LetLtxMacro\[\LWR@origopenbracket%
    \LetLtxMacro\]\LWR@origclosebracket%
}% end of outer-most lateximage
\LWR@traceinfo{lateximage: finished start of environment}%
}% end of \begin{lateximage}
```

\endlateximage When the environment closes:

```
{% start of \end{lateximage}
\LWR@traceinfo{lateximage: starting end of environment}%
```

Nested more than one deep?

```
\ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
{\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
{% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
    \LWR@origendminipage%
    \LWR@orignewpage%
    \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
    \LWR@htmlclosecomment{}\LWR@orignewline%
    \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

```
    \LWR@subinlineimage[#1]{lateximage}%
    {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
    \LWR@htmltag{/span}%
    \ifbool{HTMLDebugComments}{%
        \LWR@htmlcomment{End of lateximage}%
    }{}%
% \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
    \addtocounter{LWR@lateximagedepth}{-1}%
}% end of outer-most lateximage
\LWR@traceinfo{lateximage: done}%
}%
\catcode`\$=3% math shift
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
% \newenvironment{lateximage}[1][]{\minipage{\linewidth}}{\endminipage}
\newenvironment{lateximage}[1][]{}{}
\end{warpprint}
```

# 69 center, flushleft, flushright

**for HTML output:**

```
\begin{warpHTML}
```

Env `center`

Replace `center` functionality with CSS tags:

```
\renewenvironment*{center}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[text-align:center]{center}}
{\BlockClass{center}}
}
{\endBlockClass}
```

Env **flushright**

```
\renewenvironment*{flushright}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[text-align:right]{flushright}}
{\BlockClass{flushright}}
}
{\endBlockClass}
```

Env **flushleft**

```
\renewenvironment*{flushleft}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[text-align:left]{flushleft}}
{\BlockClass{flushleft}}
}
{\endBlockClass}
```

```
\end{warpHTML}
```

# 70 Pre-loaded packages

**for HTML output:**

```
\begin{warpHTML}
```

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
\@ifpackageloaded{textcomp}
{
\LWR@origRequirePackage{lwarp-textcomp}
}
{}
```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```
\@ifpackageloaded{graphics}
{
\LWR@origRequirePackage{lwarp-graphics}
}
{}
```

```
\end{warpHTML}
```

# 71 Siunitx

Pkg siunitx

The lwarp core passes a few options to siunitx.

⚠ per-mode

Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ math mode required

Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the siunitx extension for MathJax is not currently hosted at any public CDN, thus siunitx is not usable with MathJax unless a local copy of this extension is created first.

**for HTML output:**

```
\begin{warpHTML}
```

Options for siunitx:

```
\PassOptionsToPackage{
    detect-mode=true,
    per-mode=symbol,% fraction is not seen by pdftotext
%     text-celsius = {\protect\LWRsiunitx@degree{}C},
%     text-degree = {\protect\LWRsiunitx@degree},
}{siunitx}

```

```
\end{warpHTML}
```

# 72 Graphics print-mode modifications

## 72.1 General limitations

⚠ .pdf image files

⚠ no file extension

For `\includegraphics` with `.pdf` files, the user should provide a `.pdf` image file, and also a `.svg`, `.png`, or `.jpg` version of the same image. These should be referred to without a file extension:

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

For print output, lwarp will automatically choose the `.pdf` if available, other some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

other image files — For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

⚠ graphics vs. graphicx — If using the older graphics syntax, use both optional arguments for \includegraphics. A single optional parameter is interpreted as the newer graphicx syntax. Note that

⚠ viewports — viewports are not supported by warp; the entire image will be shown.

units — For \includegraphics, avoid px and % units for width and height, or enclose them inside warpHTML environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys width=.5\linewidth, or similar for \textwidth or \textheight to give fixed-sized images proportional to a 6 by 9 inch text area.

options — \includegraphics accepts width and height, origin, rotate and scale, plus a new class key.

HTML class — With HTML output, \includegraphics accepts an optional class=xyz keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

\rotatebox — \rotatebox accepts the optional origin key.

⚠ browser support — \rotatebox, \scalebox, and \reflectbox depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

## 72.2 Print-mode modifications

**for PRINT output:** For print output, accept and then discard the new class key:

```
\begin{warpprint}
\define@key{Gin}{class}{}
```

Print-mode additions for the overpic package. See section 202 for the HTML version.

```
\AtBeginDocument{
\@ifpackageloaded{overpic}{
\newcommand*{\overpicfontsize}{12}
\newcommand*{\overpicfontskip}{14}
}{}
}
\end{warpprint}
```

## 73 Xcolor boxes

Pkg xcolor

A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of `none`, in which case only the frame is drawn, allowing the background page color to show.

**for HTML & PRINT:**

```
\begin{warpall}
```

After xparse may have been loaded …

```
\AtBeginDocument{
```

… and *only* if xcolor was loaded:

```
\@ifpackageloaded{xcolor}{
\LWR@traceinfo{patching xcolor}
```

`\colorboxBlock`

`\colorboxBlock` is the same as `\colorbox`:

```
\LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
\warpprintonly{
\LetLtxMacro\LWRprint@colorboxBlock\colorbox
\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
}
```

`\fcolorbox` [⟨*framemodel*⟩] {⟨*framecolor*⟩} [⟨*boxmodel*⟩] {⟨*boxcolor*⟩} {⟨*text*⟩}

In print mode, `\fcolorbox` is modified to accept a background color of `none`.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

```
\newsavebox{\LWR@colorminipagebox}

\DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
\LWR@traceinfo{LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
#5%
\end{lrbox}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    \LWR@traceinfo{background is none}%
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
\LWR@traceinfo{background not none}%
\IfValueTF{#1}%
{%
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}%
{% no value #1
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}% no value #1
}% #4 not none
\LWR@traceinfo{LWRprint@fcolorbox done}%
}
```

`\fcolorboxBlock` [⟨*framemodel*⟩] {⟨*framecolor*⟩} [⟨*boxmodel*⟩] {⟨*boxcolor*⟩} {⟨*text*⟩}

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
\LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox
```

Env `fcolorminipage` [⟨*1:framemodel*⟩] {⟨*2:framecolor*⟩} [⟨*3:boxmodel*⟩] {⟨*4:boxcolor*⟩} [⟨*5:align*⟩] [⟨*6:height*⟩] [⟨*7:inner-align*⟩] {⟨*8:width*⟩}

In print mode, becomes a `\fcolorbox` containing a `minipage`:

```
\NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{} o m}
{%
\LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
\IfValueTF{#7}%
{\begin{minipage}[#5][#6][#7]{#8}}%
{\begin{minipage}[#5][#6][#5]{#8}}%
}%
{%
\end{minipage}%
\end{lrbox}%
\LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
    \IfValueTF{#1}%
    {%
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }%
    {% no value #1
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }% no value #1
}% #4 not none
\LWR@traceinfo{*** finished end fcolorminipage}%
}
```

\LWR@restoreorigprintxcolor Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `lateximage`.

```
\newcommand*{\LWR@restoreorigprintxcolor}{%
\LWR@traceinfo{LWR@restoreorigprintxcolor}%
\LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
\LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
\LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
\LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
\LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
}

\appto{\LWR@restoreorigformatting}{%
\LWR@restoreorigprintxcolor%
}
```

If print mode, immediately activate the print-mode enhancements for xcolor:

```
\warpprintonly{\LWR@restoreorigprintxcolor}

\LWR@traceinfo{xcolor patches done}
}{}% xcolor loaded
}% AtBeginDocument
```

```
\end{warpall}
```

# 74 Cleveref

Pkg cleveref cleveref package is used as-is with minor patches.

⚠ cleveref page numbers cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to “for”.

Ex:

```
    \cpageref{tab:first,tab:second}
 in HTML becomes:
```

“pages **for** table 4.1 and **for** table 4.2”

See `\cpagerefFor` at page 402 to redefine the message which is printed for page number references.

loading order

cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires cleveref. The user’s document may not require cleveref, thus the user may never explicitly load it, so during HTML output lwarp loads it last. If the user’s document preamble uses cleveref options, or functions such as `\crefname`, then cleveref may be loaded in the user’s preamble near the end, and lwarp’s additional loading of cleveref will have no effect.

Table 10 on page 340 shows the data structure of the label/reference system as revised by lwarp and cleveref.

A few patches allow cleveref to work as-is:

**for HTML output:**

```
\begin{warpHTML}
```

`\AtEndPreable` forces cleveref to be loaded last:

```
\AtEndPreamble{
\RequirePackage{cleveref}
}
```

The following patches are applied after cleveref has loaded, and after `\AtBeginDocument`:

```
\AfterEndPreamble{
\LWR@traceinfo{Patching cleveref.}
```

`\@@setcref` {⟨*kindofref*⟩} {⟨*label*⟩}

```
\renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{}{}}
```

`\@@setcrefrange` {⟨*text*⟩} {⟨*label*⟩} {⟨*label*⟩}

```
\renewcommand{\@@setcrefrange}[3]{%
#1{\ref{#2}}{\ref{#3}}{}{}{}{}}
```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

```
\newcommand*{\cpagerefFor}{for}
```

\@@setcpageref {⟨*typeofref*⟩} {⟨*label*⟩}, where typeofref is “page” or “pages”

```
\renewcommand*{\@@setcpageref}[2]{%
#1{\cpagerefFor\ \cref{#2}}{}{}%
}
```

```
\renewcommand{\@@setcpagerefrange}[3]{%
#1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}}
}% AfterEndPreamble
```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

```
\LetLtxMacro\LWR@origlabel\label
% \LetLtxMacro\label\LWR@newlabel
\RenewDocumentCommand{\label}{}{\LWR@newlabel}

\LetLtxMacro\LWR@origref\ref
% \LetLtxMacro\ref\LWR@newref
\RenewDocumentCommand{\ref}{}{\LWR@newref}%

\LetLtxMacro\LWR@origpageref\pageref
% \LetLtxMacro\pageref\LWR@newpageref
\RenewDocumentCommand{\pageref}{}{\LWR@newpageref}

\end{warpHTML}
```

# 75 Picture

Env picture The picture environment is enclosed inside a \lateximage.

**for HTML output:**

```
\begin{warpHTML}
```

Env picture

```
\BeforeBeginEnvironment{picture}{\begin{lateximage}}

\AfterEndEnvironment{picture}{\end{lateximage}}
```

```
\end{warpHTML}
```

# 76 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

⚠ inline A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.

placement Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.

side-by-side Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are surpressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an "inline" format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

size When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages A minipage of width exactly `\linewidth` is automatically given no HTML width.

full-width minipages A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML `width` attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment Nested minipages adopt their parent's text alignment in HTML, whereas in regular LaTeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

**for HTML output:**

```
7504 \begin{warpHTML}
```

## 76.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
7505 \newcounter{LWR@minipagedepth}
7506 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
\newlength{\LWR@minipagewidth}
```

Len \WR@minipageheight Used to convert the height into printable units.

```
\newlength{\LWR@minipageheight}
```

## 76.2 Footnote handling

Also see section 48 for other forms of footnotes. Minipage footnotes are gathered in section 48.5, and then placed into the document in section 76.3.

## 76.3 Minipage handling

\LWR@endminipage Used to close a minipage.

Copied the LaTeX definition and modified to create a `<div>` of class `mpfootnotes`:

```
\def\LWR@endminipage{%
    \par
    \unskip
    \ifvoid\@mpfootins\else
      \vskip\skip\@mpfootins
      \normalcolor
        \LWR@htmldivclass{mpfootnotes}
        \LWR@origmedskip
      \unvbox\@mpfootins
        \LWR@htmldivclassend{mpfootnotes}
    \fi
    \@minipagefalse
  \color@endgroup
  \egroup
  \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.

```
\newcommand*{\LWR@subminipage}{%
\LWR@stoppars
\LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
\LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
\LWR@startpars%
}
```

\LWR@endsubminipage Closes the subminipage.

```
\newcommand*{\LWR@endsubminipage}{%
\LWR@stoppars%
\LWR@endminipage% The following empty line is required:

}
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML `width`?

```
\newbool{LWR@minipagefullwidth}
\boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no `width` tag in HTML:

**for HTML output:**

```
\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
\end{warpHTML}
```

**for PRINT output:**

```
\begin{warpprint}
\newcommand*{\minipagefullwidth}{}
\end{warpprint}
```

**for HTML output:**

```
\begin{warpHTML}
```

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
\newbool{LWR@minipagethispar}
\boolfalse{LWR@minipagethispar}
```

Env minipage [⟨*vert position*⟩] [⟨*height*⟩] [⟨*inner vert position*⟩] {⟨*width*⟩}

The vertical positions may be 'c', 't', or 'b'. The inner position may also be 's'.

When using `\linewidth`, `\textwidth`, or `\textheight`, these are scaled proportionally to a 6×9 inch text area.

```
\RenewDocumentEnvironment{minipage}{O{t} o O{t} m}
{%
```

Units for printing dimensions to HTML:

```
\uselengthunit{PT}%
```

Compute width, adjusted for frames:

```
\setlength{\LWR@minipagewidth}{#4}%
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
    \addtolength{\LWR@minipagewidth}{3em}% room for frames
    \setlength{\linewidth}{6in}%
    \setlength{\textwidth}{6in}%
    \setlength{\textheight}{9in}%
}{}%
\LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
```

Compute height:

```
\setlength{\LWR@minipageheight}{\textheight}% default unless specified
\IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
\addtocounter{LWR@minipagedepth}{1}%
```

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.

```
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars%
```

If FormatWP, add a text frame:

```
\ifbool{FormatWP}{%

\addtocounter{LWR@thisautoid}{1}%
\booltrue{LWR@freezethisautoid}%
\LWR@htmltag{div id="autoid-\arabic{LWR@thisautoid}" class="wpminipage"}%

}{}%
```

Create the <div> tag with optional alignment style:

```
\LWR@traceinfo{minipage: creating div class}%
\LWR@orignewpage%
\LWR@htmltag{div class="minipage" style="%
\ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
\ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
```

```
\ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
\ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
\ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
\ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
\ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```
\LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
\uselengthunit{PT}%
\ifbool{LWR@minipagefullwidth}%
{\boolfalse{LWR@minipagefullwidth}}%
{%
    \ifthenelse{\lengthtest{#4}=\linewidth}%
    {}%
    {width:\rndprintlength{\LWR@minipagewidth} ; }%
}%
\LWR@traceinfo{minipage: about to print the height}%
\IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
"{}}%
```

Finish with an empty line to start LaTeX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

\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.

```
\setlength{\linewidth}{#4}% the original width
\setlength{\textwidth}{6in}%
\setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
\LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== begin minipage ===

}{}%
```

```
\LWR@traceinfo{minipage: finished starting the minipage}%
}
```

End the environment with LaTeX processing and closing tag:

```
{%
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== end minipage ===

}{}%
\LWR@stoppars%
\LWR@endminipage% The following empty line is required:

\ifbool{FormatWP}{%

\LWR@htmlelementend{div}%
\boolfalse{LWR@freezethisautoid}%

}{}%
\LWR@htmldivclassend{minipage}%

\LWR@origvspace{1\baselineskip}% required for subcaption
\addtocounter{LWR@minipagedepth}{-1}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
\global\booltrue{LWR@minipagethispar}%
}
```

## 76.4 Parbox, makebox, framebox, fbox, raisebox

**for HTML output:** \parbox [⟨*pos*⟩] [⟨*height*⟩] [⟨*inner-pos*⟩] {⟨*width*⟩} {⟨*text*⟩}

A parbox uses the minipage code:

```
\RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
{
\LWR@traceinfo{parbox of width #4}%
\begin{minipage}[#1][#2][#3]{#4}
#5
\end{minipage}
}
```

\makebox (⟨()⟩posn) [⟨*width*⟩] [⟨*pos*⟩] {⟨*text*⟩}

```
\LetLtxMacro\LWR@origmakebox\makebox

\RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
\IfValueTF{#2}%
{%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is `justified` alignment.

```
    {% scope
    \def\LWR@align{center}%
    \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
    \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
    \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the `width` argument:

```
    \setlength{\LWR@tempwidth}{#2}%
```

`inline-block` allows `width` and `text-alignment` to be used in a `<span>`.

```
    \uselengthunit{PT}%
    \InlineClass[%
        display:inline-block ; %
        text-align:\LWR@align\ ; %
        width:\rndprintlength{\LWR@tempwidth}%
    ]%
    {makebox}%
```

Without a width argument, the text is simply used inline:

```
    {\mbox{#4}}%
    }% scope
}%
{\mbox{#4}}%
}
```

\framebox [⟨*width*⟩] [⟨*pos*⟩] {⟨*text*⟩}

```
\LetLtxMacro\LWR@origframebox\framebox

\RenewDocumentCommand{\framebox}{o o m}{%
\fbox{\makebox[#1][#2]{#3}}%
}
```

\LWR@forceminwidth {⟨*legth*⟩}

Sets \LWR@atleastonept to be at least 1pt.

```
\newlength{\LWR@atleastonept}

\newcommand*{\LWR@forceminwidth}[1]{%
\setlength{\LWR@atleastonept}{#1}%
\ifthenelse{%
    \lengthtest{\LWR@atleastonept>0pt}\AND%
    \lengthtest{\LWR@atleastonept<1pt}%
}%
{\setlength{\LWR@atleastonept}{1pt}}%
{}%
}
```

\LWR@blackborderpadding Prints the HTML attributes for a black border and padding.

\LWR@forceminwidth must be used first in order to set the border width.

```
\newcommand*{\LWR@blackborderpadding}{%
\uselengthunit{PT}%
border:\rndprintlength{\LWR@atleastonept} solid black ; %
padding:\rndprintlength{\fboxsep}%
}
```

\fbox {⟨*text*⟩}

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
\let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
\newcommand{\LWRhtml@fbox}[1]{%
\LWR@traceinfo{HTML fbox}
\LWR@forceminwidth{\fboxrule}%
\InlineClass[%
\LWR@blackborderpadding%
]{fbox}{#1}
}
```

xcolor `\let`s things to `\fbox` when it is loaded, and this must remain even for HTML output while in a `lateximage`, so `\fbox` is not modified until `\AtBeginDocument`:

```
\AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` {⟨*text*⟩} Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```
\newcommand{\fboxBlock}[1]{%
\LWR@forceminwidth{\fboxrule}%
\begin{BlockClass}[%
\LWR@blackborderpadding%
]{fboxBlock}
#1
\end{BlockClass}
}
```

Env `fminipage` [⟨*align*⟩] [⟨*height*⟩] [⟨*align*⟩] {⟨*width*⟩}

Creates a framed HTML `<div>` around its contents.

A print-output version is also supplied below.

```
\NewDocumentEnvironment{fminipage}{O{t} o O{t} m}
{%
\LWR@traceinfo{fminipage #1 #2 #3 #4}%
\LWR@forceminwidth{\fboxrule}%
\setlength{\LWR@tempwidth}{#4}%
\IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
\begin{BlockClass}[%
\LWR@blackborderpadding ; %
\uselengthunit{PT}%
\IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
width:\rndprintlength{\LWR@tempwidth}%
]{fminipage}%
}
{%
\end{BlockClass}%
\LWR@traceinfo{fminipage done}%
}
```

`\raisebox` {⟨*raiselen*⟩} [⟨*height*⟩] [⟨*depth*⟩] {⟨*text*⟩}

```
\LetLtxMacro{\LWR@origraisebox}{\raisebox}

\RenewDocumentCommand{\raisebox}{m o o m}{%
#4%
}
```

```
\end{warpHTML}
```

**for HTML & PRINT:**

```
\begin{warpall}
```

LWRprint@fminipage is defined inside warpall. For print output, it is \let to fminipage. For HTML output, the HTML version of fminipage is used instead, but the print version is still available for use inside a lateximage.

Env LWRprint@fminipage [⟨*1:align*⟩] [⟨*2:height*⟩] [⟨*3:inner-align*⟩] {⟨*4:width*⟩}

Creates a frame around its contents.

```
\newsavebox{\LWR@fminipagebox}

\NewDocumentEnvironment{LWRprint@fminipage}{O{t} o O{t} m}
{%
```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with \fbox.

If the optional inner alignment is not given, use the outer instead:

```
\IfValueTF{#3}%
{\def\LWR@thisalign{#3}}
{\def\LWR@thisalign{#1}}%
```

Form the outer minipage depending on whether a height was given. Make the outer minipage larger to compensate for the frame.

```
\IfValueTF{#2}%
{\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
{\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
\begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
\IfValueTF{#2}%
{\minipage[#1][#2][\LWR@thisalign]{#4}}%
{\minipage[#1]{#4}}%
}
{%
```

Close the inner minipage and the LR box with the contents:

```
\endminipage%
\end{lrbox}%
```

Create a frame around the contents of the environment:

```
\fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
\endminipage%
}
```

```
\end{warpall}
```

**for PRINT output:**

```
\begin{warpprint}
```

For print output, the following are `\let` to become active.

`\fboxBlock` `{`⟨*text*⟩`}`

Creates a framed HTML `<div>` around the text.

```
\let\fboxBlock\fbox
```

Env `fminipage` `[`⟨*align*⟩`] [`⟨*height*⟩`] [`⟨*align*⟩`] {`⟨*width*⟩`}`

Creates a frame around its contents.

```
\LetLtxMacro{\fminipage}{\LWRprint@fminipage}
\LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
\end{warpprint}
```

## 77 Direct formatting

⚠ `\bfseries`, etc. `\textbf`, etc. are supported, but `\bfseries`, etc. are not yet supported.

⚠ HTML special chars `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

For high-level block and inline custom CSS classes, see section 42.8.

**for HTML output:**

```
\begin{warpHTML}
```

\LWR@HTMLtextstyle {⟨*FormatWP style*⟩} {⟨*class*⟩} {⟨*text*⟩}

If FormatWP, adds an explicit style to the text span class. This is used by LibreOffice to mark its imported text using the given style.

```
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
\ifbool{FormatWP}%
{\LWR@htmlspanclass[#1]{#2}{#3}}%
{\LWR@htmlspanclass{#2}{#3}}%
}
```

\emph {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
\DeclareRobustCommand{\LWR@nullemph}[1]{#1}
\LetLtxMacro{\emph}{\LWR@HTMLemph}
```

\textmd {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
\LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
}
\DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}

\LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

\textbf {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
\DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
\LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

\textrm {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
\LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}

\LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

\textsf {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
\LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
}
\DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
\LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

\texttt {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
\DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
\LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

\textup {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
\LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextup}[1]{#1}

\LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

\textit {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
\DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
\LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

\textsc {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
\LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}

\LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

\textsl {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
\LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
}

```

```
\DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}

\LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

\textnormal {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
\DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
\LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
\DeclareRobustCommand{\LWR@nullrmfamily}{}
\DeclareRobustCommand{\LWR@nullsffamily}{}
\DeclareRobustCommand{\LWR@nullttfamily}{}
\DeclareRobustCommand{\LWR@nullbfseries}{}
\DeclareRobustCommand{\LWR@nullmdseries}{}
\DeclareRobustCommand{\LWR@nullupshape}{}
\DeclareRobustCommand{\LWR@nullslshape}{}
\DeclareRobustCommand{\LWR@nullscshape}{}
\DeclareRobustCommand{\LWR@nullitshape}{}
\DeclareRobustCommand{\LWR@nullem}[1]{}
\DeclareRobustCommand{\LWR@nullnormalfont}{}
```

\LWR@nullfonts Removes formatting during filename operations.

```
\newcommand*{\LWR@nullfonts}{%
\LetLtxMacro{\emph}{\LWR@nullemph}%
\LetLtxMacro{\textmd}{\LWR@nulltextmd}%
\LetLtxMacro{\textbf}{\LWR@nulltextbf}%
\LetLtxMacro{\textrm}{\LWR@nulltextrm}%
\LetLtxMacro{\textsf}{\LWR@nulltextsf}%
\LetLtxMacro{\texttt}{\LWR@nulltexttt}%
\LetLtxMacro{\textup}{\LWR@nulltextup}%
\LetLtxMacro{\textit}{\LWR@nulltextit}%
\LetLtxMacro{\textsc}{\LWR@nulltextsc}%
\LetLtxMacro{\textsl}{\LWR@nulltextsl}%
\LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
\LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
\LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
\LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
\LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
\LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
\LetLtxMacro{\upshape}{\LWR@nullupshape}%
\LetLtxMacro{\slshape}{\LWR@nullslshape}%
\LetLtxMacro{\scshape}{\LWR@nullscshape}%
\LetLtxMacro{\itshape}{\LWR@nullitshape}%
\LetLtxMacro{\em}{\LWR@nullem}%
\LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
\renewcommand*{\HTMLunicode}[1]{}%
```

```
\renewcommand*{\HTMLentity}[1]{}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
\renewcommand*{\&}{and}%
\renewcommand{\textsuperscript}[1]{##1}%
\renewcommand{\textsubscript}[1]{##1}%
\LetLtxMacro\underline\LWR@origunderline%
\RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
\DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
\DeclareRobustCommand{\LWR@subsingledollar}[1]{}%
}
```

\mdseries

```
\renewcommand*{\mdseries}{}
```

\bfseries

```
\renewcommand*{\bfseries}{}
```

\rmfamily

```
\renewcommand*{\rmfamily}{}
```

\sffamily

```
\renewcommand*{\sffamily}{}
```

\ttfamily

```
\renewcommand*{\ttfamily}{}
```

\upshape

```
\renewcommand*{\upshape}{}
```

\itshape

```
\renewcommand*{\itshape}{}
```

\scshape

```
\renewcommand*{\scshape}{}
```

\normalfont

```
\renewcommand*{\normalfont}{}
```

\sp {⟨*text*⟩}

For siunitx. Must work in math mode.

```
\renewcommand{\sp}[1]{\text{<sup>#1</sup>}{}}
```

\sb {⟨*text*⟩}

For siunitx. Must work in math mode.

```
\renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}
```

\textsuperscript {⟨*text*⟩}

```
\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\@textsuperscript {⟨*text*⟩}

```
\renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\textsubscript {⟨*text*⟩}

```
\AtBeginDocument{
\renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
}
```

\@textsubscript {⟨*text*⟩}

```
\AtBeginDocument{
\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.

```
\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.

```
\AtBeginDocument{\let\fup\textsuperscript}
```

`\underline` {⟨*text*⟩}

```
\renewcommand{\underline}[1]{%
\LWR@HTMLtextstyle%
    {text-decoration:underline;text-decoration-skip}%
    {underline}{#1}%
}
```

`\hfill`

```
\renewcommand*{\hfill}{\qquad}
```

`\hrulefill`

```
\renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
\renewcommand*{\dotfill}{\dots}
```

```
\end{warpHTML}
```

## 78 Skips, spaces, font sizes

**for HTML output:**

```
\begin{warpHTML}
```

`\,` must be redefined after `\RequirePackage{printlen}`

```
\let\LWR@origcomma\,
\let\LWR@origtilde~
\let\LWR@origenskip\enskip
\let\LWR@origquad\quad
\let\LWR@origqquad\qquad
\let\LWR@orighspace\hspace
```

```
\let\LWR@origvspace\vspace
\let\LWR@origrule\rule
\let\LWR@origmedskip\medskip
\let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
\renewcommand*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space

\renewcommand*{~}{\HTMLentity{nbsp}}

\renewcommand*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
\let\LWR@orignormalsize\normalsize
\let\LWR@origsmall\small
\let\LWR@origfootnotesize\footnotesize
\let\LWR@origscriptsize\scriptsize
\let\LWR@origtiny\tiny
\let\LWR@origlarge\large
\let\LWR@origLarge\Large
\let\LWR@origLARGE\LARGE
\let\LWR@orighuge\huge
\let\LWR@origHuge\Huge
\renewcommand*{\normalsize}{}
\renewcommand*{\small}{}
\renewcommand*{\footnotesize}{}
\renewcommand*{\scriptsize}{}
\renewcommand*{\tiny}{}
\renewcommand*{\large}{}
\renewcommand*{\Large}{}
\renewcommand*{\LARGE}{}
\renewcommand*{\huge}{}
\renewcommand*{\Huge}{}

\renewcommand*{\onecolumn}{}

\renewcommand{\twocolumn}[1][]{

#1

}
```

\newline Uses the HTML <br /> element.

```
\newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
\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.

```
\LetLtxMacro\LWR@origendofline\\
\NewDocumentCommand{\LWR@endofline}{s o}
{%
\newline%
}
```

\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 surpresses 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 surpressed around \hspace, \enskip, \quad, and \qquad until the end of the paragraph, when the closing p tag is created.

\hspace
\enskip
\quad
\qquad

When a minipage is seen, the boolean LWR@minipagethispar is set, telling the following horizontal whitespace commands to try to surpress 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.

```
\newcommand*{\LWR@minipagestartpars}{%
\ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
}
```

\LWR@minipagestoppars Placed just after \hspace, \quad, or \qquad's HTML output.

```
\newcommand*{\LWR@minipagestoppars}{%
\ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
}
```

\quad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\quad}{%
\LWR@minipagestoppars%
\HTMLunicode{2001}%
\LWR@minipagestartpars%
}
```

\qquad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
\renewcommand*{\enskip}{%
\LWR@minipagestoppars%
\HTMLunicode{2000}%
\LWR@minipagestartpars%
}
```

Len \WR@tempwidth
Len \WR@tempheight
Len \WR@tempraise

Used to compute span width, height, raise for `\hspace` and `\rule`:

```
\newlength{\LWR@tempwidth}
\newlength{\LWR@tempheight}
\newlength{\LWR@tempraise}
```

\LWR@hspace * {⟨*length*⟩}

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\qquad`.

```
\NewDocumentCommand{\LWR@hspace}{s m}{%
\setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qquad`:

```
\ifnum\gluestretchorder\LWR@tempwidth>0%
\setlength{\LWR@tempwidth}{2em}%
\fi%
```

Only if the width is not zero:

```
\ifdimcomp{\LWR@tempwidth}{=}{0pt}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
    \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
    {%
        \HTMLunicode{2009}% thin breakable space
    }%
```

Print the span with the converted width. Not rounded.

```
    {%
        \uselengthunit{PT}%
        \LWR@htmltagc{%
            span style="width:\printlength{\LWR@tempwidth}; %
                display:inline-block"%
        }%
```

If formatting for a word processor, approximate with a number of `\quad`s, in case a span of a given width is not supported:

```
        \ifbool{FormatWP}{%
            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
                \quad%
                \addtolength{\LWR@templengthone}{-1em}%
            }%
        }{}%
```

Close the span:

```
        \LWR@htmltagc{/span}%
    }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% width not 0
}
```

`\LWR@nohspace` * {⟨*length*⟩}

Used to disable `\hspace` while creating `description` `\item`s.

```
\NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` * {⟨*length*⟩}

Handles special minipage & horizontal space interactions.

```
\LetLtxMacro{\hspace}{\LWR@hspace}
```

\LWR@vspace * {⟨*length*⟩} Nullified vspace.

```
\NewDocumentCommand{\LWR@vspace}{s m}{}
```

\vspace * {⟨*length*⟩} Nullified.

```
\let\vspace\LWR@vspace
```

\linebreak [⟨*num*⟩] Inserts an HTML br tag.

```
\renewcommand*{\linebreak}[1][]{\newline}
```

\nolinebreak [⟨*num*⟩]

```
\renewcommand*{\nolinebreak}[1][]{}
```

\pagebreak [⟨*num*⟩] Starts a new paragraph.

```
\renewcommand*{\pagebreak}[1][]{

}
```

\nopagebreak [⟨*num*⟩]

```
\renewcommand*{\nopagebreak}[1][]{}
```

\enlargethispage * {⟨*len*⟩}

```
\RenewDocumentCommand{\enlargethispage}{s m}{}
```

\clearpage
\cleardoublepage

```
\renewcommand*{\clearpage}{}
\renewcommand*{\cleardoublepage}{}
```

\LWR@currenttextcolor The color to use for text and \rule, defaulting to black:

```
\newcommand*{\LWR@currenttextcolor}{black}
```

\LWR@rule [⟨*raise*⟩] {⟨*width*⟩} {⟨*height*⟩}

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
\NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary LaTeX length, from which comparisons and conversions may be made:

```
\setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
\ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
{}% zero- width
{% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
    \ifthenelse{%
        \lengthtest{\LWR@tempwidth>0pt}\AND%
        \lengthtest{\LWR@tempwidth<1pt}%
    }%
    {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
    \setlength{\LWR@tempheight}{#3}%
    \ifthenelse{%
        \lengthtest{\LWR@tempheight>0pt}\AND%
        \lengthtest{\LWR@tempheight<1pt}%
    }%
    {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in LaTeX code.

```
    \uselengthunit{PT}%
```

```
\LWR@htmltagc{%
span
style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
\ifbool{FormatWP}{}{background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
width:\printlength{\LWR@tempwidth} ; %
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 LaTeX length allows a typical LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of LaTeX ex height, then print that to HTML with an ex unit.

```
\IfValueT{#1}%
{%
    \setlength{\LWR@tempraise}{0pt-#1}%
    \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
    \LWR@orignewline%
    -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
    \LWR@orignewline%
    -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
    \LWR@orignewline%
    transform: translate(0pt,\printlength{\LWR@tempraise}); %
    \LWR@orignewline%
}%
```

Display inline-block to place the span inline with the text:

```
display:inline-block;"%
}%
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```
    \ifbool{FormatWP}{%
        \setlength{\LWR@templengthone}{\LWR@tempwidth}%
        \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
            \_{}%
            \addtolength{\LWR@templengthone}{-1em}%
        }%
        }{}%
```

Close the span:

```
    \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% non-zero width
}
```

\rule [⟨*raise*⟩] {⟨*width*⟩} {⟨*height*⟩}

Handles special minipage & horizontal space interactions.

```
\renewcommand{\rule}{\LWR@rule}
```

```
\end{warpHTML}
```

## 79 \phantomsection

**for HTML output:**

```
\begin{warpHTML}
```

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```
\newcommand*{\phantomsection}{%
\section*{}%
}
```

```
\end{warpHTML}
```

## 80 \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

## 80.1 HTML logos

for HTML output:

```
\begin{warpHTML}
```

\TeX TeX

latexlogo is a CSS class used to properly typeset the E and A in LaTeX and friends.

latexlogofont is a CSS class used to select the font for the rest of the logo in LaTeX, LuaTeX, ConTeXt, etc.

```
\let\LWR@origTeX\TeX

\newcommand*{\LWR@TeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
            {latexlogo}%
        {T\textsubscript{e}X}%
    }%
}
```

\LaTeX
\LaTeXe LaTeX, LaTeX 2$_\varepsilon$

```
\let\LWR@origLaTeX\LaTeX

\newcommand*{\LWR@LaTeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
            {latexlogo}%
        {L\textsuperscript{a}T\textsubscript{e}X}%
    }%
}

\let\LWR@origLaTeXe\LaTeXe

\renewcommand*{\LaTeXe}
```

```
{\LaTeX\InlineClass{latexlogofont}%
{\,2\textsubscript{\textit{\HTMLunicode{3B5}}}}}
```

\LuaTeX \LuaLaTeX LuaTeX, LuaLaTeX

```
\newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
\newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX \XeLaTeX XƎTeX, XƎLaTeX

xetexlogo is a CSS class which aligns the backwards E in XƎTeX and spaces TeX appropriately.

xelatexlogo is a CSS class which aligns the backwards E in XƎLaTeX and spaces LaTeX appropriately.

```
\newcommand*{\Xe}
  {X\textsubscript{\HTMLunicode{18e}}}
\newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
\newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConTeXt

```
\newcommand*{\LWR@ConTeXt}
{\InlineClass{latexlogofont}{Con}\TeX{}%
\InlineClass{latexlogofont}{t}}
```

\BibTeX \MakeIndex BibTeX, *MakeIndex*

```
\providecommand*{\BibTeX}
{\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}

\newcommand*{\MakeIndex}
{\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

\AmS *AMS*

amslogo is a CSS class used for the *AMS*logo.

```
\AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
{\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

\MiKTeX MiKTeX

```
\newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

\LyX LyX

lyxlogo is a CSS class used for the LyXlogo.

```
\newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
\end{warpHTML}
```

### 80.2 Print logos

for PRINT output:

```
\begin{warpprint}
\newcommand*{\XeTeXrevE}
  {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
\providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
\providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
\providecommand*{\AmS}{%
\leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
\hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
\newcommand*{\LyX}{\textsf{LyX}}
\providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
\providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
\providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
\providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
\providecommand*{\ConTeXt}{\mbox{Con\TeX{}t}}
\providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
\end{warpprint}
```

## 81 \AtBeginDocument, \AtEndDocument

for HTML output:

```
\begin{warpHTML}
```

\LWR@LwarpStart
\LWR@LwarpEnd
Automatically sets up the HTML-related actions for the start and end of the document.

```
\AfterEndPreamble{\LWR@LwarpStart}
\AtEndDocument{\LWR@LwarpEnd}
```

```
\end{warpHTML}
```

## 82 Trademarks

- TEX is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

The following adjustments apply to the lwarp-* package listings:

File 2 `lwarp-patch-komascript.sty`

## § 83 Package patch-komascript

Pkg `lwarp-patch-komascript`

Patches for komascript classes.

lwarp loads this package when scrbook, scrartcl, or scrreprt are detected.

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

⚠ Not yet tested! Please send bug reports!

Some features have not yet been tested. Please contact the author with any bug reports.

**for HTML output:**

```
\ProvidesPackage{lwarp-patch-komascript}
```

typearea is emulated.

```
\RequirePackage{lwarp-typearea}
```

tocbasic is emulated.

```
\RequirePackage{lwarp-tocbasic}
```

scrextend patches most of the new macros.

```
\RequirePackage{lwarp-scrextend}
```

The `\minisec` is placed inside a `<div>` of class `minisec`.

```
\renewcommand*{\minisec}[1]{
\begin{BlockClass}{minisec}
#1
\end{BlockClass}
}
```

The part and chapter preambles are placed as plain text just after each heading.

```
\@ifundefined{setpartpreamble}{}{
```

```
\RenewDocumentCommand{\setpartpreamble}{o o +m}{%
\renewcommand{\part@preamble}{#3}%
}
}

\@ifundefined{setchapterpreamble}{}{
\RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
\renewcommand{\chapter@preamble}{#3}%
}
}
```

Simple captions are used in all cases.

```
\LetLtxMacro\captionbelow\caption
\LetLtxMacro\captionabove\caption

\LetLtxMacro\captionofbelow\captionof
\LetLtxMacro\captionofabove\captionof

\RenewDocumentEnvironment{captionbeside}{o m o o o s}
{}
{%
\IfValueTF{#1}%
{\caption[#1]{#2}}%
{\caption{#2}}%
}

\RenewDocumentEnvironment{captionofbeside}{m o m o o o s}
{}
{%
\IfValueTF{#2}%
{\captionof{#1}[#2]{#3}}%
{\captionof{#1}{#3}}%
}

\RenewDocumentCommand{\setcapindent}{s m}{}
\renewcommand*{\setcaphanging}{}
\renewcommand*{\setcapwidth}[2][]{}
\renewcommand*{\setcapdynwidth}[2][]{}
\RenewDocumentCommand{\setcapmargin}{s o m}{}
```

# File 3 **lwarp-a4.sty**

## § 84 Package **a4**

Pkg a4 a4 is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{a4}
```

```
\newcommand*{\WideMargins}{}
```

---

## File 4 `lwarp-a4wide.sty`

# § 85 Package **a4wide**

Pkg `a4wide`

a4wide is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{a4wide}
```

---

## File 5 `lwarp-a5comb.sty`

# § 86 Package **a5comb**

Pkg `a5comb`

a5comb is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{a5comb}
```

---

## File 6 `lwarp-abstract.sty`

# § 87 Package **abstract**

(*Emulates or patches code by* PETER WILSON.)

Pkg `abstract`

abstract is supported and patched by lwarp.

⚠ missing TOC

If using the `number` option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a file split, which would put a table of contents out of the home page if it is after the abstract.

**for HTML output:**

Accept all options for lwarp-abstract:

```
\LWR@ProvidesPackagePass{abstract}
```

```
\AtBeginDocument{
\BeforeBeginEnvironment{abstract}{
\LWR@forcenewpage
\BlockClass{abstract}
}
\AfterEndEnvironment{abstract}{\endBlockClass}
}

\renewcommand{\@bsrunintitle}{%
\hspace*{\abstitleskip}%
{\abstractnamefont%
\InlineClass{abstractrunintitle}{\abstractname}%
\@bslabeldelim}%
}

\if@titlepage
  \renewenvironment{abstract}{%
%     \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
          \begin{\absnamepos}%
  \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
            \@endparpenalty\@M
          \end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
    }
\else
  \renewenvironment{abstract}{%
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
\begin{\absnamepos}%
\abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
\end\absnamepos%
```

```
%%   \vspace{\abstitleskip}%
    \fi
   \fi
   \vspace{\abstitleskip}%
  \fi
  \put@bsintoc%
  \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
  {\par\end{@bstr@ctlist}}
\fi

```

---

File 7 **lwarp-adjmulticol.sty**

# § 88 Package **adjmulticol**

(*Emulates or patches code by* Boris Veytsman.)

Pkg adjmulticol

adjmulticol is emulated.

Emulation similar to multicols is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, lwarp allows a variable number of columns up to three.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{adjmulticol}
```

```
\RequirePackage{multicol}
```

adjmulticols * {⟨*numcols*⟩} {⟨*left margi*⟩} {⟨*right margin*⟩}

```
\NewDocumentEnvironment{adjmulticols}{s m m m}
{%
```

Compute the margins, and limit to positive only:

```
\setlength{\LWR@templengthone}{#3}%
\ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
\setlength{\LWR@templengthtwo}{#4}
\ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%
```

If one column is specified, use a <div> of class singlecolumn, else use multicols:

```
\newcommand*{\LWR@mcolstype}{multicols}%
\ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{singlecolumn}}{}%
```

Help avoid page overflow:

```
\LWR@forcenewpage%
```

Print margins in points:

```
\uselengthunit{PT}%
```

Create the `<div>` with the given margin and class:

```
\BlockClass[%
margin-left:\rndprintlength{\LWR@templengthone} ; %
margin-right:\rndprintlength{\LWR@templengthtwo}%
]{\LWR@mcolstype}%
}
{\endBlockClass}
```

---

File 8 **lwarp-addlines.sty**

## § 89 Package **addlines**

(*Emulates or patches code by* WILL ROBERTSON.)

Pkg addlines

addlines is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{addlines}
```

```
\newcommand*\addlines[1][1]{}
\let\addline\addlines
\newcommand*\removelines[1][1]{}
\let\removeline\removelines
```

---

File 9 **lwarp-afterpage.sty**

## § 90 Package **afterpage**

(*Emulates or patches code by* DAVID CARLISLE.)

Pkg afterpage

Emulated.

**for HTML output:**

Discard all options for lwarp-afterpage:

```
\LWR@ProvidesPackageDrop{afterpage}
```

```
\newcommand{\afterpage}[1]{#1}
```

File 10 **lwarp-algorithmicx.sty**

## § 91 Package **algorithmicx**

(*Emulates or patches code by* Szász János.)

Pkg algorithmicx algorithmicx is supported with minor adjustments.

**for HTML output:**

```
\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 264.1.

**for HTML output:**

```
\begin{warpHTML}
```

```
\AtBeginEnvironment{algorithmic}{%
%
\let\origALG@doentity\ALG@doentity%
%
\renewcommand*{\ALG@doentity}{%
\origALG@doentity%
\uselengthunit{PT}%
\LWR@htmltagc{%
span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
}%
\ifbool{FormatWP}{%
\setlength{\LWR@templengthone}{\the\ALG@thistlm}%
\whiledo{\lengthtest{\LWR@templengthone>1em}}{%
\quad%
\addtolength{\LWR@templengthone}{-1em}%
}%
}{}%
\LWR@htmltagc{/span}%
}%

\let\LWR@origComment\Comment%

\renewcommand{\Comment}[1]{%
    \InlineClass{floatright}{\LWR@origComment{#1}}%
}%
}

\renewcommand\algorithmiccomment[1]{%
\hfill\HTMLunicode{25B7} #1% white right triangle
}%
```

```
\end{warpHTML}
```

File 11 **lwarp-alltt.sty**

## § 92 Package **alltt**

(*Emulates or patches code by* Johannes Braams.)

Pkg alltt alltt is patched for use by lwarp.

**for HTML output:**

```
\LWR@ProvidesPackagePass{alltt}
```

```
\AfterEndPreamble{
\LWR@traceinfo{Patching alltt.}
\AtBeginEnvironment{alltt}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
}
```

File 12 **lwarp-amsthm.sty**

## § 93 Package **amsthm**

(*Emulates or patches code by* Publications Technical Group — American Mathematical Society.)

Pkg amsthm amsthm is patched for use by lwarp.

**for HTML output:**

```
\LWR@ProvidesPackagePass{amsthm}
```

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\renewcommand{\theoremstyle}[1]{%
  \@ifundefined{th@#1}{%
    \PackageWarning{amsthm}{Unknown theoremstyle ‘#1’}%
    \thm@style{plain}%
```

Table 12: AMSthm package — CSS styling of theorems and proofs

**Theorem:** `<div>` of class `amsthmbody<theoremstyle>`

**Theorem Name:** `<span>` of class `amsthmname<theoremtyle>`

**Theorem Number:** `<span>` of class `amsthmnumber<theoremstyle>`

**Theorem Note:** `<span>` of class `amsthmnote<theoremstyle>`

**Proof:** `<div>` of class `amsthmproof`

**Proof Name:** `<span>` of class `amsthmproofname`

where `<theoremstyle>` is `plain`, `definition`, etc.

```
    \renewcommand{\LWR@newtheoremstyle}{plain}% new
  }{%
    \thm@style{#1}%
    \renewcommand{\LWR@newtheoremstyle}{#1}% new
  }%
}
```

Patched to remember the style for this theorem type:

```
\def\@xnthm#1#2{%
  \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
  \let\@tempa\relax
  \@xp\@ifdefinable\csname #2\endcsname{%
    \global\@xp\let\csname end#2\endcsname\@endtheorem
    \ifx *#1% unnumbered, need to get one more mandatory arg
      \edef\@tempa##1{%
        \gdef\@xp\@nx\csname#2\endcsname{%
          \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
            {}{##1}}}%
    \else % numbered theorem, need to check for optional arg
      \def\@tempa{\@oparg{\@ynthm{#2}}[]}%
    \fi
    \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
  }%
  \@tempa
}
```

Patched to enclose with CSS:

```
\newcommand{\LWR@haveamsthmname}{
\renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
}
```

```

\newcommand{\LWR@haveamsthmnumber}{
\renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
}

\newcommand{\LWR@haveamsthmnote}{
\renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
}

\LWR@haveamsthmname
\LWR@haveamsthmnumber
\LWR@haveamsthmnote
```

Patches for css:

```
\def\@begintheorem#1#2[#3]{%
\LWR@forcenewpage% new
  \BlockClass{amsthmbody\LWR@thisthmstyle}% new
  \deferred@thm@head{
    \the\thm@headfont \thm@indent
    \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
    \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
    \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
    \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
    \the\thm@headpunct~
    \thmheadnl % possibly a newline.
    \hskip\thm@headsep
  }%
  \ignorespaces}
```

Patched for css:

```
\def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }
```

Proof QED symbol:

```
\AtBeginDocument{
\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
}
```

Patched for css:

```
\renewenvironment{proof}[1][\proofname]{\par
\LWR@forcenewpage% new
    \BlockClass{amsthmproof}% new
  \pushQED{\qed}%
  \normalfont \topsep6\p@\@plus6\p@\relax
  \trivlist
  \item[\hskip\labelsep
        \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
}{%
  \InlineClass{theoremendmark}{\popQED}\endtrivlist%
  \endBlockClass% new
  \@endpefalse
}
```

---

File 13 `lwarp-anonchap.sty`

# § 94 Package **anonchap**

(*Emulates or patches code by* Peter Wilson.)

Pkg anonchap

anonchap is emulated.

⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

The code is shared by tocbibind.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{anonchap}
```

```
\newcommand{\simplechapter}[1][\@empty]{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\newcommand{\restorechapter}{%
\let\@chapcntformat\@seccntformat%
}
```

---

File 14 `lwarp-anysize.sty`

# § 95 Package **anysize**

(*Emulates or patches code by* Michael Salzenberg, Thomas Esser.)

Pkg anysize

anysize is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{anysize}
```

```
\def\papersize#1#2{}
\def\marginsize#1#2#3#4{}
```

File 15 **lwarp-appendix.sty**

## § 96 Package **appendix**

(*Emulates or patches code by* Peter Wilson.)

Pkg appendix appendix is patched for use by lwarp.

⚠ incorrect TOC link During HTML conversion, the option toc without the option page results in a TOC link to whichever section was before the appendices environment. It is recommended to use both toc and also page at the same time.

**for HTML output:**

```
\LWR@ProvidesPackagePass{appendix}
```

```
\renewcommand*{\@chap@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}

\renewcommand*{\@sec@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}
```

File 16 **lwarp-arabicfront.sty**

## § 97 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{arabicfront}
```

## File 17 lwarp-atbegshi.sty

## § 98 Package **atbegshi**

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg atbegshi Emulated.

**for HTML output:** Discard all options for lwarp-atbegshi:

```
\LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]
```

```
\newcommand*{\AtBeginShipout}[1]{}
\newbox\AtBeginShipoutBox
\newcommand*{\AtBeginShipoutNext}[1]{}
\newcommand*{\AtBeginShipoutFirst}[1]{}
\newcommand*{\AtBeginShipoutDiscard}{}
\newcommand*{\AtBeginShipoutInit}{}
\newcommand*{\AtBeginShipoutAddToBox}[1]{}
\newcommand*{\AtBeginShipoutAddToBoxForeground}[1]{}
\newcommand*{\AtBeginShipoutUpperLeft}[1]{}
\newcommand*{\AtBeginShipoutUpperLeftForeground}[1]{}
\newcommand*{\AtBeginShipoutOriginalShipout}[1]{}
\def\AtBeginShipoutBoxWidth{0pt}
\def\AtBeginShipoutBoxHeight{0pt}
\def\AtBeginShipoutBoxDepth{0pt}

```

## File 18 lwarp-authblk.sty

## § 99 Package **authblk**

(*Emulates or patches code by* Patrick W. Daly.)

Pkg authblk authblk is patched for HTML.

package support ⚠ load order lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

\published and \subtitle If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 55.7.

(*Emulates or patches code by* Patrick W. Daly.)

**for HTML output:** Require that authblk be loaded before titling:

```
\@ifpackageloaded{titling}{
\PackageError{lwarp-authblk}
{Package authblk must be loaded before titling}
{Titling appends authblk's author macro, so authblk must be loaded first.}
}
{}
```

Load authblk:

```
\LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```
\LetLtxMacro\LWRAB@affil\affil

\renewcommand{\affil}[2][]{%
\LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
}
```

Create an HTML break for an \authorcr:

```
\renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

---

File 19 **lwarp-balance.sty**

# § 100 Package balance

(*Emulates or patches code by* Patrick W. Daly.)

Pkg balance Emulated.

**for HTML output:** Discard all options for lwarp-balance:

```
\LWR@ProvidesPackageDrop{balance}
```

```
\newcommand*{\balance}{}
\newcommand*{\nobalance}{}
```

---

File 20 **lwarp-bigdelim.sty**

# § 101 Package bigdelim

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `bigdelim`

bigdelim is used as-is for print or `lateximage`, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `<span>` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`.

⚠ use `\mrowcell`

`\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

| | | |
|---|---|---|
| | a | b |
| left { | c | d |
| | e | f |
| | g | h |

**for HTML output:**

First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case bigdelim was not loaded:

```
\let\ldelim\relax
\let\rdelim\relax
```

Next, load the package's new definitions:

```
\LWR@ProvidesPackagePass{bigdelim}
```

Remember the print-mode versions:

```
\LetLtxMacro\LWR@origldelim\ldelim
\LetLtxMacro\LWR@origrdelim\rdelim
```

`\ldelim` `\rdelim` {⟨*1:delimiter*⟩} {⟨*2:#rows*⟩} {⟨*3:width*⟩} [⟨*4:text*⟩]

```
\RenewDocumentCommand{\ldelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{right}%
\multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
}

\RenewDocumentCommand{\rdelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{left}%
\multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
}
```

When entering a `lateximage`, restore the print-mode versions:

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro{\ldelim}{\LWR@origldelim}%
\LetLtxMacro{\rdelim}{\LWR@origrdelim}%
}
```

---

## File 21 `lwarp-bigstrut.sty`

# § 102 Package **bigstrut**

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `bigstrut` bigstrut is used as-is for print or `lateximage`, and patched for HTML.

**for HTML output:**

```
\LWR@ProvidesPackagePass{bigstrut}
```

```
\LetLtxMacro\LWR@origbigstrut\bigstrut

\renewcommand\bigstrut[1][x]{}

\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
}
```

---

## File 22 `lwarp-bookmark.sty`

# § 103 Package **bookmark**

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `bookmark` bookmark is emulated.

**for HTML output:** Discard all options for lwarp-bookmark:

```
\LWR@ProvidesPackageDrop{bookmark}
```

```
\newcommand*{\bookmarksetup}[1]{}
\newcommand*{\bookmarksetupnext}[1]{}
\newcommand*{\bookmark}[2][]{}
\newcommand*{\bookmarkdefinestyle}[2]{}
```

```
\newcommand*{\bookmarkget}[1]{}
\newcommand{\BookmarkAtEnd}[1]{}
```

File 23 **lwarp-booktabs.sty**

# § 104 Package **booktabs**

(*Emulates or patches code by* Simon Fear.)

Pkg booktabs

booktabs is emulated during HTML output, and used as-is during print output and inside an HTML `lateximage`.

**for HTML output:**

```
\LWR@ProvidesPackagePass{booktabs}
```

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML `lateximage`s. The HTML versions temporarily overwrite these print versions when `tabular` is started.

```
\LetLtxMacro\LWR@origtoprule\toprule
\LetLtxMacro\LWR@origmidrule\midrule
\LetLtxMacro\LWR@origcmidrule\cmidrule
\LetLtxMacro\LWR@origbottomrule\bottomrule
\LetLtxMacro\LWR@origaddlinespace\addlinespace
\LetLtxMacro\LWR@origmorecmidrules\morecmidrules
\LetLtxMacro\LWR@origspecialrule\specialrule
```

File 24 **lwarp-boxedminipage.sty**

# § 105 Package **boxedminipage**

Pkg boxedminipage

boxedminipage is superceded by boxedminipage2e.

**for HTML output:**

```
\LWR@loadnever{boxedminipage}{boxedminipage2e}
```

File 25 **lwarp-boxedminipage2e.sty**

# § 106 Package **boxedminipage2e**

(*Emulates or patches code by* Scott Pakin.)

Pkg boxedminipage2e

boxedminipage2e is emulated.

**for HTML output:** Discard all options for lwarp-boxedminipage2e:

```
\LWR@ProvidesPackageDrop{boxedminipage2e}
```

```
\newenvironment{boxedminipage}{%
\begin{BlockClass}{framebox}%
\minipage%
}
{
\endminipage%
\end{BlockClass}
}
```

File 26 **lwarp-breakurl.sty**

## § 107 Package **breakurl**

(*Emulates or patches code by* Vilar Camara Neto.)

Pkg breakurl breakurl is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{breakurl}
```

```
\LetLtxMacro\burl\url

\NewDocumentCommand{\LWR@burlaltb}{O{} +m m}{%
\LWR@subhyperref{#3}{#2}% order reversed from \href
\endgroup%
\LWR@ensuredoingapar%
}

\DeclareRobustCommand*{\burlalt}{%
\LWR@ensuredoingapar%
\begingroup%
\catcode`\_=12
\LWR@burlaltb%
}
\LetLtxMacro\urlalt\burlalt
```

File 27 **lwarp-cancel.sty**

## § 108 Package **cancel**

Pkg cancel cancel is used as-is for SVG math, and emulated for HTML text output.

**for HTML output:**

```
\LWR@ProvidesPackagePass{cancel}
```

\cancelto is math-only, so is used as-is.

```
\LetLtxMacro\LWR@origcancel\cancel
\LetLtxMacro\LWR@origbcancel\bcancel
\LetLtxMacro\LWR@origxcancel\xcancel

\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\cancel\LWR@origcancel%
\LetLtxMacro\bcancel\LWR@origbcancel%
\LetLtxMacro\xcancel\LWR@origxcancel%
}
```

\LWR@cancelcolor {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
\newcommand{\LWR@cancelcolor}[5]{%
\ifcsempty{#2}%
{\LWR@HTMLtextstyle{#5}{#3}{#1}}%
{\LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}}%
}
```

\cancel {⟨*text*⟩}

```
\DeclareRobustCommand{\cancel}[1]{%
\begingroup%
\CancelColor%
\LWR@findcurrenttextcolor%
\color{black}%
\LWR@cancelcolor{#1}{LWR@tempcolor}{sout}{text-decoration-color}%
    {text-decoration:line-through}%
\endgroup%
}

\LetLtxMacro\bcancel\cancel
\LetLtxMacro\xcancel\cancel
```

File 28 **lwarp-caption2.sty**

## § 109 Package caption2

Pkg caption2 caption2 is not used. The user is recommended to use caption instead.

**for HTML output:**

```
\LWR@loadnever{caption2}{caption}
```

File 29 **lwarp-ccaption.sty**

## § 110 Package **ccaption**

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

**for HTML output:**

```
\LWR@loadnever{ccaption}{caption}
```

File 30 **lwarp-changepage.sty**

## § 111 Package **changepage**

(*Emulates or patches code by* Peter Wilson.)

Pkg changepage changepage is emulated.

**for HTML output:** Discard all options for lwarp-changepage:

```
\LWR@ProvidesPackageDrop{changepage}
```

```
\newif\ifoddpage
\DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
\DeclareRobustCommand{\changetext}[5]{}
\DeclareRobustCommand{\changepage}[9]{}
\newenvironment{adjustwidth}[2]{}{}
\newenvironment{adjustwidth*}[2]{}{}
```

```
\newcommand*{\strictpagecheck}{}
\newcommand*{\easypagecheck}{}
```

File 31 **lwarp-chngpage.sty**

## § 112 Package **chngpage**

Pkg chngpage chngpage is superceded by changepage.

**for HTML output:**

```
\LWR@loadnever{chngpage}{changepage}
```

File 32 **lwarp-chappg.sty**

## § 113 Package **chappg**

(*Emulates or patches code by* Robin Fairbairns.)

Pkg chappg chappg is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{chappg}
```

```
\renewcommand{\pagenumbering}[2][]{}
\providecommand{\chappgsep}{--}
```

File 33 **lwarp-color.sty**

## § 114 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

⚠ missing colors It should be sufficient for the user's document to load color then load xcolor as well.

**for HTML output:**

```
\LWR@ProvidesPackagePass{color}
\RequirePackage{xcolor}
```

File 34 **lwarp-crop.sty**

## § 115 Package **crop**

(*Emulates or patches code by* Melchior FRANZ.)

Pkg crop Emulated.

**for HTML output:** Discard all options for lwarp-crop:

```
\LWR@ProvidesPackageDrop{crop}
```

```
\newcommand*{\crop}[1][]{}
\newcommand*{\cropdef}[6][]{}
```

File 35 **lwarp-cuted.sty**

## § 116 Package **cuted**

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg cuted cuted is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{cuted}
```

```
\newenvironment{strip}{}{}
\newskip\stripsep
\def\oldcolsbreak#1{}
```

File 36 **lwarp-cutwin.sty**

## § 117 Package **cutwin**

(*Emulates or patches code by* Peter Wilson and Alan Hoenig.)

Pkg cutwin Emulated.

**for HTML output:** Discard all options for lwarp-cutwin:

```
\LWR@ProvidesPackageDrop{cutwin}
```

```
\newcommand*{\opencutleft}{}
\newcommand*{\opencutright}{}
\newcommand*{\opencutcenter}{}
\newcommand*{\cutfuzz}{}

\newenvironment{cutout}[4]
{\marginpar{\windowpagestuff}}
{}

\newcommand*{\windowpagestuff}{}

\newcommand*{\pageinwindow}{%
% \begin{minipage}{.3\linewidth}
\windowpagestuff
% \end{minipage}
}

```

```
\newenvironment{shapedcutout}[3]
{\marginpar{\picinwindow}}
{}

\newcommand*{\putstuffinpic}{}

\newcommand*{\picinwindow}{%
\begin{picture}(0,0)
\putstuffinpic
\end{picture}}
```

File 37 **lwarp-dblfnote.sty**

# § 118 Package **dblfnote**

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg dblfnote dblfnote is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{dblfnote}
```

```
\newcounter{DFNsloppiness}
\newdimen\DFNcolumnsep
\newdimen\DFNcolumnwidth
\def\DFNallowcbreak{}
\def\DFNinhibitcbreak{}
\def\DFNtrysingle{}
\def\DFNalwaysdouble{}
\def\DFNruleboth{}
\def\DFNruleleft{}
```

File 38 **lwarp-dcolumn.sty**

# § 119 Package **dcolumn**

Pkg dcolumn dcolumn is emulated by the lwarp core.

```
\LWR@ProvidesPackageDrop{dcolumn}
```

## File 39 lwarp-draftwatermark.sty

# § 120 Package draftwatermark

(*Emulates or patches code by* Sergio Callegari.)

Pkg draftwatermark

draftwatermark is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{draftwatermark}
```

```
\newcommand{\SetWatermarkAngle}[1]{}
\newcommand{\SetWatermarkColor}[1]{}
\newcommand{\SetWatermarkLightness}[1]{}
\newcommand{\SetWatermarkFontSize}[1]{}
\newcommand{\SetWatermarkScale}[1]{}
\newcommand{\SetWatermarkHorCenter}[1]{}
\newcommand{\SetWatermarkVertCenter}[1]{}
\newcommand{\SetWatermarkText}[1]{}
```

## File 40 lwarp-easy-todo.sty

# § 121 Package easy-todo

(*Emulates or patches code by* Juan Rada-Vilela.)

Pkg easy-todo

easy-todo is patched for use by lwarp.

**for HTML output:**

```
\LWR@ProvidesPackagePass{easy-todo}
```

\listoftodos

Modified to correct buggy use of \flushright.

```
\let\LWR@origlistoftodos\listoftodos

\renewcommand{\listoftodos}{%
\begingroup
\renewcommand{\flushright}{}
\LWR@origlistoftodos
\endgroup
}
```

\todoii

Modified to use \textcolor instead of \color.

```
\renewcommand{\todoii}[2]{%
\ifthenelse{\equal{\@todoobeyfinal}{true}}{%
    \ifoptionfinal{\todoenable{false}}{\todoenable{true}}%
}{}%
\ifthenelse{\equal{\@todoenable}{true}}{%
\refstepcounter{todos}%
\noindent{%
    \todocolor%
    \LWR@textcurrentcolor{%
        \normalfont\scriptsize{\bfseries{\thetodos.#1}}%
    }%
}%
\addcontentsline{lod}{todos}{\protect{\thetodos. }#2}%
}{}%
}
```

File 41 **`lwarp-ebook.sty`**

## § 122 Package **ebook**

(*Emulates or patches code by* Jørgen Steensgaard.)

Pkg `ebook` ebook is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{ebook}
```

```
\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{2}

\providecommand{\pagefill}[1][0.001mm]{\noindent}

\providecommand{\ebook}{
\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{2}
}
```

File 42 **`lwarp-ellipsis.sty`**

## § 123 Package **ellipsis**

(*Emulates or patches code by* Peter J. Heslin.)

Pkg `ellipsis` ellipsis is emulated.

```
\LWR@ProvidesPackageDrop{ellipsis}
```

```

\newcommand{\ellipsisgap}{0.1em}
```

File 43 **`lwarp-emptypage.sty`**

## § 124 Package **emptypage**

Pkg `emptypage` emptypage is ignored.

**for HTML output:** Discard all options for lwarp-emptypage:

```
\LWR@ProvidesPackageDrop{emptypage}
```

File 44 **`lwarp-endnotes.sty`**

## § 125 Package **endnotes**

(*Emulates or patches code by* John Lavagnino.)

Pkg `endnotes` Used as-is.

table of contents To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\enoteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

**for HTML output:**

```
\LWR@ProvidesPackagePass{endnotes}
```

```
\def\enoteformat{%
% \rightskip\z@ \leftskip\z@ \parindent=1.8em
\leavevmode
% \llap{
\makeenmark
% }
}

\def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
\def\makeenmark{\@makeenmark}
```

## File 45 lwarp-enumerate.sty

## § 126 Package enumerate

Pkg enumerate

enumerate is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

**for HTML output:**

```
\LWR@ProvidesPackagePass{enumerate}
```

## File 46 lwarp-enumitem.sty

## § 127 Package enumitem

(*Emulates or patches code by* Javier Bezos.)

Pkg enumitem

enumitem is supported with minor adjustments.

**for HTML output:**

```
\LWR@ProvidesPackagePass{enumitem}
```

**for HTML output:**

```
\begin{warpHTML}
```

`\newlist` {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}
`\renewlist` {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}

For enumitem lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```
\let\LWR@orignewlist\newlist

\renewcommand*{\newlist}[3]{%
\LWR@orignewlist{#1}{#2}{#3}%
\AtBeginEnvironment{#1}{\csuse{LWR@#2start}}%
\AtEndEnvironment{#1}{\csuse{LWR@#2end}}%
}
```

```
\end{warpHTML}
```

File 47 **lwarp-epigraph.sty**

## § 128 Package **epigraph**

(*Emulates or patches code by* Peter Wilson.)

Pkg epigraph

epigraph is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{epigraph}
```

```
\newcommand{\qitem}[2]
{
\begin{BlockClass}{qitem}
#1
\ifbool{FormatWP}
{\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
{\begin{BlockClass}{epigraphsource}}
#2
\end{BlockClass}
\end{BlockClass}
}
```

```
\newcommand{\epigraph}[2]
{
\begin{LWR@BlockClassWP}{text-align:right}{}{epigraph}
\qitem{#1}{#2}
\end{LWR@BlockClassWP}
}

\newenvironment*{epigraphs}
{\LWR@BlockClassWP{text-align:right}{}{epigraph}}
{\endLWR@BlockClassWP}
```

Use css to format epigraphs.

The following are null commands for source compatibility:

```
\newlength{\epigraphwidth}
\setlength{\epigraphwidth}{.5\linewidth}
\newenvironment*{flushepinormal}{}{}
\newcommand{\textflush}[1]{flushepinormal}
\newcommand{\epigraphflush}[1]{flushright}
\newcommand{\sourceflush}[1]{flushright}
\newcommand*{\epigraphsize}{\small}
\newlength{\epigraphrule}
```

```
\newlength{\beforeepigraphskip}
\newlength{\afterepigraphskip}
\newcommand{\epigraphhead}[2][0]{#2}
\newcommand{\dropchapter}[1]{}
\newcommand*{\undodrop}{}
\newcommand{\cleartoevenpage}[1][]{}
```

## File 48 `lwarp-eso-pic.sty`

# § 129 Package **eso-pic**

(*Emulates or patches code by* ROLF NIEPRASCHK.)

Pkg `eso-pic` eso-pic is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{eso-pic}
```

```
\newcommand*{\LenToUnit}{}
\newcommand{\AtPageUpperLeft}[1]{}
\newcommand{\AtPageLowerLeft}[1]{}
\newcommand{\AtPageCenter}[1]{}
\newcommand{\AtStockLowerLeft}[1]{}
\newcommand{\AtStockUpperLeft}[1]{}
\newcommand{\AtStockCenter}[1]{}
\newcommand{\AtTextUpperLeft}[1]{}
\newcommand{\AtTextLowerLeft}[1]{}
\newcommand{\AtTextCenter}[1]{}
\NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
\renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
\NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
\newcommand*{\ClearShipoutPictureBG}{}
\newcommand*{\ClearShipoutPicture}{}
\newcommand*{\ClearShipoutPictureFG}{}
\newcommand{\gridSetup}[6][]{}
```

## File 49 `lwarp-everypage.sty`

# § 130 Package **everypage**

(*Emulates or patches code by* SERGIO CALLEGARI.)

Pkg `everypage` everypage is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{everypage}
```

```
\newcommand*{\AddEverypageHook}[1]{}
\newcommand*{\AddThispageHook}[1]{}
```

File 50 **lwarp-everyshi.sty**

## § 131 Package **everyshi**

(*Emulates or patches code by* Martin Schröder.)

Pkg everyshi Emulated.

**for HTML output:** Discard all options for lwarp-everyshi:

```
\LWR@ProvidesPackageDrop{everyshi}
```

```
\newcommand*{\EveryShipout}[1]{}
\newcommand*{\AtNextShipout}[1]{}
```

File 51 **lwarp-extramarks.sty**

## § 132 Package **extramarks**

(*Emulates or patches code by* Piet van Oostrum.)

Pkg extramarks extramarks is emulated.

**for HTML output:** Discard all options for lwarp-extramarks:

```
\LWR@ProvidesPackageDrop{extramarks}
```

```
\newcommand*{\extramarks}[2]{}
\newcommand*{\firstleftxmark}{}
\newcommand*{\lastleftxmark}{}
\newcommand*{\firstrightxmark}{}
\newcommand*{\lastrightxmark}{}
\newcommand*{\firstxmark}{}
\newcommand*{\lastxmark}{}
\newcommand*{\topxmark}{}
\newcommand*{\topleftxmark}{}
\newcommand*{\firstleftmark}{}
\newcommand*{\lastrightmark}{}
```

File 52 **lwarp-fancybox.sty**

# § 133 Package **fancybox**

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg fancybox

fancybox is supported with some patches.

framed equation example

fancybox's documentation has an example FramedEqn environment which combines math, \Sbox, a minipage, and an \fbox. This combination requires that the entire environment be enclosed inside a lateximage, which is done by adding \lateximage at the very start of FramedEqn's beginning code, and \endlateximage at the very end of the ending code. Unfortunately, the HTML alt attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives

\fbox works with fancybox. Also see lwarp's \fboxBlock macro and fminipage environment for alternatives to \fbox for framing environments.

framed table example

The fancybox documentation's example framed table using an \fbox containing a tabular does not work with lwarp, but the FramedTable environment does work if \fbox is replaced by \fboxBlock. This method loses HTML formatting. A better method is to enclose the table's contents inside a fminipage environment. The caption may be placed either inside or outside the fminipage:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim

lwarp does not support the verbatim environment inside a span, box, or fancybox's \Sbox, but a verbatim may be placed inside a fminipage. The fancybox documentation's example FramedVerb may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed \VerbBox fancybox's \VerbBox may be used inside \fbox.

indented alignment LVerbatim, \LVerbatimInput, and \LUseVerbatim indent with horizontal space which may not line up exactly with what pdftotext detects. Some lines may be off slightly in their left edge.

**for HTML output:**

```
\begin{warpHTML}
```

```
\LWR@ProvidesPackagePass{fancybox}
```

```
\renewcommand*{\@shadowbox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black]{shadowbox}{#1}}%
{\InlineClass{shadowbox}{#1}}%
}

\renewcommand*{\@doublebox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px double black]{doublebox}{#1}}%
{\InlineClass{doublebox}{#1}}%
}

\renewcommand*{\@ovalbox}[2]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
{%
    \ifthenelse{\isequivalentto{#1}{\thinlines}}%
    {\InlineClass{ovalbox}{#2}}%
    {\InlineClass{Ovalbox}{#2}}%
}%
}
```

Convert minipages, parboxes, and lists into linear text using the LWR@nestspan environment:

```
\let\LWR@origSbox\Sbox

\def\Sbox{\LWR@origSbox\LWR@nestspan}

```

```

\let\LWR@origendSbox\endSbox

\def\endSbox{\endLWR@nestspan\LWR@origendSbox}
```

Beqnarray is adapted for MathJax or enclosed inside a lateximage:

```
\RenewEnviron{Beqnarray}
{\LWR@eqnarrayfactor}

\csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

\GenericCaption is enclosed in an HTML block:

```
\renewcommand{\GenericCaption}[1]{%
\LWR@figcaption%
#1%
\endLWR@figcaption%
}
```

Btrivlist is enclosed in an HTML block:

```
\RenewDocumentEnvironment{Btrivlist}{m o}
{\begin{BlockClass}{Btrivlist}\tabular{#1}}
{\endtabular\end{BlockClass}}
```

Btrivlist is also neutralized when used inside a span:

```
\AtBeginEnvironment{LWR@nestspan}{%
\RenewDocumentEnvironment{Btrivlist}{m o}{}{}%
\RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
}
```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```
\let\LWRFB@origitemizeitem\LWR@itemizeitem
\let\LWRFB@origdescitem\LWR@descitem
\LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}

\RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
\RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}
```

The various boxed lists become regular lists:

```
\renewenvironment{Bitemize}[1][]{\begin{itemize}}{\end{itemize}}
\renewenvironment{Benumerate}[1][]{\begin{enumerate}}{\end{enumerate}}
\renewenvironment{Bdescription}[1][]{\begin{description}}{\end{description}}
```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```
\RenewDocumentCommand{\boxput}{s d() m m}{%
\IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
}
```

Neutralized commands:

```
\RenewDocumentCommand{\fancyput}{s d() m}{}
\RenewDocumentCommand{\thisfancyput}{s d() m}{}

\RenewDocumentCommand{\fancypage}{m m}{}
\RenewDocumentCommand{\thisfancypage}{m m}{}

\def\LandScape#1{}
\def\endLandScape{}
\def\@Landscape#1#2#3{}
\def\endLandscape{}
```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```
\let\LWRFB@UseVerbatim\UseVerbatim
\renewcommand*{\UseVerbatim}[1]{%
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
\LWRFB@UseVerbatim{#1}%
\LWR@afterendverbatim%
}

\let\LWRFB@LUseVerbatim\LUseVerbatim

\renewcommand*{\LUseVerbatim}[1]{%
\LWR@atbeginverbatim{LVerbatim}%\unskip\LWR@origvspace*{-\baselineskip}%%
\noindent%
\LWRFB@LUseVerbatim{#1}%
\LWR@afterendverbatim%
}

\def\@BUseVerbatim[#1]#2{%
\LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
\LWRFB@UseVerbatim{#2}%
\LWR@afterendverbatim%
}
```

```
\end{warpHTML}
```

File 53 **lwarp-fancyhdr.sty**

## § 134 Package **fancyhdr**

(*Emulates or patches code by* Piet van Oostrum.)

Pkg fancyhdr

fancyhdr is nullified.

**for HTML output:** Discard all options for lwarp-fancyhdr:

```
\LWR@ProvidesPackageDrop{fancyhdr}
```

```
\newcommand*{\fancyhead}[2][]{}
\newcommand*{\fancyfoot}[2][]{}
\newcommand*{\fancyhf}[2][]{}
\newcommand*{\fancypagestyle}[2]{}
\newcommand*{\lhead}[2][]{}
\newcommand*{\chead}[2][]{}
\newcommand*{\rhead}[2][]{}
\newcommand*{\lfoot}[2][]{}
\newcommand*{\cfoot}[2][]{}
\newcommand*{\rfoot}[2][]{}
\newcommand*{\headrulewidth}{}
\newcommand*{\footrulewidth}{}
\newcommand*{\fancyheadoffset}[2][]{}
\newcommand*{\fancyfootoffset}[2][]{}
\newcommand*{\fancyhfoffset}[2][]{}
\newcommand*{\iffloatpage}[2]{#2}
\newcommand*{\ifftopfloat}[2]{#2}
\newcommand*{\iffbotfloat}[2]{#2}
```

File 54 **lwarp-fancyref.sty**

## § 135 Package **fancyref**

Pkg fancyref

fancyref is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{fancyref}
```

```
\newcommand*{\fancyrefhook}[1]{#1}
\newcommand*{\fref}[2][]{\cref{#2}}
\newcommand*{\Fref}[2][]{\Cref{#2}}
\newcommand*{\fancyrefchaplabelprefix}{chap}
```

```
\newcommand*{\fancyrefenumlabelprefix}{enum}
\newcommand*{\fancyrefeqlabelprefix}{eq}
\newcommand*{\fancyreffiglabelprefix}{fig}
\newcommand*{\fancyreffnlabelprefix}{fn}
\newcommand*{\fancyrefseclabelprefix}{sec}
\newcommand*{\fancyreftablabelprefix}{tab}
\newcommand*{\fancyrefchangeprefix}[2]{}
\newcommand*{\fancyrefargdelim}{:}
\newcommand*{\fancyrefloosespacing}{~}
\newcommand*{\fancyreftightspacing}{\,}
\newcommand*{\fancyrefdefaultspacing}{\fancyrefloosespacing}
\providecommand*{\chaptername}{Chapter}
\providecommand*{\figurename}{Figure}
\providecommand*{\pagename}{Page}
\providecommand*{\tablename}{Table}
\renewcommand*{\Frefchapname}{\chaptername}
\renewcommand*{\Frefenumname}{Item}
\renewcommand*{\Frefeqname}{Equation}
\renewcommand*{\Freffigname}{\figurename}
\renewcommand*{\Freffnname}{Footnote}
\renewcommand*{\Frefonname}{On}
\renewcommand*{\Frefpgname}{\pagename}
\renewcommand*{\Frefsecname}{Section}
\renewcommand*{\Frefseename}{See}
\renewcommand*{\Freftabname}{\tablename}
\renewcommand*{\Freffigshortname}{Fig.}
\renewcommand*{\Frefpgshortname}{P.}
\renewcommand*{\Freftabshortname}{Tab.}
\renewcommand*{\frefchapname}{\MakeLowercase{\Frefchapname}}
\renewcommand*{\frefenumname}{\MakeLowercase{\Frefenumname}}
\renewcommand*{\frefeqname}{\MakeLowercase{\Frefeqname}}
\renewcommand*{\freffigname}{\MakeLowercase{\Freffigname}}
\renewcommand*{\freffnname}{\MakeLowercase{\Freffnname}}
\renewcommand*{\frefonname}{\MakeLowercase{\Frefonname}}
\renewcommand*{\frefpgname}{\MakeLowercase{\Frefpgname}}
\renewcommand*{\frefsecname}{\MakeLowercase{\Frefsecname}}
\renewcommand*{\frefseename}{\MakeLowercase{\Frefseename}}
\renewcommand*{\freftabname}{\MakeLowercase{\Freftabname}}
\renewcommand*{\freffigshortname}{\MakeLowercase{\Freffigshortname}}
\renewcommand*{\frefpgshortname}{\MakeLowercase{\Frefpgshortname}}
\renewcommand*{\freftabshortname}{\MakeLowercase{\Freftabshortname}}
\newcommand*{\fancyrefaddcaptions}[2]{}
\newcommand*{\fancyrefdefaultformat}{\@empty}
\newcommand{\frefformat}[3]{}
\newcommand{\Frefformat}[3]{}
```

File 55 **lwarp-fancyvrb.sty**

## § 136 Package **fancyvrb**

(*Emulates or patches code by* TIMOTHY VAN ZANDT.)

Pkg fancyvrb

fancyvrb is supported with some patches.

**for HTML output:**

```
\RequirePackage{xcolor}% for \convertcolorspec
\LWR@ProvidesPackagePass{fancyvrb}
```

**for HTML output:**

```
\begin{warpHTML}
```

Initial default patch for fancyvrb:

```
\fvset{frame=none}%
```

For `\VerbatimFootnotes`:

```
\renewcommand{\VerbatimFootnotes}{
\PackageError{lwarp}
{Verbatim footnotes are not yet supported by lwarp.}
{This may be improved some day.}
}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
\AfterEndPreamble{
\LWR@traceinfo{Patching Verbatim.}
```

Remember the original defintion of `Verbatim`:

```
\let\LWRFV@origVerbatim\Verbatim
```

Env Verbatim

Patched to place the environment in a `fancyvrb` div, and the label in a `fancyvrblabel` div. Also corrects the left margin for line numbers. Also uses `VerbatimHTMLWidth` to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
\renewcommand*{\Verbatim}{%
\LWR@forcenewpage
\LWRFV@origVerbatim%
}
```

\LWR@FVstyle Holds the style of the verbatim.

```
\newcommand*{\LWR@FVstyle}{}
```

The following patches to Verbatim are executed at the start and end of the environment, depending on the choice of frame. Original code is from the fancyvrb package.

```
\newcommand*{\LWR@fvstartnone}{%
\LWR@traceinfo{fvstartnone}%
\hbox to\z@{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
}

\newcommand*{\LWR@fvendnone}{%
\LWR@traceinfo{fvendnone}%
\hbox to\z@{\LWR@afterendverbatim}%
}

\newcommand*{\LWR@fvstartsingle}{%
\LWR@traceinfo{fvstartsingle}%
\LWR@fvstartnone%
\FV@BeginListFrame@Single%
}

\newcommand*{\LWR@fvendsingle}{%
\LWR@traceinfo{fvendsingle}%
\FV@EndListFrame@Single%
\LWR@fvendnone%
}

\newcommand*{\LWR@fvstartline}{%
\LWR@traceinfo{fvstartline}%
\LWR@fvstartnone%
\FV@BeginListFrame@Lines%
}

\newcommand*{\LWR@fvendline}{%
\LWR@traceinfo{fvendline}%
\FV@EndListFrame@Lines%
\LWR@fvendnone%
}
```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```
\newcommand*{\LWR@FVfindbordercolor}{%
\FancyVerbRuleColor%
\LWR@findcurrenttextcolor%
\color{black}%
```

```
}

% border width of \FV@FrameRule
\newcommand*{\LWR@FVborderstyle}[1]{%
padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
\LWR@FVfindbordercolor %
border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ; %
}

\def\FV@Frame@none{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\FV@Frame@none% default values

\def\FV@Frame@single{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
\let\FV@BeginListFrame\LWR@fvstartsingle%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\FV@RightListFrame@Single%
\let\FV@EndListFrame\LWR@fvendsingle}

\def\FV@Frame@lines{%
\renewcommand*{\LWR@FVstyle}{%
    \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendline}

\def\FV@Frame@topline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\def\FV@Frame@bottomline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendline}

\def\FV@Frame@leftline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%
```

```
% To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
\ifx\FancyVerbFillColor\relax%
\let\FV@FrameFillLine\relax%
\else%
\@tempdima\FV@FrameRule\relax%
\multiply\@tempdima-\tw@%
\edef\FV@FrameFillLine{%
{\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
\kern-\number\@tempdima sp}}%
\fi%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}
```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```
\def\FV@SingleFrameLine#1{%
  \hbox to\z@{%
%     \kern\leftmargin
    \ifnum#1=\z@\relax
      \let\FV@Label\FV@LabelBegin
    \else
      \let\FV@Label\FV@LabelEnd
    \fi
    \ifx\FV@Label\relax
%       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
    \else
      \ifnum#1=\z@
%         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
        \ifx\FV@LabelPositionTopLine\relax
        \else
        \LWR@FVfindbordercolor
        \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
        \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
        \LWR@htmltagc{/div}
        \fi
      \else
%         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
        \ifx\FV@LabelPositionBottomLine\relax
        \else
        \LWR@FVfindbordercolor
        \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
        \LWR@origtextrm{\FV@LabelEnd}
        \LWR@htmltagc{/div}
        \fi
      \fi

    \fi
    \hss
    }
}
```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```
\def\FV@ListProcessLine#1{%
   \hbox to \hsize{%
%      \kern\leftmargin
      \hbox to \VerbatimHTMLWidth {%
       \ifcsvoid{FV@LeftListNumber}{}{\kern 2.5em}%
        \FV@LeftListNumber%
%       \FV@LeftListFrame
      \FancyVerbFormatLine{#1}%
      \hss%
%       \FV@RightListFrame
      \FV@RightListNumber%
    }%
      \hss% required to avoid underfull hboxes
}
}
```

Env `BVerbatim`

```
\AtBeginEnvironment{BVerbatim}
{
\LWR@forcenewpage
\LWR@atbeginverbatim{bverbatim}

}

\AfterEndEnvironment{BVerbatim}
{
\leavevmode\par\LWR@origvspace{-\baselineskip}
\LWR@afterendverbatim
}
```

End of the modifications to make at the end of the preamble:

```
} % \AfterEndPreamble
```

```
\end{warpHTML}
```

File 56 **lwarp-figcaps.sty**

## § 137 Package **figcaps**

(*Emulates or patches code by* Patrick W. Daly.)

Pkg figcaps Emulated.

**for HTML output:** Discard all options for lwarp-figcaps:

```
\LWR@ProvidesPackageDrop{figcaps}
```

```
\newcommand*{\figcapson}{}
\newcommand*{\figcapsoff}{}
\newcommand*{\printfigures}{}
\newcommand*{\figmarkon}{}
\newcommand*{\figmarkoff}{}
\def\figurecapname{Figure Captions}
\def\tablepagename{Tables}
\def\figurepagename{Figures}
```

File 57 **lwarp-fix2col.sty**

## § 138 Package **fix2col**

Pkg fix2col fix2col is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{fix2col}
```

File 58 **lwarp-fixme.sty**

## § 139 Package **fixme**

(*Emulates or patches code by* Didier Verna.)

Pkg fixme fixme is patched for use by lwarp.

⚠ external layouts External layouts (\fxloadlayouts) are not supported.

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after fixme is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

**for HTML output:**

```
\LWR@ProvidesPackagePass{fixme}
```

Restore lwarp's version of `\@wrindex`, ignoring the fixme package's `target` option:

```
\let\@wrindex\LWR@wrindex
```

Float-related macros required by lwarp:

```
\newcommand{\ext@fixme}{lox}

\renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}
```

Other modifications:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}

\renewcommand*\FXLayoutInline[3]{ %
\InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
    {\@fxtextstd{#1}{#2}{#3}}%
}

\def\FXFaceEnvHTMLStyle{font-weight:bold}

\renewcommand*\FXEnvLayoutPlainBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\ignorespaces#2 \fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutPlainEnd[2]{\endBlockClass}

\renewcommand*\FXEnvLayoutSignatureBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutSignatureEnd[2]{\@fxsignature{#2}\endBlockClass}

\def\FXFaceSignatureHTMLStyle{font-style:italic}

\DeclareRobustCommand*\@fxsignature[1]{%
\ifthenelse{\equal{#1}{}}%
{}%
{ -- {\InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{#1}}}%
}


```

```
\def\FXFaceTargetHTMLStyle{font-style:italic}

\renewcommand\FXTargetLayoutPlain[2]{%
    \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
}
```

File 59 **lwarp-fixmetodonotes.sty**

## § 140 Package **fixmetodonotes**

(*Emulates or patches code by* Gioele Barabucci.)

Pkg fixmetodonotes

fixmetodonotes is patched for use by lwarp.

**for HTML output:**

```
\LWR@ProvidesPackagePass{fixmetodonotes}
```

```
\renewcommand{\NOTES@addtolist}[2]{%
    \refstepcounter{NOTES@note}%
%   \phantomsection% REMOVED
    \addcontentsline{notes}{NOTES@note}{%
        \protect\numberline{\theNOTES@note}{{#1}: {#2}}%
    }%
}

\renewcommand{\NOTES@marker}[2]{\fbox{%
    \textcolor{#2}{% WAS \color
        \textbf{#1}}%
    }}

\renewcommand{\NOTES@colorline}[2]{%
  \bgroup%
    \ULon{\LWR@backgroundcolor{#1}{#2}}%
}
```

File 60 **lwarp-float.sty**

## § 141 Package **float and \newfloat**

(*Emulates or patches code by* Anselm Lingnau.)

Pkg float

float is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{float}[2016/03/04]
```

See section 64.2 for the \listof command.

`\newfloat` {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the `\newfloat` command from the float package.

“placement” is ignored.

```
\NewDocumentCommand{\newfloat}{m m m o}{%
\IfValueTF{#4}
{\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
{\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.

```
\cslet{listof#1s}\relax
\cslet{listof#1es}\relax
}
```

`\floatname` {⟨*type*⟩} {⟨*name*⟩}

Sets the text name of the float, such as “Figure”.

```
\NewDocumentCommand{\floatname}{m +m}{%
\SetupFloatingEnvironment{#1}{name=#2}%
}
```

`\floatplacement` {⟨*type*⟩} {⟨*placement*⟩}

Float placement is ignored.

```
\newcommand*{\floatplacement}[2]{%
\SetupFloatingEnvironment{#1}{placement=#2}%
}
```

`\floatstyle` {⟨*style*⟩}

Float styles are ignored.

```
\newcommand{\floatstyle}[1]{%
}
```

`\restylefloat` * {⟨*style*⟩}

Float styles are ignored.

```
\NewDocumentCommand{\restylefloat}{s m}{%
}
```

## File 61 lwarp-floatflt.sty

## § 142 Package **floatflt**

(*Emulates or patches code by* Mats Dahlgren.)

Pkg floatflt Emulated.

**for HTML output:** Discard all options for lwarp-floatflt:

```
\LWR@ProvidesPackageDrop{floatflt}
```

Env [⟨⟩] offset {⟨*type*⟩} {⟨*width*⟩} Borrowed from the lwarp version of keyfloat:

```
\NewDocumentEnvironment{KFLTfloatflt@marginfloat}{O{-1.2ex} m m}
{%
\setlength{\LWR@templengthone}{#3}%
\uselengthunit{PT}%
\LWR@BlockClassWP{%
    float:right; %
    width:\rndprintlength{\LWR@templengthone}; %
    margin:10pt%
}{%
    width:\rndprintlength{\LWR@templengthone}%
}%
{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}
```

Env floatingfigure [⟨*placement*⟩] {⟨*width*⟩}

```
\DeclareDocumentEnvironment{floatingfigure}{o m}
  {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
  {\end{KFLTfloatflt@marginfloat}}
```

Env floatingtable [⟨*placement*⟩]

```
\DeclareDocumentEnvironment{floatingtable}{o}
  {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
  {\end{KFLTfloatflt@marginfloat}}
```

File 62 **lwarp-floatpag.sty**

## § 143 Package floatpag

(*Emulates or patches code by* Vytas Statulevičius and Sigitas Tolušis.)

Pkg floatpag Emulated.

**for HTML output:** Discard all options for lwarp-floatpag:

```
\LWR@ProvidesPackageDrop{floatpag}
```

```
\newcommand*{\floatpagestyle}[1]{}
\newcommand*{\rotfloatpagestyle}[1]{}
\newcommand*{\thisfloatpagestyle}[1]{}
```

File 63 **lwarp-floatrow.sty**

## § 144 Package floatrow

(*Emulates or patches code by* Olga Lapko.)

Pkg floatrow floatrow is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{floatrow}
```

⚠ subfig package When combined with the subfig package, while inside a subfloatrow \ffigbox and \ttabbox must have the caption in the first of the two of the mandatory arguments.

⚠ \FBwidth, \FBheight The emulation of floatrow does not support \FBwidth or \FBheight. These values are pre-set to .3\linewidth and 2in. Possible solutions include:

- Use fixed lengths. lwarp will scale the HTML lengths appropriately.
- Use warpprint and warpHTML environments to select appropriate values for each case.
- Inside a warpHTML environment, manually change \FBwidth or \FBheight before the \ffigbox or \ttabbox. Use \FBwidth or \FBheight normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
\newbool{LWR@subcaptionloaded}

\AtBeginDocument{
\@ifpackageloaded{subcaption}
{\booltrue{LWR@subcaptionloaded}}
{\boolfalse{LWR@subcaptionloaded}}
}
```

`\floatbox` [⟨*1 preamble*⟩] {⟨*2 captype*⟩} [⟨*3 width*⟩] [⟨*4 height*⟩] [⟨*5 vert pos*⟩] {⟨*6 caption*⟩} {⟨*7 object*⟩}

Only parameters for captype, width, caption, and object are used.

`LWR@insubfloatrow` is true if inside a `subfloatrow` environment.

There are two actions, depending on the use of subcaption or subfig.

```
\NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
\ifbool{LWR@subcaptionloaded}%
{% subcaption
```

For subcaption:

```
    \ifbool{LWR@insubfloatrow}%
    {% subcaption in a subfloatrow
```

`subfigure` and `subtable` environments take width as an argument.

```
        \IfValueTF{#3}%
        {\@nameuse{sub#2}{#3}}%
        {\@nameuse{sub#2}{\linewidth}}%
    }% subcaption in a subfloatrow
    {% subcaption not in subfloatrow
```

`figure` and `table` environments do not take a width argument.

```
        \@nameuse{#2}%
    }% subcaption not in subfloatrow
    #6

    #7
```

End the environments:

```
    \ifbool{LWR@insubfloatrow}%
    {\@nameuse{endsub#2}}%
    {\@nameuse{end#2}}%
}% subcaption
{% assume subfig
```

For subfig:

```
\ifbool{LWR@insubfloatrow}%
{% subfig in a subfloatrow
```

`\subfloat` is a macro, not an environment.

Package subfig's `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```
    \begingroup
    \let\caption\@firstofone
    \subfloat[#6]{#7}
    \endgroup
}% subfig in a subfloatrow
{% subfig package, but not a subfig
```

`figure` and `table` are environments:

```
\@nameuse{#2}
#6

#7
\@nameuse{end#2}
}% subfig package, but not a subfig
}% assume subfig
}
```

Not used:

```
\newcommand*{\nocapbeside}{}
\newcommand*{\capbeside}{}
\newcommand*{\captop}{}
\newlength{\FBwidth}
\setlength{\FBwidth}{.3\linewidth}
\newlength{\FBheight}
\setlength{\FBheight}{2in}
\newcommand*{\useFCwidth}{}
\newcommand{\floatsetup}[2][]{}
\newcommand{\thisfloatsetup}[1]{}
\newcommand{\clearfloatsetup}[1]{}
\newcommand*{\killfloatstyle}{}
```

`\newfloatcommand` {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\newfloatcommand}{m m o o}{%
\@namedef{#1}{
\floatbox{#2}
}
}
```

\renewfloatcommand {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\renewfloatcommand}{m m o o}{%
\@namedef{#1}{%
\floatbox{#2}
}
}
```

\ffigbox [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ffigbox}{figure}[\nocapbeside][]
```

\ttabbox [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

\fcapside [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{fcapside}{figure}[\capbeside][]
```

Env floatrow [⟨*numfloats*⟩]

The row of floats is placed into a `<div>` of class `floatrow`.

```
\newenvironment*{floatrow}[1][2]
{
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, divide the `\linewidth` by the number of floats.

```
\booltrue{LWR@infloatrow}
\setlength{\linewidth}{6in/#1}
}
{
\boolfalse{LWR@infloatrow}
\endBlockClass
}
```

Keys for `\DeclareNewFloatType`:

```
\newcommand*{\LWR@frowkeyplacement}{}
\newcommand*{\LWR@frowkeyname}{}
\newcommand*{\LWR@frowkeyfileext}{}
\newcommand*{\LWR@frowkeywithin}{}
\newcommand*{\LWR@frowkeycapstyle}{}

\define@key{frowkeys}{placement}{}%
```

```
\define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
\define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
\define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
\define@key{frowkeys}{relatedcapstyle}{}%
```

`\DeclareNewFloatType` {⟨*type*⟩} {⟨*options*⟩}

Use `\listof{type}{Title}` to print a list of the floats.

```
\newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
\renewcommand*{\LWR@frowkeyplacement}{}
\renewcommand*{\LWR@frowkeyname}{}
\renewcommand*{\LWR@frowkeyfileext}{}
\renewcommand*{\LWR@frowkeywithin}{}
\renewcommand*{\LWR@frowkeycapstyle}{}
```

Read new key values:

```
\LWR@traceinfo{about to setkeys frowkeys}%
\setkeys{frowkeys}{#2}%
\LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
\ifthenelse{\equal{\LWR@frowkeywithin}{}}
{
    \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
        \LWR@frowkeyfileext}%
    \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
}%
{%
    \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
        \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
    \newfloat{#1}{\LWR@frowkeyplacement}%
    {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
    \LWR@traceinfo{finished newfloat #1}
}%
```

Rename the float if a name was given:

```
\ifthenelse{\equal{\LWR@frowkeyname}{}}
{}
{\floatname{#1}{\LWR@frowkeyname}}%
}
```

Not used:

```
\newcommand{\buildFBBOX}[2]{}
\newcommand*{\CenterFloatBoxes}{}
\newcommand*{\TopFloatBoxes}{}
```

```
\newcommand*{\BottomFloatBoxes}{}
\newcommand*{\PlainFloatBoxes}{}

\newcommand{\capsubrowsettings}{}

\NewDocumentCommand{\RawFloats}{o o}{}
```

`\RawCaption` {⟨*text*⟩}

To be used inside a minipage or parbox.

```
\newcommand{\RawCaption}[1]{#1}
```

`\floatfoot` {⟨*text*⟩}

Places additional text inside a float, inside a CSS `<div>` of class `floatfoot`.

```
\NewDocumentCommand{\floatfoot}{s +m}{%
\begin{BlockClass}{floatfoot}
#2
\end{BlockClass}
}
```

Used to compute `\linewidth`.

```
\newbool{LWR@insubfloatrow}
\boolfalse{LWR@insubfloatrow}
```

Env `subfloatrow` [⟨*num_floats*⟩]

```
\newenvironment*{subfloatrow}[1][2]
{
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
\begingroup
\booltrue{LWR@insubfloatrow}
}
{
\endgroup
\endBlockClass
\boolfalse{LWR@insubfloatrow}
}
```

File 64 **lwarp-flushend.sty**

# § 145 Package flushend

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg flushend Emulated.

**for HTML output:** Discard all options for lwarp-flushend:

```
\LWR@ProvidesPackageDrop{flushend}
%    \end{ma-crocode}
%
%    \begin{macrocode}
\newcommand*{\flushend}{}
\newcommand*{\raggedend}{}
\newcommand*{\flushcolsend}{}
\newcommand*{\raggedcolsend}{}
\newcommand*{\atColsBreak}[1]{}
\newcommand*{\atColsEnd}[1]{}
\newcommand*{\showcolsendrule}{}
```

File 65 **lwarp-fncychap.sty**

# § 146 Package fncychap

(*Emulates or patches code by* Ulf A. Lindgren.)

Pkg fncychap fncychap is emulated.

**for HTML output:** Discard all options for lwarp-fncychap:

```
\LWR@ProvidesPackageDrop{fncychap}
```

```
\def\mghrulefill#1{}
\def\ChNameLowerCase{}
\def\ChNameUpperCase{}
\def\ChNameAsIs{}
\def\ChTitleLowerCase{}
\def\ChTitleUpperCase{}
\def\ChTitleAsIs{}
\newcommand{\ChRuleWidth}[1]{}
\newcommand{\ChNameVar}[1]{}
\newcommand{\ChNumVar}[1]{}
```

```
\newcommand{\ChTitleVar}[1]{}
\newcommand{\TheAlphaChapter}{}
\newcommand{\DOCH}{}
\newcommand{\DOTI}[1]{}
\newcommand{\DOTIS}[1]{}
\newlength{\mylen}
\newlength{\myhi}
\newlength{\px}
\newlength{\py}
\newlength{\pyy}
\newlength{\pxx}
\newlength{\RW}
\newcommand{\FmN}[1]{#1}
\newcommand{\FmTi}[1]{#1}
```

---

File 66 `lwarp-fnpos.sty`

## § 147 Package **fnpos**

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `fnpos` fnpos is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{fnpos}
```

```
\newcommand*{\makeFNbottom}{}
\newcommand*{\makeFNmid}{}
\newcommand*{\makeFNbelow}{}
\newcommand*{\makeFNabove}{}
```

---

File 67 `lwarp-fontenc.sty`

## § 148 Package **fontenc**

Pkg `fontenc` If using pdfLaTeX, lwarp used to require fontspec be loaded before lwarp, but now lwarp itself loads `\fontspec` with T1 encoding, which lwarp requires. fontspec is now allowed to be loaded with another encoding after lwarp.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

**for HTML output:**

```
\LWR@ProvidesPackagePass{fontenc}
```

## File 68 `lwarp-fontspec.sty`

# § 149 Package **fontspec**

Pkg `fontspec` Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{fontspec}
```

```
\LWR@loadbefore{fontspec}
```

## File 69 `lwarp-footmisc.sty`

# § 150 Package **footmisc**

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `footmisc` footmisc is emulated.

```
\LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
\newcommand{\footnotelayout}{}
\newcommand{\setfnsymbol}[1]{}
\NewDocumentCommand{\DefineFNsymbols}{s m o m}{}

\newdimen\footnotemargin
\footnotemargin1.8em\relax

\newcommand*\hangfootparskip{0.5\baselineskip}
\newcommand*\hangfootparindent{0em}%

\let\pagefootnoterule\footnoterule
\let\mpfootnoterule\footnoterule
\def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}

\providecommand*{\multiplefootnotemarker}{3sp}
\providecommand*{\multfootsep}{,}
```

Using cleveref:

```
\providecommand*{\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
\newcommand\mpfootnotemark{%
  \@ifnextchar[%
    \@xmpfootnotemark
    {%
      \stepcounter\@mpfn
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark
    }%
}
\def\@xmpfootnotemark[#1]{%
  \begingroup
    \csname c@\@mpfn\endcsname #1\relax
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotemark
}
```

File 70 **`lwarp-footnote.sty`**

## § 151 Package **footnote**

(*Emulates or patches code by* Mark Wooding.)

Pkg footnote

footnote is used with minor patches.

**for HTML output:**

```
\LWR@ProvidesPackagePass{footnote}
```

Removed print-version formatting:

```
\def\fn@startnote{%
%   \@parboxrestore%
  \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
%   \color@begingroup% *** conflicts with lwarp
}

% \let\fn@endnote\color@endgroup% *** conflicts with lwarp
\newcommand*{\fn@endnote}{%
\LWR@htmltagc{/\LWR@tagregularparagraph}%
\LWR@orignewline%
}
```

Removed print-version formatting:

```
\def\fn@startfntext{%
  \setbox\z@\vbox\bgroup%
    \fn@startnote%
    \fn@prefntext%
    \ignorespaces%
}
```

Removed print-version formatting, added closing paragraph tag:

```
\def\fn@endfntext{%
    \LWR@htmltagc{/\LWR@tagregularparagraph}%
    \LWR@orignewline%
    \fn@postfntext%
  \egroup%
  \begingroup%
    \let\@makefntext\@empty%
    \let\@finalstrut\@gobble%
    \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
    \@footnotetext{\unvbox\z@}%
  \endgroup%
}
```

These have been redefined, so re-`\let` them again:

```
\let\endfootnote\fn@endfntext
\let\endfootnotetext\endfootnote
```

---

File 71 **lwarp-footnotehyper.sty**

# § 152 Package footnotehyper

Pkg footnotehyper

footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

**for HTML output:** Discard all options for lwarp-footnotehyper:

```
\RequirePackage{footnote}
\LWR@ProvidesPackageDrop{footnotehyper}
```

---

File 72 **lwarp-framed.sty**

# § 153 Package framed

(*Emulates or patches code by* Donald Arseneau.)

Pkg framed

framed is supported and patched by lwarp.

**for HTML output:** Accept all options for lwarp-framed:

```
\LWR@ProvidesPackagePass{framed}
\RequirePackage{xcolor}% for \convertcolorspec
```

```

\renewenvironment{framed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}

\renewenvironment{oframed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}


\renewenvironment{shaded}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{framed}%
}
{\endBlockClass}

\renewenvironment{shaded*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{framed}%
}
{\endBlockClass}


\renewenvironment{leftbar}{%
\LWR@forcenewpage
    \BlockClass{framedleftbar}
  \def\FrameCommand{}%
  \MakeFramed {}
}%
 {\endMakeFramed\endBlockClass}


\renewenvironment{snugshade}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{snugframed}%
}
```

```
{\endBlockClass}

\renewenvironment{snugshade*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{snugframed}%
}
{\endBlockClass}

\let\oframed\framed
\let\endoframed\endframed


\RenewEnviron{titled-frame}[1]{%
\CustomFBox{#1}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
}
```

\CustomFBox {⟨*toptitle*⟩} {⟨*bottitle*⟩} {⟨*thicknesstop*⟩} {⟨*bottom*⟩} {⟨*left*⟩} {⟨*right*⟩} {⟨*text contents*⟩}

```
\renewcommand{\CustomFBox}[7]{%
\convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
\ifthenelse{\isempty{#1}}{}{% not empty
    \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#1}}%
    \end{BlockClass}
}% not empty

#7

\ifthenelse{\isempty{#2}}{}{% not empty
    \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
    \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#2}}%
    \end{BlockClass}
}% not empty
\end{BlockClass}
}
```

\TitleBarFrame [⟨*marker*⟩] {⟨*title*⟩} {⟨*contents*⟩}

```
\renewcommand\TitleBarFrame[3][]{
\CustomFBox
    {#2}{}%
    \fboxrule\fboxrule\fboxrule\fboxrule
    {#3}%
}
```

```
\renewcommand{\TF@Title}[1]{#1}
```

`MakeFramed` {⟨*settings*⟩}

```
\let\MakeFramed\relax
\let\endMakeFramed\relax

\NewEnviron{MakeFramed}[1]{%
\FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
}
```

`\fb@put@frame` {⟨*frame cmd no split*⟩} {⟨*frame cmd split*⟩}

```
\renewcommand*{\fb@put@frame}[2]{%
\relax%
\@tempboxa%
}
```

---

File 73 **`lwarp-ftnright.sty`**

## § 154 Package **ftnright**

Pkg `ftnright` ftnright is ignored.

**for HTML output:** Discard all options for lwarp-ftnright:

```
\LWR@ProvidesPackageDrop{ftnright}
```

---

File 74 **`lwarp-fullpage.sty`**

## § 155 Package **fullpage**

Pkg `fullpage` fullpage is ignored.

**for HTML output:** Discard all options for lwarp-fullpage:

```
\LWR@ProvidesPackageDrop{fullpage}
```

## File 75 `lwarp-fullwidth.sty`

# § 156 Package **fullwidth**

(*Emulates or patches code by* Marco Daniel.)

Pkg `fullwidth` fullwidth is emulated.

A minipage is used, of no HTML width.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{fullwidth}
```

```
\newenvironment*{fullwidth}[1][]{%
\minipagefullwidth%
\minipage{\linewidth}%
}
{%
\endminipage%
}
```

## File 76 `lwarp-geometry.sty`

# § 157 Package **geometry**

(*Emulates or patches code by* Hideo Umeki.)

Pkg `geometry` geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

**for HTML output:** Discard all options for lwarp-geometry:

```
\LWR@ProvidesPackageDrop{geometry}
```

```
\renewcommand*{\geometry}[1]{}
\renewcommand*{\newgeometry}[1]{}
\renewcommand*{\restoregeometry}{}
\renewcommand*{\savegeometry}[1]{}
\renewcommand*{\loadgeometry}[1]{}
```

File 77 **lwarp-glossaries.sty**

## § 158 Package **glossaries**

(*Emulates or patches code by* Nicola L.C. Talbot.)

Pkg glossaries

xindy is required for glossaries.

The default style=item option for glossaries conflicts with lwarp, so the style is forced to index instead.

The page number list in the printed form would become \namerefs in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

placement and TOC options

The glossaries may be placed in a numbered or unnumbered section, given a TOC entry, and placed inline or on their own HTML page:

**Numbered section, on its own HTML page:**

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

**Unnumbered section, inline with the current HTML page:**

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

**Unnumbered section, on its own HTML page:**

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

Opt IndexLanguage

The lwarp package takes an option IndexLanguage=english to set the language used by xindy. This is passed to xindy using its -L option, and is used for both index and glossary generation.

Opt lwarpmk printglossary
Opt lwarpmk htmlglossary

lwarpmk has the commands lwarpmk printglossary and lwarpmk htmlglossary to process the glossaries created by glossaries using xindy.

**for HTML output:**

```
\PassOptionsToPackage{xindy}{glossaries}
\LWR@ProvidesPackagePass{glossaries}
\setupglossaries{nonumberlist}
\setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
\renewcommand*{\@p@glossarysection}[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the TOC entry is made after the section is created:

```
      \csname\@@glossarysec\endcsname*{#2}%
    \@gls@toc{#1}{\@@glossarysec}% Moved after the previous line.
  }%
  \@@glossaryseclabel
}
```

File 78 **lwarp-graphics.sty**

# § 159 Package graphics

(*Emulates or patches code by* D. P. CARLISLE.)

Pkg graphics graphics is emulated.

**for HTML output:**

```
\LWR@ProvidesPackagePass{graphics}
```

## § 159.1 Graphics extensions

\DeclareGraphicsExtensions {⟨*list*⟩}

\AtBeginDocument allow SVG files instead of PDF:

```
\AtBeginDocument{
\DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
\DeclareGraphicsRule{.svg}{svg}{.svg}{}
\DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
}
```

Inside a lateximage, allow PDF instead of SVG:

```
\appto\LWR@restoreorigformatting{%
\DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
}
```

## § 159.2 Length conversions and graphics options

⚠ whitespace

A scaled image in LaTeX 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.

```
\AtBeginDocument{
\renewcommand*{\unitspace}{}
}
```

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyx` option.

```
\newlength{\LWR@igwidth}
\newlength{\LWR@igheight}
\newcommand*{\LWR@igwidthstyle}{}
\newcommand*{\LWR@igheightstyle}{}
\newcommand*{\LWR@igorigin}{}
\newcommand*{\LWR@igangle}{}
\newcommand*{\LWR@igxscale}{1}
\newcommand*{\LWR@igyscale}{1}
\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:

```
\define@key{igraph}{width}{%
\setlength{\LWR@igwidth}{#1}%
\ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \uselengthunit{PT}%
    \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
}
```

If an optional height was given, set an HTML style:

```
\define@key{igraph}{height}{%
\setlength{\LWR@igheight}{#1}%
\ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \uselengthunit{PT}%
    \renewcommand*{\LWR@igheightstyle}{%
    height:\rndprintlength{\LWR@igheight} %
    }%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
}
```

Handle origin key:

```
\define@key{igraph}{origin}{%
\renewcommand*{\LWR@igorigin}{#1}%
}
```

Handle angle key:

```
\define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
\define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}

```

It appears that graphicx does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
\define@key{igraph}{scale}{%
\renewcommand*{\LWR@igxscale}{#1}%
\renewcommand*{\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
\define@key{igraph}{bb}{}
\define@key{igraph}{bbllx}{}
\define@key{igraph}{bblly}{}
\define@key{igraph}{bburx}{}
\define@key{igraph}{bbury}{}
\define@key{igraph}{natwidth}{}
\define@key{igraph}{natheight}{}
\define@key{igraph}{hiresbb}{}
\define@key{igraph}{viewport}{}
\define@key{igraph}{trim}{}
\define@key{igraph}{totalheight}{}
\define@key{igraph}{keepaspectratio}{}
\define@key{igraph}{clip}{}
\define@key{igraph}{draft}{}
\define@key{igraph}{type}{}
\define@key{igraph}{ext}{}
\define@key{igraph}{read}{}
\define@key{igraph}{command}{}
```

## § 159.3 Printing HTML styles

`\LWR@rotstyle` {⟨*prefix*⟩} {⟨*degrees*⟩}

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
\newcommand*{\LWR@rotstyle}[2]{%
 #1transform:rotate(-#2deg);
}
```

\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.

```
\newcommand*{\LWR@scalestyle}[3]{%
 #1transform:scale(#2,#3);
}
```

## § 159.4 \includegraphics

Bool LWR@infloatrow Used to compute \linewidth.

```
\newbool{LWR@infloatrow}
\boolfalse{LWR@infloatrow}
```

\LWR@opacity may be set by the transparent package. For HTML it is only used for \includegraphics.

```
\def\LWR@opacity{1}
```

Used to determine the actual image size if needed:

```
\newsavebox{\LWR@imagesizebox}
```

```
\let\LWR@origGin@setfile\Gin@setfile
```

Define the new class key for the print-mode version of \includegraphics, which is enabled inside a lateximage.

```
\AtBeginDocument{
\define@key{Gin}{class}{}
}
```

\LWR@includegraphicsb * [⟨*2: options*⟩] [⟨*3: options*⟩] {⟨*4: filename*⟩}

graphics syntax is \includegraphics * [⟨*llx,lly*⟩] [⟨*urx,ury*⟩] {⟨*file*⟩}

graphicx syntax is \includegraphics [⟨*key values*⟩] {⟨*file*⟩}

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
\NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
{%
\LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
\LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
{%
    \ifbool{LWR@infloatrow}%
    {}
    {% not in a minipage or a floatrow:
        \setlength{\linewidth}{6in}%
        \setlength{\textwidth}{6in}%
        \setlength{\textheight}{9in}%
    }%
}{}%
```

```
\begingroup%
\renewcommand*{\Gin@setfile}[3]{%
\LWR@traceinfo{Gin@setfile ##3}%
\xdef\LWR@parsedfilename{##3}%
}%
\Ginclude@graphics{\detokenize\expandafter{#4}}%
\endgroup%
\filename@parse{\LWR@parsedfilename}%
```

For correct em sizing during the width and height conversions:

```
\large%
```

Reset some defaults, possibly will be changed below if options were given:

```
\setlength{\LWR@igwidth}{0pt}%
\setlength{\LWR@igheight}{0pt}%
\renewcommand*{\LWR@igwidthstyle}{}%
\renewcommand*{\LWR@igheightstyle}{}%
\renewcommand*{\LWR@igorigin}{}%
\renewcommand*{\LWR@igangle}{}%
\renewcommand*{\LWR@igxscale}{1}%
\renewcommand*{\LWR@igyscale}{1}%
\renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```
\IfValueF{#3}{%
\IfValueTF{#2}%
{\setkeys{igraph}{#2}}%
{\setkeys{igraph}{}}%
}%
```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```
\ifbool{FormatWP}{%
    \begingroup%
    \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
    \define@key{Gin}{angle}{}%
    \IfBooleanTF{#1}%
    {% starred
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*{#4}}%
            }%
        }%
    }% starred
    {% not starred
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics{#4}}%
            }%
        }%
    }% not starred
    \endgroup%
    \uselengthunit{PT}%
    \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
    \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igheightstyle}{height:\rndprintlength{\LWR@igheight}}%
}{}%
```

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.

```
\LWR@traceinfo{LWR@includegraphicsb: about to create href}%
\href{\LWR@parsedfilename}%
{% start of href
\LWR@traceinfo{LWR@includegraphicsb: about to LWR@htmltag}%
\LWR@htmltag{% start of image tags
img src="\LWR@parsedfilename" \LWR@orignewline%
\LWR@origtilde{} alt="[\filename@base]" \LWR@orignewline%
```

Only include a style tag if a width, height, angle, or scale was given:

```
\ifthenelse{
    \NOT\equal{\LWR@igwidthstyle}{} \OR
    \NOT\equal{\LWR@igheightstyle}{} \OR
    \NOT\equal{\LWR@igorigin}{} \OR
    \NOT\equal{\LWR@igangle}{} \OR
    \NOT\equal{\LWR@igxscale}{1} \OR
    \NOT\equal{\LWR@igyscale}{1}
}%
{\LWR@origtilde{} style="%
\ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
{\LWR@igwidthstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
{\LWR@igheightstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
{\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
\ifthenelse{\NOT\equal{\LWR@igangle}{}}%
{%
\LWR@rotstyle{-ms-}{\LWR@igangle}%
\LWR@rotstyle{-webkit-}{\LWR@igangle}%
\LWR@rotstyle{}{\LWR@igangle%
}}{}%
\ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
\NOT\equal{\LWR@igyscale}{1}}%
{\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
\LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
\LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}{}%
%
\ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
{opacity:\LWR@opacity;}%
{}%
%
" \LWR@orignewline}{}%
```

Set the class:

```
\LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
```

```
}% end of image tags
}% end of href
\endgroup
```

Return to small-sized output:

```
\LWR@origscriptsize
\LWR@traceinfo{LWR@includegraphicsb done}%
}
```

\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.

```
\AtBeginDocument{

\LWR@traceinfo{Patching includegraphics.}

\LetLtxMacro\LWR@origincludegraphics\includegraphics

\renewcommand*{\includegraphics}
{%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
\LWR@traceinfo{includegraphics}
\LWR@ensuredoingapar%
\begingroup%
\catcode`\_=12
\LWR@includegraphicsb%
}% includegraphics
}% AtBeginDocument
```

## § 159.5 Boxes

\LWR@rotboxorigin Holds the origin key letters.

```
\newcommand*{\LWR@rotboxorigin}{}
```

\LWR@originname {⟨*letter*⟩}

Given one LaTeX origin key value, translate into an HTML origin word:

```
\newcommand*{\LWR@originname}[1]{%
\ifthenelse{\equal{#1}{t}}{top}{}%
\ifthenelse{\equal{#1}{b}}{bottom}{}%
\ifthenelse{\equal{#1}{c}}{center}{}%
\ifthenelse{\equal{#1}{l}}{left}{}%
\ifthenelse{\equal{#1}{r}}{right}{}%
}
```

`\LWR@originnames` {⟨*letters*⟩}

Given one- or two-letter LaTeX origin key values, translate into HTML origin words:

```
\newcommand*{\LWR@originnames}[1]{%
\StrChar{#1}{1}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
\StrChar{#1}{2}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
}
```

Handle the origin key for `\rotatebox`:

```
\define@key{krotbox}{origin}{%
\renewcommand*{\LWR@rotboxorigin}{#1}%
}
```

These keys are ignored:

```
\define@key{krotbox}{x}{}
\define@key{krotbox}{y}{}
\define@key{krotbox}{units}{}
```

`\rotatebox` [⟨*keyval list*⟩] {⟨*angle*⟩} {⟨*text*⟩}

```
\LetLtxMacro\LWR@origrotatebox\rotatebox

\AtBeginDocument{
\RenewDocumentCommand{\rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
\renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
\setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
\ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
{transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
\LWR@rotstyle{-ms-}{#2} %
\LWR@rotstyle{-webkit-}{#2} %
\LWR@rotstyle{}{#2} %
"{}}\LWR@orignewline%
```

Print the text to be rotated:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
}% AtBeginDocument
```

`\scalebox` {⟨*h-scale*⟩} [⟨*v-scale*⟩] {⟨*text*⟩}

```
\LetLtxMacro\LWR@origscalebox\scalebox

\AtBeginDocument{
\RenewDocumentCommand{\scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
\LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
\LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
\LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
"{}}%
```

Print the text to be scaled:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
}% AtBeginDocument
```

\reflectbox {⟨*text*⟩}

```
\let\LWR@origreflectbox\reflectbox

\AtBeginDocument{
\renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
}
```

\resizebox {⟨*h-length*⟩} {⟨*v-length*⟩} {⟨*text*⟩}

Simply prints its text argument.

```
\LetLtxMacro\LWR@origresizebox\resizebox

\AtBeginDocument{
\renewcommand{\resizebox}[3]{#3}
}
```

---

File 79 **lwarp-graphicx.sty**

## § 160 Package **graphicx**

Pkg graphicx graphicx is emulated.

graphicx loads graphics, which also loads lwarp-graphics, which remembers the original graphics definitions for use inside a lateximage, and then patches them \AtBeginDocument for HTML output.

lwarp-graphics handles the syntax of either graphics or graphicx.

**for HTML output:**

```
\LWR@ProvidesPackagePass{graphicx}
```

---

File 80 **lwarp-grffile.sty**

## § 161 Package **grffile**

Pkg grffile grffile is supported as-is. File types known to the browser are displayed, and unknown

⚠ matching PDF and SVG

file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

lwarp-grffile now exists as a placeholder since grffile used to be emulated by lwarp, and thus older versions of lwarp-grffile may exist and should be overwritten by this newer version.

**for HTML output:**

```
\LWR@ProvidesPackagePass{grffile}
```

File 81 **lwarp-hang.sty**

## § 162 Package **hang**

Pkg hang

hang is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{hang}
```

```
\newlength{\hangingindent}
\setlength{\hangingindent}{1em}
\newlength{\hangingleftmargin}
\setlength{\hangingleftmargin}{0em}

\newcommand*{\LWR@findhangingleftmargin}{%
\setlength{\LWR@templengthone}{\hangingleftmargin}%
\addtolength{\LWR@templengthone}{\hangingindent}%
}

\newenvironment{hangingpar}
{
    \LWR@findhangingleftmargin%
    \uselengthunit{PT}%
    \BlockClass[%
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}%
    ]%
    {hangingpar}%
}
{\endBlockClass}

\newenvironment{hanginglist}
{%
    \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
    \renewcommand*{\LWR@printopenlist}{%
        \LWR@findhangingleftmargin%
        \uselengthunit{PT}%
        ul style="list-style-type:none; %
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}"%
    }%
    \let\item\LWR@itemizeitem%
    \list{}{}%
}
{\endlist}

\newenvironment{compacthang}
{\hanginglist}
{\endhanginglist}

\newlength{\labeledleftmargin}
\setlength{\labeledleftmargin}{0em}

\newenvironment{labeledpar}[2]
{%
    \BlockClass[%
        \LWR@findhangingleftmargin%
        \uselengthunit{PT}%
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}%
    ]{labeledpar}#2%
}
{\endBlockClass}

\newenvironment{labeledlist}[1]
{\hanginglist}
{\endhanginglist}

\newenvironment{compactlabel}[1]
{\hanginglist}
{\endhanginglist}
```

File 82 **lwarp-hyperref.sty**

## § 163 Package **hyperref**

(*Emulates or patches code by* Sebastian Rahtz, Heiko Oberdiek.)

Pkg hyperref hyperref is emulated.

**for HTML output:**

```
% \LWR@ProvidesPackageDrop{hyperref}
\typeout{Using the lwarp html version of package ‘hyperref’ -- discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘hyperref’.}
% \ProvidesPackage{lwarp-#1-#2}
\DeclareOption*{}
```

```
\ProcessOptions\relax
```

```
\newcommand*{\hypersetup}[1]{}
\newcommand*{\hyperbaseurl}[1]{}
```

\hyperimage {⟨*url*⟩} {⟨*alt text*⟩}

Insert an image with alt text:

```
\NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
\LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{}}%
\endgroup%
\LWR@ensuredoingapar%
}

\DeclareRobustCommand*{\hyperimage}{%
\LWR@ensuredoingapar%
\begingroup\catcode`\_=12
\LWR@hyperimageb%
}
```

\hyperdef {⟨*1: category*⟩} {⟨*2: name*⟩} {⟨*3: text*⟩}

Creates an HTML anchor to `category.name` with the given text.

```
\NewDocumentCommand{\hyperdef}{m m +m}{%
\LWR@ensuredoingapar%
\LWR@subsublabel{#1.#2}%
#3%
}
```

\LWR@hyperrefb {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩}

Creates an HTML link to `URL#category.name` with the given text.

```
\NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
\LWR@htmltag{a href="#1\LWR@hashmark#2.#3"}%
#4%
\LWR@htmltag{/a}%
\endgroup%
}
```

\LWR@hyperrefc [⟨*label*⟩] {⟨*text*⟩}

Creates text as an HTML link to the LaTeX label.

```
\NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
\LWR@startref{#1}%
#2%
\LWR@htmltag{/a}%
\endgroup%
}
```

\hyperref {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩} — or —
[⟨*1: label*⟩] {⟨*2: text*⟩}

```
\DeclareRobustCommand*{\hyperref}{%
\LWR@ensuredoingapar%
\begingroup\catcode`\_=12
\@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
}
```

\hypertarget {⟨*name*⟩} {⟨*text*⟩}

Creates an anchor to name with the given text.

```
\NewDocumentCommand{\hypertarget}{m +m}{%
\label{#1}%
#2%
}
```

\hyperlink {⟨*name*⟩} {⟨*text*⟩}

Creates a link to the anchor created by hypertarget, with the given link text.

```
\NewDocumentCommand{\hyperlink}{m +m}{%
\hyperref[#1]{#2}%
}
```

\autoref * {⟨*label*⟩}

For HTML, \cleveref is used instead.

```
\NewDocumentCommand{\autoref}{s m}{%
\IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
}
```

\autopageref {⟨*label*⟩}

For HTML, \cleveref is used instead.

```
\NewDocumentCommand{\autopageref}{s m}{%
\IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
}
```

\pdfstringdef {⟨*macroname*⟩} {⟨*TEXstring*⟩}

```
\newcommand{\pdfstringdef}[2]{}
```

`\pdfbookmark` [⟨*level*⟩] {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\pdfbookmark}[3][]{}
```

`\currentpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\currentpdfbookmark}[2]{}
```

`\subpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\subpdfbookmark}[2]{}
```

`\belowpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\belowpdfbookmark}[2]{}
```

`\texorpdfstring` {⟨*TeXstring*⟩} {⟨*PDFstring*⟩}

```
\newcommand{\texorpdfstring}[2]{#2}
```

`\hypercalcbp` {⟨*dimen*⟩} From hyperref.

```
\def\hypercalcbp#1{%
\strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
}%
```

`\Acrobatmenu` {⟨*menuoption*⟩} {⟨*text*⟩}

```
\newcommand{\Acrobatmenu}[2]{}
```

`\TextField` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\TextField}[2][]{}
```

`\CheckBox` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\CheckBox}[2][]{}
```

`\ChoiceMenu` [⟨*parameters*⟩] {⟨*label*⟩} {⟨*choices*⟩}

```
\DeclareRobustCommand{\ChoiceMenu}[3][]{}
```

`\PushButton` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\PushButton}[2][]{}
```

`\Submit` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Submit}[2][]{}
```

`\Reset` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Reset}[2][]{}
```

`\Gauge` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Gauge}[2][]{}
```

`\LayoutTextField` {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutTextField}[2]{}
```

`\LayoutChoiceField` {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutChoiceField}[2]{}
```

`\LayoutCheckField` {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutCheckField}[2]{}
```

`\MakeRadioField` {⟨*width*⟩} {⟨*height*⟩}

```
\newcommand*{\MakeRadioField}[2]{}
```

`\MakeCheckField` {⟨*width*⟩} {⟨*height*⟩}

```
\newcommand*{\MakeCheckField}[2]{}
```

`\MakeTextField` {⟨*width*⟩} {⟨*height*⟩}

```
\newcommand*{\MakeTextField}[2]{}
```

`\MakeChoiceField` {⟨*width*⟩} {⟨*height*⟩}

```
\newcommand*{\MakeChoiceField}[2]{}
```

`\MakeFieldButton` {⟨*text*⟩}

```
\newcommand{\MakeFieldButton}[1]{}
```

File 83 **lwarp-hyperxmp.sty**

## § 164 Package **hyperxmp**

Pkg hyperxmp Emulated.

**for HTML output:** Discard all options for lwarp-hyperxmp:

```
\LWR@ProvidesPackageDrop{hyperxmp}
```

File 84 **lwarp-idxlayout.sty**

## § 165 Package **idxlayout**

(*Emulates or patches code by* Thomas Titz.)

Pkg idxlayout Emulated.

**for HTML output:** Discard all options for lwarp-idxlayout:

```
\LWR@ProvidesPackageDrop{idxlayout}
```

```
\newcommand{\LWR@indexprenote}{}

\renewcommand*{\printindex}
{
\LWR@startpars

\LWR@indexprenote

\LWR@origprintindex
}

\newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
\newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{}}

\newcommand{\idxlayout}[1]{}
\newcommand*{\indexfont}{}
\newcommand*{\indexjustific}{}
\newcommand*{\indexsubsdelim}{}
\newcommand*{\indexstheadcase}{}
```

## File 85 `lwarp-ifoddpage.sty`

# § 166 Package ifoddpage

(*Emulates or patches code by* Martin Scharrer.)

Pkg `ifoddpage`

ifoddpage is emulated.

**for HTML output:** Discard all options for lwarp-ifoddpage:

```
\LWR@ProvidesPackageDrop{ifoddpage}
```

```
\newif\ifoddpage

\newif\ifoddpageoroneside

\DeclareRobustCommand{\checkoddpage}{\oddpagetrue\oddpageoronesidetrue}

\def\oddpage@page{1}

\def\@ifoddpage{%
        \expandafter\@firstoftwo
}

\def\@ifoddpageoroneside{%
        \expandafter\@firstoftwo
}
```

## File 86 `lwarp-indentfirst.sty`

# § 167 Package indentfirst

Pkg `indentfirst`

indentfirst is ignored.

Discard all options for lwarp-indentfirst:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{indentfirst}
```

File 87 **lwarp-inputenc.sty**

## § 168 Package **inputenc**

Pkg inputenc Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{inputenc}
```

```
\LWR@loadbefore{inputenc}
```

File 88 **lwarp-keyfloat.sty**

## § 169 Package **keyfloat**

(*Emulates or patches code by* BRIAN DUNN.)

Pkg keyfloat keyfloat is supported with minor adjustments.

⚠ keywrap If placing a \keyfig[H] inside a keywrap, use an absolute width for \keyfig, instead of lw-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the lw option.)

**for HTML output:**

```
\LWR@ProvidesPackagePass{keyfloat}
```

After keyfloat has loaded:

```
\AtBeginDocument{
```

```
\let\KFLT@boxinner\relax
\let\endKFLT@boxinner\relax

\NewEnviron{KFLT@boxinner}
{%
\LWR@traceinfo{KFLT@boxinner}%
\LWR@stoppars%
\KFLT@frame{\BODY}%
\LWR@startpars%
\LWR@traceinfo{KFLT@boxinner: done}%
}
```

```
\DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
{%
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}
```

```
\DeclareDocumentEnvironment{marginfigure}{o}
  {\begin{KFLT@marginfloat}{figure}}
  {\end{KFLT@marginfloat}}

\DeclareDocumentEnvironment{margintable}{o}
  {\begin{KFLT@marginfloat}{table}}
  {\end{KFLT@marginfloat}}
```

```
\DeclareDocumentEnvironment{keywrap}{m +m}
{%
\LWR@ensuredoingapar%
\setlength{\LWR@templengthone}{#1}%
\uselengthunit{PT}%
\begin{LWR@BlockClassWP}{%
    float:right; width:\rndprintlength{\LWR@templengthone}; %
    margin:10pt%
}%
{%
    width:\rndprintlength{\LWR@templengthone}%
}%
{marginblock}%
\setlength{\linewidth}{.95\LWR@templengthone}%
#2%
\end{LWR@BlockClassWP}%
}
{%
}
```

```
}% AtBeginDocument
```

File 89 **lwarp-layout.sty**

## § 170 Package **layout**

(*Emulates or patches code by* Kent McPherson, Johannes Braams, Hideo Umeki.)

Pkg `layout` layout is emulated.

**for HTML output:** Discard all options for lwarp-layout:

```
\LWR@ProvidesPackageDrop{layout}
```

```
\NewDocumentCommand{\layout}{s}{}
```

## File 90 `lwarp-letterspace.sty`

# § 171 Package **letterspace**

(*Emulates or patches code by* R SCHLICHT.)

Pkg `letterspace` letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

**for HTML output:** Discard all options for lwarp-letterspace:

```
\LWR@ProvidesPackageDrop{letterspace}
```

```
\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
```

## File 91 `lwarp-lettrine.sty`

# § 172 Package **lettrine**

(*Emulates or patches code by* DANIEL FLIPO.)

Pkg `lettrine` Emulated.

**for HTML output:** Discard all options for lwarp-lettrine:

```
\LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `<span>` of class `lettrine`, and the following text is in a `<span>` of class `lettrinetext`. `\lettrine [`⟨*keys*⟩`] {`⟨*letter*⟩`} {`⟨*additional text*⟩`}`

```
\DeclareDocumentCommand{\lettrine}{o m m}{%
\InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
}

\newcounter{DefaultLines}
```

```
\setcounter{DefaultLines}{2}
\newcounter{DefaultDepth}
\newcommand*{\DefaultOptionsFile}{\relax}
\newcommand*{\DefaultLoversize}{0}
\newcommand*{\DefaultLraise}{0}
\newcommand*{\DefaultLhang}{0}
\newdimen\DefaultFindent
\setlength{\DefaultFindent}{\z@}
\newdimen\DefaultNindent
\setlength{\DefaultNindent}{0.5em}
\newdimen\DefaultSlope
\setlength{\DefaultSlope}{\z@}
\newdimen\DiscardVskip
\setlength{\DiscardVskip}{0.2\p@}
\newif\ifLettrineImage
\newif\ifLettrineOnGrid
\newif\ifLettrineRealHeight

\newcommand*{\LettrineTextFont}{\scshape}
\newcommand*{\LettrineFontHook}{}
\newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
\newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

File 92 **lwarp-lips.sty**

# § 173 Package **lips**

(*Emulates or patches code by* Matt Swift.)

Pkg lips

lips is emulated.

```
% \LWR@ProvidesPackageDrop{lips}
\PackageInfo{lwarp}{Using the lwarp version of package ‘lips’.}%
\ProvidesPackage{lwarp-lips}

\NewDocumentCommand{\Lips}{}{\textellipsis}

\NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}

\let\lips\Lips
\let\olips\lips

\DeclareOption*{}
\DeclareOption{mla}{
\let\lips\BracketedLips
}
\ProcessOptions\relax
```

```

\newcommand \LPNobreakList {}
```

File 93 **lwarp-listings.sty**

## § 174 Package **listings**

(*Emulates or patches code by* Carsten Heinz, Brooks Moses, Jobst Hoffmann.)

Pkg listings

listings is supported with some limitations. Text formatting is not yet supported.

**for HTML output:**

```
\begin{warpHTML}
```

```
\LWR@ProvidesPackagePass{listings}
```

Patches to embed listings inside pre tags:

```
\let\LWR@origlst@Init\lst@Init
\let\LWR@origlst@DeInit\lst@DeInit

\let\LWR@origlsthkEveryPar\lsthk@EveryPar

\renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{lol}{#1}{#2}}
```

\lst@Init {⟨*backslash-processing*⟩} Done at the start of a listing.

```
\renewcommand{\lst@Init}[1]{%
```

First, perform the listings initialization:

```
\LWR@traceinfo{lst@Init}%
\renewcommand*{\@captype}{lstlisting}%
\LWR@origlst@Init{#1}%
\LWR@traceinfo{finished origlst@Init}%
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```
\LWR@traceinfo{About to create verbatim.}%
\let\lsthk@EveryPar\relax%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}%

\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so open a <span>:

```
\ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
\fi%
}
```

\lst@DeInit Done at the end of a listing.

```
\renewcommand*{\lst@DeInit}{%
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```
\let\lsthk@EveryPar\relax%

\LWR@afterendverbatim%
\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so create the closing </span>:

```
\ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
\fi%
```

Final listings deinit:

```
\LWR@origlst@DeInit%
}
```

\lst@MakeCaption {⟨*t/b*⟩}

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
\def\lst@MakeCaption#1{%
\LWR@traceinfo{MAKING CAPTION at #1}%
  \lst@ifdisplaystyle
\LWR@traceinfo{making a listings display caption}%
    \ifx #1t%
         \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
                                 \expandafter\refstepcounter
         \fi {lstlisting}%
\LWR@traceinfo{About to assign label: !\lst@label!}%
%         \ifx\lst@label\@empty\else
% \label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
        \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
        \global\let\lst@name\lst@arg \global\let\lstname\lst@name
        \lst@ifnolol\else
            \ifx\lst@@caption\@empty
                \ifx\lst@caption\@empty
```

```
                    \ifx\lst@intname\@empty \else \def\lst@temp{ }%
                    \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
\LWR@traceinfo{addcontents lst@name: -\lst@name-}%
%                            \addcontentsline{lol}{lstlisting}{\lst@name}
                    \fi\fi
                \fi
            \else
```

This would have to be modified for lwarp:

```
\LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
                 \addcontentsline{lol}{lstlisting}%
{\protect\numberline{\thelstlisting}%
{\protect\ignorespaces \lst@@caption \protect\relax}}%
            \fi
         \fi
     \fi
    \ifx\lst@caption\@empty\else
\LWR@traceinfo{lst@caption not empty-}%
        \lst@IfSubstring #1\lst@captionpos
            {\begingroup
\LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
%              \let\@@vskip\vskip
%              \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
%              \def\lst@vskip{\nobreak\@@vskip\@tempskipa\nobreak}%
%              \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
%              \ifx #1t\allowbreak \fi
             \ifx\lst@title\@empty
```

New lwarp code to create a caption:

```
                  \lst@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}
             \else
```

New lwarp code to create a title:

```
%                  \lst@maketitle\lst@title % (AS)
\LWR@traceinfo{Making title: \lst@title}%
\begin{BlockClass}{lstlistingtitle}% lwarp
\lst@maketitle\lst@title% lwarp
\end{BlockClass}% lwarp
             \fi
\LWR@traceinfo{About to assign label: !\lst@label!}%
        \ifx\lst@label\@empty\else
\leavevmode% gets rid of bad space factor error
\GetTitleStringExpand{\lst@caption}%
\edef\LWR@lntemp{\GetTitleStringResult}%
```

```
\edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
\label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
```

Not needed for lwarp:

```
%              \ifx #1b\allowbreak \fi
             \endgroup}{}%
    \fi
\LWR@traceinfo{end of making a listings display caption}%
  \else
\LWR@traceinfo{INLINE}%
  \fi
\LWR@traceinfo{DONE WITH CAPTION at #1}%
}
```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field `\VerbatimHTMLWidth` wide.

```
\lst@Key{numbers}{none}{%
    \let\lst@PlaceNumber\@empty
    \lstKV@SwitchCases{#1}%
    {none&\\%
     left&\def\lst@PlaceNumber{%
% \llap{
\LWR@orignormalfont%
\lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
% }
}
\\%
     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
                \kern\VerbatimHTMLWidth \kern\lst@numbersep
                \lst@numberstyle{\thelstnumber}}}%
    }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
```

```
\end{warpHTML}
```

File 94 **`lwarp-longtable.sty`**

# § 175 Package longtable

(*Emulates or patches code by* David Carlisle.)

Pkg `longtable`

longtable is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{longtable}
```

⚠ Longtable `\endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid “Misplaced `\noalign`.” when using `\begin`{warpprint}.

Keep the `\endfirsthead` row, which is still relevent to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside `\warpingprintonly`.

See:
http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable

Env `longtable` * [⟨*horizalignment*⟩] {⟨*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.

```
\newenvironment{longtable*}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\booltrue{LWR@starredlongtable}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

\newenvironment{longtable}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\refstepcounter{\LTcaptype}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

```

Provided for compatibility, but ignored:

```
\newcounter{LTchunksize}
\def\endhead{\LWR@tabularendofline}% throws away options //[dim] and //*
\def\endfirsthead{\LWR@tabularendofline}
\def\endfoot{\LWR@tabularendofline}
\def\endlastfoot{\LWR@tabularendofline}
\newcommand\tabularnewline{\LWR@tabularendofline}
\newcommand{\setlongtables}{}% Obsolete command, does nothing.
\newlength{\LTleft}
\newlength{\LTright}
\newlength{\LTpre}
\newlength{\LTpost}
\newlength{\LTcapwidth}
```

```
\renewcommand*{\kill}{\LWR@tabularendofline}
```

File 95 **lwarp-lscape.sty**

## § 176 Package lscape

(*Emulates or patches code by* D. P. Carlisle.)

Pkg lscape

lscape is emulated.

**for HTML output:** Discard all options for lwarp-lscape.

```
\LWR@ProvidesPackageDrop{lscape}
```

```
\newenvironment*{landscape}{}{}
```

File 96 **lwarp-ltcaption.sty**

## § 177 Package ltcaption

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg ltcaption

ltcaption is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{ltcaption}
```

`\LTcaptype` is already defined by lwarp.

`longtable*` is already defined by lwarp-longtable.

```
\newlength{\LTcapskip}
\newlength{\LTcapleft}
\newlength{\LTcapright}
\newcommand*{\LTcapmarginsfalse}{}
```

---

File 97 **`lwarp-ltxtable.sty`**

# § 178 Package **ltxtable**

Pkg `ltxtable` ltxtable is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{ltxtable}
```

`\LTXtable` {⟨*width*⟩} {⟨*file*⟩}

```
\newcommand*{\LTXtable}[2]{%
\input{#2}%
}
```

---

File 98 **`lwarp-luatodonotes.sty`**

# § 179 Package **luatodonotes**

(*Emulates or patches code by* Fabian Lipp.)

Pkg `luatodonotes` luatodonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

**for HTML output:**

```
\LWR@ProvidesPackagePass{luatodonotes}
```

Nullify options:

```
\@todonotes@additionalMarginEnabledfalse
```

```
\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}


\renewcommand{\@todonotes@drawMarginNoteWithLine}{%
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
        {\@todonotes@author:\,}%
        \fi%
        \@todonotes@text%
    }%
}

\newcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

\LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon

\RenewDocumentCommand{\@todocommon}{m m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\LWRTODONOTES@orig@todocommon{#1}{#2}%
\endgroup%
}

\renewcommand{\@todoarea}[3][]{%
    \@todonotes@areaselectedtrue%
    \@todocommon{#1}{#2}%
    \todonotes@textmark@highlight{#3}%
    \zref@label{@todonotes@\arabic{@todonotes@numberoftodonotes}@end}%
}%


\DeclareDocumentCommand{\todonotes@textmark@highlight}{m}{%
\InlineClass[background:\#B3FFB3]{highlight}{#1}%
}

\fi% \if@todonotes@disabled
```

## File 99 lwarp-marginfit.sty

## § 180 Package marginfit

Pkg marginfit marginfit is ignored.

**for HTML output:** Discard all options for lwarp-marginfit:

```
\LWR@ProvidesPackageDrop{marginfit}
```

## File 100 lwarp-marginfix.sty

## § 181 Package marginfix

(*Emulates or patches code by* Stephen Hicks.)

Pkg marginfix Emulated.

**for HTML output:** Discard all options for lwarp-marginfix:

```
\LWR@ProvidesPackageDrop{marginfix}
```

```
\newcommand*{\marginskip}[1]{}
\newcommand*{\clearmargin}{}
\newcommand*{\softclearmargin}{}
\newcommand*{\extendmargin}[1]{}
\newcommand*{\mparshift}[1]{}
\newdimen\marginheightadjustment
\newdimen\marginposadjustment
\newcommand*{\blockmargin}[1][]{}
\newcommand*{\unblockmargin}[1][]{}
\newcommand*{\marginphantom}[2][]{}
```

File 101 `lwarp-marginnote.sty`

## § 182 Package **marginnote**

(*Emulates or patches code by* Markus Kohm.)

Pkg `marginnote` Emulated.

**for HTML output:** Discard all options for lwarp-marginnote:

```
\LWR@ProvidesPackageDrop{marginnote}
```

```
\NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
\newcommand*{\marginnoteleftadjust}{}
\newcommand*{\marginnoterightadjust}{}
\newcommand*{\marginnotetextwidth}{}
\let\marginnotetextwidth\textwidth
\newcommand*{\marginnotevadjust}{}
\newcommand*{\marginfont}{}
\newcommand*{\raggedleftmarginnote}{}
\newcommand*{\raggedrightmarginnote}{}
```

File 102 `lwarp-mcaption.sty`

## § 183 Package **mcaption**

(*Emulates or patches code by* Stephan Hennig.)

Pkg `mcaption` mcaption is nullified.

**for HTML output:** Discard all options for lwarp-mcaption:

```
\LWR@ProvidesPackageDrop{mcaption}
```

```
\newenvironment{margincap}{}{}
\newcommand*{\margincapalign}{}
\newlength{\margincapsep}
```

---

File 103 **lwarp-mdframed.sty**

# § 184 Package **mdframed**

(*Emulates or patches code by* Marco Daniel, Elke Schubert.)

Pkg mdframed mdframed is loaded with options forced to `framemethod=none`.

## § 184.1 Package loading

**for HTML output:**

```
\RequirePackage{xcolor}% for \convertcolorspec
\LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before mdframed

```
\LWR@origRequirePackage{amsthm}
```

Do not require Ti*k*z or pstricks:

```
\LWR@origRequirePackage[framemethod=none]{mdframed}
```

## § 184.2 Limitations

support Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for mdframed environments and frame titles.

⚠ loading When used, lwarp loads mdframed in HTML with `framemethod=none`.

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where \textbf must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarp does not support \bfseries and friends, only one font selection may be made at a time.

theoremtitlefont theoremtitlefont is not supported, since the following text is not in braces in the mdframed source.

footnotes Footnotes are currently placed at the bottom of the HTML page.

ignored options userdefinedwidth and align are currently ignored.

CSS classes Environments created or encapsulated by mdframed are enclosed in a <div> of class md<environmentname>, or mdframed otherwise.

Frame titles are placed into a <span> of class mdframedtitle. Subtitles are in a <span> of class mdframedsubtitle, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

To handle CSS and paragraphs, patch code at start and end of environment and contents. \LWR@origraggedright helps avoid hyphenation.

```
\mdfsetup{
startcode={\LWR@mdframedstart\LWR@origraggedright},
endcode={\LWR@mdframedend},
startinnercode={\LWR@startpars\LWR@origraggedright},
endinnercode={\LWR@stoppars},
}
```

## § 184.3 Color and length HTML conversion

\LWR@mdfprintcolor {⟨*mdfcolorkey*⟩}

Given the mdframed key, print the color.

```
\newcommand*{\LWR@mdfprintcolor}[1]{%
\convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
\#\LWR@tempcolor
}
```

\LWR@mdfprintlength {⟨*mdflengthkey*⟩}

Given the mdframed key, print the length.

```
\newcommand*{\LWR@mdfprintlength}[1]{%
\rndprintlength{\csuse{mdf@#1@length}}
}
```

## § 184.4 Environment encapsulation

\LWR@mdframedstart Actions before an mdframe starts.

Encapsulate a frame inside a <div> of the desired class.

```
\newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
\LWR@stoppars%
```

Below, print HTML pt units:

```
\uselengthunit{PT}%
```

Open a <div> and with custom class and custom style:

```
\LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the background color:

```
background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
border: \LWR@mdfprintlength{linewidth} solid
\LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
\ifbool{mdf@shadow}{%
    box-shadow:
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintcolor{shadowcolor} ;
}
{box-shadow: none ;}
\LWR@orignewline

"}
% \LWR@htmldivclass{\LWR@mdthisenv}
```

mdframed environment may not work with the HTML versions of the following, so restore them to their originals while inside mdframed:

```
\LetLtxMacro{\hspace}{\LWR@orighspace}%
\LetLtxMacro\rule\LWR@origrule%
\LetLtxMacro\makebox\LWR@origmakebox%
}
```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
\newcommand*{\LWR@mdframedend}{
```

Close the custom `<div>`:

```
\LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
\gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
\LWR@startpars%
}
```

## § 184.5 Titles and subtitles

`\mdfframedtitleenv` {⟨*title*⟩}

Encapsulation of the original which places the title inside a `<span>` of class `mdframedtitle`:

```
\LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv

\newlength{\LWR@titleroundcorner}

\renewrobustcmd\mdfframedtitleenv[1]{%
\LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
\uselengthunit{PT}%
```

Open a `<span>` with a custom `class` and custom `style`:

```
\LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the title background color:

```
background:
\LWR@mdfprintcolor{frametitlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the title rule:

```
\ifbool{mdf@frametitlerule}{%
    border-bottom:
    \LWR@mdfprintlength{frametitlerulewidth}
    solid
    \LWR@mdfprintcolor{frametitlerulecolor}
    ; \LWR@orignewline
}{}%
```

The title's top border radius is adjusted for the line width:

```
border-radius:
\setlength{\LWR@titleroundcorner}
    {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
    \rndprintlength{\LWR@titleroundcorner}
    \rndprintlength{\LWR@titleroundcorner}
    0pt 0pt
    \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
" \LWR@orignewline
}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Print the title inside the span:

```
#1%
```

Closee the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
}
```

`\LWR@mdfsubtitlecommon` {⟨*sub -or- subsub*⟩} [⟨*options*⟩] {⟨*title*⟩}

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `<span>` of class `mdframedsubtitle`:

```
\NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
{% the following empty line is required

```

Special handling for mdframed: Subtitles have `\par`s around them, so temporarily disable them here.

```
\let\par\LWR@origpar%
```

Open a `<span>` with a custom `class` and custom `style`:

```
\LWR@htmltagc{span class="mdframed#1title"
style=" \LWR@orignewline
```

Convert and print the background color:

```
background:
\LWR@mdfprintcolor{#1titlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the above line:

```
\ifbool{mdf@#1titleaboveline}{%
```

```
    border-top:
    \LWR@mdfprintlength{#1titleabovelinewidth}
    solid
    \LWR@mdfprintcolor{#1titleabovelinecolor}
    ; \LWR@orignewline
}{}%
```

Convert and print the below line:

```
\ifbool{mdf@#1titlebelowline}{%
    border-bottom:
    \LWR@mdfprintlength{#1titlebelowlinewidth}
    solid
    \LWR@mdfprintcolor{#1titlebelowlinecolor}
    ; \LWR@orignewline
}{}%
```

Finish the custom style and the opening span tag:

```
"}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
\IfNoValueTF{#2}
{\csuse{LWR@origmdf#1title}{#3}}%
{\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}% the following empty line is required
\end{LWR@nestspan}% must follow the /span or an extra <p> appears

}
```

`\LWR@mdfsubtitle` [⟨*options*⟩] {⟨*title*⟩}

```
\let\LWR@origmdfsubtitle\mdfsubtitle

\newcommand*{\LWR@mdfsubtitle}{%
\LWR@mdfsubtitlecommon{sub}%
}
\let\mdfsubtitle\LWR@mdfsubtitle
```

`\LWR@mdfsubsubtitle` [⟨*options*⟩] {⟨*title*⟩}

```
\let\LWR@origmdfsubsubtitle\mdfsubsubtitle

\newcommand*{\LWR@mdfsubsubtitle}{%
\LWR@mdfsubtitlecommon{subsub}%
}
\let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

## § 184.6 New environments

\LWR@mdthisenv Stores the environment of the frame about to be created:

```
\newcommand*{\LWR@mdthisenv}{mdframed}
```

\newmdenv [⟨*options*⟩] {⟨*env-name*⟩}

Modified from the original to remember the environment.

```
\renewrobustcmd*\newmdenv[2][]{%
\newenvironment{#2}%
{%
\mdfsetup{#1}%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}%
}
{\end{mdframed}}%
}
```

\surroundwithmdframed [⟨*options*⟩] {⟨*environment*⟩}

Modified from the original to remember the environment.

```
\renewrobustcmd*{\surroundwithmdframed}[2][]{%
\BeforeBeginEnvironment{#2}{%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{\end{mdframed}}%
}
```

\mdtheorem [⟨*mdframed-options*⟩] envname [⟨*numberedlike*⟩] {⟨*caption*⟩} [⟨*within*⟩]

Modified from the original to remember the environment.

```
\DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
 {\ifcsdef{#2}%
   {\mdf@PackageWarning{Environment #2 already exits\MessageBreak}}%
   {%
    \IfNoValueTF {#3}%
     {%#3 not given -- number relationship
      \IfNoValueTF {#5}%
        {%#3+#5 not given
        \@definecounter{#2}%
        \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#2}%
          \ifstrempty{##1}%
```

```
           {\let\@temptitle\relax}%
           {%
            \def\@temptitle{\mdf@theoremseparator%
                            \mdf@theoremspace%
                            \mdf@theoremtitlefont%
                            ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }%
         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
         {\end{mdframed}}%
       \newenvironment{#2*}[1][]{%
         \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
         {\end{mdframed}}%
       }%
       {%#5 given -- reset counter
       \@definecounter{#2}\@newctr{#2}[#5]%
       \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
       \expandafter\xdef\csname the#2\endcsname{%
             \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
                \@thmcounter{#2}}%
       \newenvironment{#2}[1][]{%
         \refstepcounter{#2}%
         \ifstrempty{##1}%
           {\let\@temptitle\relax}%
           {%
            \def\@temptitle{\mdf@theoremseparator%
                            \mdf@theoremspace%
                            \mdf@theoremtitlefont%
                            ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }
         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
         {\end{mdframed}}%
       \newenvironment{#2*}[1][]{%
         \ifstrempty{##1}%
           {\let\@temptitle\relax}%
           {%
            \def\@temptitle{\mdf@theoremseparator%
                            \mdf@theoremspace%
                            \mdf@theoremtitlefont%
                            ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }%
         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
         {\end{mdframed}}%
       }%
    }%
```

```
     {%#3 given -- number relationship
        \global\@namedef{the#2}{\@nameuse{the#3}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#3}%
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                             \mdf@theoremspace%
                             \mdf@theoremtitlefont%
                             ##1}%
             \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
             }
          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
          {\end{mdframed}}%
        \newenvironment{#2*}[1][]{%
          \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
     }%
    \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
    \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
   }%
 }
```

\newmdtheoremenv [⟨*mdframed-options*⟩] envname [⟨*numberedlike*⟩] {⟨*caption*⟩} [⟨*within*⟩]

Modified from the original to remember the environment.

```
\DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
    {\newtheorem{#2}{#4}}%
    {%
     \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
     \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
    }%
\BeforeBeginEnvironment{#2}{%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{%
\end{mdframed}}%
}
```

File 104 **lwarp-metalogo.sty**

# § 185 Package **metalogo**

(*Emulates or patches code by* Andrew Gilbert Moschou.)

Pkg metalogo metalogo is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{metalogo}
```

```
\newcommand\setlogokern[2]{}
\newcommand\setlogodrop[2][XeTeX]{}
\newcommand\setLaTeXa[1]{}
\newcommand\setLaTeXee[1]{}
\newcommand\seteverylogo[1]{}
\newcommand\everylogo[1]{}
```

File 105 **lwarp-microtype.sty**

# § 186 Package **microtype**

(*Emulates or patches code by* R Schlicht.)

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

**for HTML output:** Discard all options for lwarp-microtype:

```
\LWR@ProvidesPackageDrop{microtype}
```

```
\DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
\DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
\DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
\DeclareDocumentCommand{\SetProtrusion}{o m m}{}
\DeclareDocumentCommand{\SetExpansion}{o m m}{}
\DeclareDocumentCommand{\SetTracking}{o m m}{}
\DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
\DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
\DeclareDocumentCommand{\DisableLigatures}{o m}{}
\DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
\DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
\DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
\DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
```

```
\DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
\DeclareDocumentCommand{\microtypesetup}{m}{}
\DeclareDocumentCommand{\microtypecontext}{m}{}
\DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
\DeclareDocumentCommand{\lsstyle}{}{}
\DeclareDocumentCommand{\textls}{o +m}{}
\DeclareDocumentCommand{\lslig}{m}{#1}
}
\def\DeclareMicrotypeSet#1#{\@gobbletwo}
\def\DeclareMicrotypeVariants#1#{\@gobble}
\@onlypreamble\DeclareMicrotypeSet
\@onlypreamble\UseMicrotypeSet
\@onlypreamble\DeclareMicrotypeSetDefault
\@onlypreamble\DisableLigatures
\@onlypreamble\DeclareMicrotypeVariants
\@onlypreamble\DeclareMicrotypeBabelHook
```

File 106 **lwarp-midfloat.sty**

## § 187 Package midfloat

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg midfloat

midfloat is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{midfloat}
```

```
\newenvironment{strip}[1][]{}{}
\newskip\stripsep
```

File 107 **lwarp-moreverb.sty**

## § 188 Package moreverb

(*Emulates or patches code by* Robin Fairbairns.)

Pkg moreverb

moreverb is supported with some patches.

**for HTML output:**

```
\begin{warpHTML}
```

```
\LWR@ProvidesPackagePass{moreverb}
```

```
\BeforeBeginEnvironment{verbatimtab}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{verbatimtab}{%
\LWR@afterendverbatim%
}


\LetLtxMacro\LWRMV@orig@verbatimtabinput\@verbatimtabinput

\renewcommand{\@verbatimtabinput}[2][]{%
\LWR@forcenewpage
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
\LWRMV@orig@verbatimtabinput[#1]{#2}%
\LWR@afterendverbatim%
}

\BeforeBeginEnvironment{listing}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
}

\AfterEndEnvironment{listing}{%
% \unskip\LWR@origvspace*{-\baselineskip}%
\LWR@afterendverbatim%
}

\BeforeBeginEnvironment{listingcont}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
}

\AfterEndEnvironment{listingcont}{%
% \unskip\LWR@origvspace*{-\baselineskip}%
\LWR@afterendverbatim%
}
```

```
\LetLtxMacro\LWRMV@@listinginput\@listinginput

\renewcommand{\@listinginput}[3][]{
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
\LWRMV@@listinginput[#1]{#2}{#3}
\LWR@afterendverbatim%
}


\renewenvironment*{boxedverbatim}
{
\LWR@forcenewpage
\LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
\verbatim%
}
{
\endverbatim%
\LWR@afterendverbatim%
}



\end{warpHTML}
```

---

## File 108 `lwarp-morewrites.sty`

# § 189 Package **morewrites**

Pkg `morewrites` Error if morewrites is loaded after lwarp.

Discard all options for lwarp-morewrites:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{morewrites}
```

```
\LWR@loadbefore{morewrites}
```

---

## File 109 `lwarp-mparhack.sty`

# § 190 Package **mparhack**

Pkg `mparhack` Ignored.

**for HTML output:** Discard all options for lwarp-mparhack:

```
\LWR@ProvidesPackageDrop{mparhack}
```

---

## File 110 `lwarp-multicol.sty`

# § 191 Package **multicol**

(*Emulates or patches code by* Frank Mittelbach.)

Pkg `multicol` multicol is emulated.

**for HTML output:**

```
\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`.

```
\begin{warpHTML}
```

Env `multicols` * {⟨*numcols*⟩} [⟨*heading*⟩]

```
\NewDocumentEnvironment{multicols}{s m o}
```

HTML `<div>` class to contain everything:

```
{
\LWR@forcenewpage
\BlockClass{multicols}
```

Optional HTML `<div>` class for the heading:

```
\IfValueT{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}}
```

When done with the environment, close the `<div>`:

```
{\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
\newcommand*{\columnbreak}{}
\newcommand*{\RLmulticolcolumns}{}
\newcommand*{\LRmulticolcolumns}{}

\newlength{\premulticols}
\newlength{\postmulticols}
\newlength{\multicolsep}
\newlength{\multicolbaselineskip}
\newlength{\multicoltolerance}
\newlength{\multicolpretolerance}
\newcommand*{\columnseprulecolor}{\normalcolor}
\newcounter{columnbadness}
\newcounter{finalcolumnbadness}
\newcounter{collectmore}
\newcounter{unbalance}
\newlength{\multicolovershoot}
\newlength{\multicolundershoot}
```

```
\end{warpHTML}
```

File 111 **lwarp-multirow.sty**

# § 192 Package **multirow**

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg multirow

multirow is emulated during HTML output, and used as-is while inside a `lateximage`.

In a `lateximage`, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 61.18 for the print-mode versions.

**for HTML output:**

```
\LWR@ProvidesPackagePass{multirow}
```

Remember the print-mode version:

```
\LetLtxMacro\LWR@origmultirow\multirow
```

`\LWR@multirowborder` Set to `left` or `right` to create a thick border for the cell, for use by bigdelim:

```
\newcommand{\LWR@multirowborder}{}
```

## § 192.1 Multirow

`\multirow` [⟨*vpos*⟩] {⟨*numrows*⟩} [⟨*bigstruts*⟩] {⟨*width*⟩} [⟨*fixup*⟩] {⟨*text*⟩}

```
\RenewDocumentCommand{\multirow}{O{c} m o m o +m}%
{%
\LWR@traceinfo{*** multirow #1 #2 #4}
\LWR@maybenewtablerow%
\LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
\LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
\IfValueT{#1}{%
\ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
\ifstrequal{#1}{t}{style="vertical-align:top" }{}%
}%
```

The left/right border, if given:

```
\ifdefvoid{\LWR@multirowborder}{}{%
style="border-\LWR@multirowborder: 2px dotted black ; %
padding-\LWR@multirowborder: 2px" %
}%
```

A class adds the column spec and the rule:

```
class="td%
```

Append this column's spec:

```
\StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add the vertical bar class.

```
\LWR@addcmidruletrim%
\LWR@addleftmostbartag%
\LWR@printbartag{\arabic{LWR@tablecolindex}}%
"%
```

```
\LWR@tdstartstyles%
\LWR@addcmidrulewidth%
\LWR@tdendstyles%
}%
```

The column's < spec:

```
\LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tablecolindex}}%
```

While printing the text, redefine \\ to generate a new line

```
\begingroup\LetLtxMacro{\\}{\LWR@endofline}#6\endgroup%
\LWR@stoppars%
\global\boolfalse{LWR@intabularmetadata}%
\renewcommand{\LWR@multirowborder}{}%
\LWR@traceinfo{*** multirow done}%
}%
```

## § 192.2 Combined multicolumn and multirow

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

\multicolumnrow {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

```
\NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```
\begingroup%
\LetLtxMacro{\\}{\LWR@endofline}%
\LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
\endgroup%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tablecolindex}{#1}%
\addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
\booltrue{LWR@skipatbang}%
}
```

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\multirow\LWR@origmultirow%
\renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
}
```

File 112 **lwarp-nameref.sty**

## § 193 Package **nameref**

Pkg nameref

nameref is emulated by lwarp.

**for HTML output:** Discard all options for lwarp-nameref:

```
\typeout{Using the lwarp html version of package ‘nameref’ -- discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘nameref’.}
\DeclareOption*{}
\ProcessOptions\relax
```

File 113 **lwarp-needspace.sty**

## § 194 Package **needspace**

(*Emulates or patches code by* Peter Wilson.)

Pkg needspace

needspace is not used during HTML conversion.

**for HTML output:** Discard all options for lwarp-needspace:

```
\LWR@ProvidesPackageDrop{needspace}

\newcommand*{\needspace}[1]{}
\DeclareDocumentCommand{\Needspace}{s m}{}
```

File 114 **lwarp-newclude.sty**

## § 195 Package **newclude**

Pkg newclude

Error if newclude is loaded after lwarp.

Discard all options for lwarp-newclude:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{newclude}
```

```
\LWR@loadbefore{newclude}
```

File 115 **lwarp-newunicodechar.sty**

## § 196 Package newunicodechar

Pkg newunicodechar

Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{newunicodechar}
```

```
\LWR@loadbefore{newunicodechar}
```

File 116 **lwarp-nextpage.sty**

## § 197 Package nextpage

(*Emulates or patches code by* Peter Wilson.)

Pkg nextpage

nextpage is nullified.

**for HTML output:** Discard all options for lwarp-nextpage.

```
\LWR@ProvidesPackageDrop{nextpage}
```

```
\newcommand{\cleartoevenpage}[1][]{}
\newcommand{\movetoevenpage}[1][]{}
\newcommand{\cleartooddpage}[1][]{}
\newcommand{\movetooddpage}[1][]{}
```

File 117 **lwarp-nonumonpart.sty**

## § 198 Package nonumonpart

Pkg nonumonpart

nonumonpart is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{nonumonpart}
```

File 118 **lwarp-nopageno.sty**

## § 199 Package **nopageno**

Pkg nopageno

nopageno is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{nopageno}
```

File 119 **lwarp-nowidow.sty**

## § 200 Package **nowidow**

(*Emulates or patches code by* Raphaël Pinson.)

Pkg nowidow

nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{nowidow}
```

\nowidow [⟨*lines*⟩]

\setnowidow [⟨*lines*⟩]

```
\newcommand*{\nowidow}[1][]{}
\newcommand*{\setnowidow}[1][]{}
```

\noclub [⟨*lines*⟩]

\setnoclub [⟨*lines*⟩]

```
\newcommand*{\noclub}[1][]{}
\newcommand*{\setnoclub}[1][]{}
```

File 120 **lwarp-ntheorem.sty**

## § 201 Package **ntheorem**

(*Emulates or patches code by* Wolfgang May, Andreas Schedler.)

Pkg ntheorem

ntheorem is patched for use by lwarp.

Table 13: Ntheorem package — CSS styling of theorems and proofs

**Theorem:** `<div>` of class `theorembody<theoremstyle>`

**Theorem Header:** `<span>` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

## § 201.1 Limitations

⚠ Font control This conversion is not total. Font control is via CSS, and the custom LaTeX font settings are ignored.

⚠ Equation numbering ntheorem has a bug with equation numbering in AMS environments when the option `thref` is used. lwarp does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarp's HTML output, but not with the print output. It is recommended to use cleveref instead of ntheorem's `thref` option.

## § 201.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ Disabled options The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because cleveref functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because lwarp emulated hyperref.

**for HTML output:** Some disabled options:

```
\DeclareOption{thref}{}


\newbool{LWR@ntheoremmarks}
\boolfalse{LWR@ntheoremmarks}

\DeclareOption{thmmarks}{
\booltrue{LWR@ntheoremmarks}
\newif\ifsetendmark\setendmarktrue
}


\newbool{LWR@ntheoremamsthm}
\boolfalse{LWR@ntheoremamsthm}

```

```
\DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}


\DeclareOption{amsmath}{}
\DeclareOption{hyperref}{}


\LWR@ProvidesPackagePass{ntheorem}
```

## § 201.3 Remembering the theorem style

Storage for the style being used for new theorems.

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
   \@ifundefined{th@#1}{\@warning
          {Unknown theoremstyle ‘#1’. Using ‘plain’}%
          \theorem@style{plain}
            \renewcommand{\LWR@newtheoremstyle}{plain}% new
            }%
      {
        \theorem@style{#1}
        \renewcommand{\LWR@newtheoremstyle}{#1}% new
        }
}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

\gdef\@xnthm#1#2[#3]{%
  \ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \expandafter\@ifundefined{c@#1}%
        {\@definecounter{#1}}{}%
     \@newctr{#1}[#3]%
     \expandafter\xdef\csname the#1\endcsname{%
       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
          {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi
}

\gdef\@ynthm#1#2{%
  \ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \expandafter\@ifundefined{c@#1}%
        {\@definecounter{#1}}{}%
     \expandafter\xdef\csname the#1\endcsname
        {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi
}

\gdef\@othm#1[#2]#3{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}%
   {\ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \global\@namedef{the#1}{\@nameuse{the#2}}%
     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
              \noexpand\@num@addtheoremline{#1}{#3}}%
     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
              \noexpand\@nonum@addtheoremline{#1}{#3}}%
    \theoremkeyword{#3}%
    \expandafter\protected@xdef\csname #1Keyword\endcsname
             {\the\theoremkeyword}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
                \@thm{#1}{#2}{#3}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi}
}
```

## § 201.4 HTML cross-referencing

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```
\newcommand{\LWR@inctheorem}{%
\addtocounter{LWR@thisautoid}{1}%
\LWR@stoppars%
\LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"{}}\LWR@htmltag{/a}%
\LWR@startpars%
}
```

## § 201.5 \newtheoremstyle

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader<style>.

```
\gdef\newtheoremstyle#1#2#3{%
   \expandafter\@ifundefined{th@#1}%
   {\expandafter\gdef\csname th@#1\endcsname{%
    \def\@begintheorem####1####2{%
\LWR@forcenewpage% new
\BlockClass{theorembody#1}%\LWR@thisthmstyle% new
\LWR@inctheorem% new
#2}%
    \def\@opargbegintheorem####1####2####3{%
\LWR@forcenewpage% new
\BlockClass{theorembody#1}%\LWR@thisthmstyle% new
\LWR@inctheorem% new
#3}%
}%
}%
{\PackageError{\basename}{Theorem style #1 already defined}\@eha}
}
```

## § 201.6 Standard styles

```
\renewtheoremstyle{plain}%
  {\item[\hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
  {\item[\hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}

\renewtheoremstyle{break}%
  {\item[
%  \rlap{\vbox{\hbox{
    \hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}%
  {\item[
%  \rlap{\vbox{\hbox{
    \hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderbreak}%
```

```
        {##1\ ##2\ (##3)\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}

\renewtheoremstyle{change}%
  {\item[\hskip\labelsep
    \theorem@headerfont
    \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
  {\item[\hskip\labelsep
    \theorem@headerfont
    \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{changebreak}%
  {\item[
%  \rlap{\vbox{\hbox{
        \hskip\labelsep \theorem@headerfont
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}%
  {\item[
%  \rlap{\vbox{\hbox{
        \hskip\labelsep \theorem@headerfont
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\ (##3)\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}

\renewtheoremstyle{margin}%
  {\item[\hskip\labelsep\theorem@headerfont
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\theorem@separator}
    ]}%
  {\item[\hskip\labelsep\theorem@headerfont
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\ (##3)\theorem@separator}
    ]}

\renewtheoremstyle{marginbreak}%
  {\item[\hskip\labelsep\theorem@headerfont
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\theorem@separator}\newline
    ]}%
  {\item[\hskip\labelsep\theorem@headerfont
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{nonumberplain}%
  {\item[\theorem@headerfont\hskip\labelsep
    \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
  {\item[\theorem@headerfont\hskip \labelsep
```

```
    \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{nonumberbreak}%
  {\item[
%  \rlap{\vbox{\hbox{
    \hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}%
  {\item[
%  \rlap{\vbox{\hbox{
    \hskip\labelsep \theorem@headerfont
    \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
%  }\hbox{\strut}}}
    ]}

\renewtheoremstyle{empty}%
  {\item[]}%
  {\item[\theorem@headerfont \hskip\labelsep\relax
    \InlineClass{theoremheaderplain}{##3}]}

\renewtheoremstyle{emptybreak}%
  {\item[]}%
  {\item[\theorem@headerfont \hskip\labelsep\relax
    \InlineClass{theoremheaderplain}{##3}] \ \newline}
```

## § 201.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```
\ifbool{LWR@ntheoremamsthm}{}{%
% upright text via CSS
    \newtheoremstyle{plainupright}%
    {\item[\hskip\labelsep \theorem@headerfont
        \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
    {\item[\hskip\labelsep \theorem@headerfont
        \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}

% upright text and small caps header via CSS
    \newtheoremstyle{nonumberplainuprightsc}%
    {\item[\theorem@headerfont\hskip\labelsep
        \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
    {\item[\theorem@headerfont\hskip \labelsep
        \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}
}% not amsthm
```

## § 201.8 Renewed standard configuration

The following standard configuration is renewed using the new CSS:

```
\ifbool{LWR@ntheoremamsthm}{}{%
    \theoremstyle{plainupright}
    \theorembodyfont{\upshape}
    \theoremsymbol{\HTMLunicode{25A1}}% UTF-8 white box
    \renewtheorem{Example}{Example}
    \renewtheorem{example}{Example}
    \renewtheorem{Beispiel}{Beispiel}
    \renewtheorem{beispiel}{Beispiel}
    \renewtheorem{Bemerkung}{Bemerkung}
    \renewtheorem{bemerkung}{Bemerkung}
    \renewtheorem{Anmerkung}{Anmerkung}
    \renewtheorem{anmerkung}{Anmerkung}
    \renewtheorem{Remark}{Remark}
    \renewtheorem{remark}{Remark}
    \renewtheorem{Definition}{Definition}
    \renewtheorem{definition}{Definition}

    \theoremstyle{nonumberplainuprightsc}
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \renewtheorem{Proof}{Proof}
    \renewtheorem{proof}{Proof}
    \renewtheorem{Beweis}{Beweis}
    \renewtheorem{beweis}{Beweis}
    \qedsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof

    \theoremsymbol{}
}% not amsthm
```

## § 201.9 amsthm option

Only if the amsthm option was given:

```
\ifbool{LWR@ntheoremamsthm}{

\gdef\th@plain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyplain}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderplain}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
```

```
        \BlockClass{theorembodyplain}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberplain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyplain}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderplain}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyplain}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderplain}{##1\ (##3).}
        ]}}

\gdef\th@definition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodydefinition}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderdefinition}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodydefinition}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberdefinition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodydefinition}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderdefinition}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodydefinition}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderdefinition}{##1\ (##3).}
        ]}}

\gdef\th@remark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyremark}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderremark}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyremark}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberremark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyremark}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderremark}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyremark}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderremark}{##1\ (##3).}
```

```
        ]}}

\gdef\th@proof{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyproof}% new
        \LWR@inctheorem% new
      \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderproof}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
        \BlockClass{theorembodyroof}% new
        \LWR@inctheorem% new
     \item[\hskip\labelsep
%  \theorem@headerfont
  \InlineClass{theoremheaderproof}{##1\ (##3).}
        ]}}



\newcounter{proof}%
\if@thmmarks
    \newcounter{currproofctr}%
    \newcounter{endproofctr}%
\fi

\gdef\proofSymbol{\openbox}

\newcommand{\proofname}{Proof}

\newenvironment{proof}[1][\proofname]{
    \th@proof
    \def\theorem@headerfont{\itshape}%
    \normalfont
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \@thm{proof}{proof}{#1}
}%
{\@endtheorem}

}{}% amsthm option
```

## § 201.10 Ending a theorem

Patched for css:

```
\let\LWR@origendtheorem\@endtheorem
\renewcommand{\@endtheorem}{%
\ifbool{LWR@ntheoremmarks}{%
    \ifsetendmark%
    \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
    \setendmarkfalse%
    \fi%
}{}%
\LWR@origendtheorem%
\ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{}%
\endBlockClass%
}
```

## § 201.11 \NoEndMark

```
\gdef\NoEndMark{\global\setendmarkfalse}
```

## § 201.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

\thm@thmline {⟨*1: printed type*⟩} {⟨*2: #*⟩} {⟨*3: optional*⟩} {⟨*4: page*⟩}

```
\renewcommand{\thm@@thmline@noname}[4]{%
\hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
}

\renewcommand{\thm@@thmline@name}[4]{%
\hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
}
```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```
\def\thm@@thmline{\thm@@thmline@name}
```

Patch for css:

```
\def\listtheorems#1{
\LWR@htmlelementclass{nav}{lothm}%
\begingroup
\c@tocdepth=-2%
\def\thm@list{#1}\thm@processlist
\endgroup
\LWR@htmlelementclassend{nav}{lothm}%
}
```

## § 201.13 Symbols

Proof QED symbol:

```
\newcommand{\qed}{\qquad\the\qedsymbol}

\AtBeginDocument{
\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
}
```

## § 201.14 Cross-referencing

`\thref {⟨label⟩}`

```
\newcommand*{\thref}[1]{\cref{#1}}
```

---

File 121 **lwarp-overpic.sty**

# § 202 Package overpic

(*Emulates or patches code by* Rolf Niepraschk.)

Pkg overpic — overpic is patched for use by lwarp.

⚠ scaling — The macros \overpicfontsize and \overpicfontskip are used during HTML generation. These are sent to \fontsize to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the overpic and Overpic environments.

See section 72.2 for the print-mode version of \overpicfontsize and \overpicfontskip.

**for HTML output:**

```
\LWR@ProvidesPackagePass{overpic}
```

```
\newcommand*{\overpicfontsize}{12}
\newcommand*{\overpicfontskip}{14}

\BeforeBeginEnvironment{overpic}{%
    \begin{lateximage}%
    \fontsize{\overpicfontsize}{\overpicfontskip}%
    \selectfont%
}

\AfterEndEnvironment{overpic}{\end{lateximage}}

\BeforeBeginEnvironment{Overpic}{%
```

```
    \begin{lateximage}%
    \fontsize{\overpicfontsize}{\overpicfontskip}%
    \selectfont%
}

\AfterEndEnvironment{Overpic}{\end{lateximage}}
```

File 122 **lwarp-pagenote.sty**

## § 203 Package **pagenote**

Pkg pagenote pagenote works as-is, but the page option is disabled.

**for HTML output:**

```
\DeclareOption{page}{}
\LWR@ProvidesPackagePass{pagenote}
```

File 123 **lwarp-paralist.sty**

## § 204 Package **paralist**

(*Emulates or patches code by* BERND SCHANDL.)

Pkg paralist paralist is supported with minor changes.

**for HTML output:**

```
\LWR@ProvidesPackagePass{paralist}
```

The compact environments are identical to the regular ones:

```
\AtBeginEnvironment{compactitem}{\LWR@itemizestart}
\AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
\AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
\AtEndEnvironment{compactitem}{\LWR@listend}
\AtEndEnvironment{compactenum}{\LWR@listend}
\AtEndEnvironment{compactdesc}{\LWR@listend}
```

For the inline environments, revert `\item` to its original print-mode version:

```
\AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\LWR@origitem}
```

Manual formatting of the description labels:

```
\def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

File 124 **lwarp-parskip.sty**

## § 205 Package parskip

Pkg parskip parskip is ignored.

**for HTML output:** Discard all options for lwarp-parskip.

```
\LWR@ProvidesPackageDrop{parskip}
```

File 125 **lwarp-pdflscape.sty**

## § 206 Package pdflscape

Pkg pdflscape Emulated.

**for HTML output:** Discard all options for lwarp-pdflscape:

```
\LWR@ProvidesPackageDrop{pdflscape}
```

File 126 **lwarp-pdfsync.sty**

## § 207 Package pdfsync

(*Emulates or patches code by* J. Laurens.)

Pkg pdfsync Emulated.

**for HTML output:** Discard all options for lwarp-pdfsync:

```
\LWR@ProvidesPackageDrop{pdfsync}
```

```
\newcommand*{\pdfsync}{}
\newcommand*{\pdfsyncstart}{}
\newcommand*{\pdfsyncstop}{}
```

File 127 **lwarp-pfnote.sty**

## § 208 Package **pfnote**

Pkg pfnote pfnote is emulated.

⚠ pfnote numbers While emulating pfnote, lwarp is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. lwarp therefore uses continuous footnote numbering even for pfnote.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{pfnote}
```

File 128 **lwarp-placeins.sty**

## § 209 Package **placeins**

(*Emulates or patches code by* Donald Arseneau.)

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{placeins}

\newcommand*{\FloatBarrier}{}
```

File 129 **lwarp-prelim2e.sty**

## § 210 Package **prelim2e**

(*Emulates or patches code by* Martin Schröder.)

Pkg prelim2e Emulated.

**for HTML output:** Discard all options for lwarp-prelim2e:

```
\LWR@ProvidesPackageDrop{prelim2e}

\newcommand{\PrelimText}{}
\newcommand{\PrelimTextStyle}{}
\newcommand{\PrelimWords}{}
```

File 130 **lwarp-preview.sty**

## § 211 Package **preview**

Pkg preview

preview is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{preview}
```

```
\newenvironment{preview}{}{}
\newenvironment{nopreview}{}{}
\NewDocumentCommand{\PreviewMacro}{s o o +m}{}
\NewDocumentCommand{\PreviewEnvironment}{s o o +m}{}
\newcommand{\PreviewSnarfEnvironment}[2][]{}
\NewDocumentCommand{\PreviewOpen}{s o}{}
\NewDocumentCommand{\PreviewClose}{s o}{}
\let\ifPreview\iffalse% \fi for syntax highlighting
```

File 131 **lwarp-quotchap.sty**

## § 212 Package **quotchap**

(*Emulates or patches code by* Karsten Tinnefeld, Jan Klever.)

Pkg quotchap

quotchap is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{quotchap}
```

```
\newcommand{\@quotchap}{}
\newlength{\LWR@quotchapwidth}

\let\@printcites\relax

\newcommand*{\@iprintcites}{%
```

Place the quotes inside a `<div>` of class `quotchap`, of the maximum selected width:

```
\uselengthunit{PT}%
\begin{BlockClass}[max-width: \rndprintlength{\LWR@quotchapwidth}]{quotchap}
%\begin{minipage}{\LWR@quotchapwidth}
\@quotchap
%\end{minipage}
\end{BlockClass}
```

Deactivate the quote printing:

```
\global\let\@printcites\relax
}

\NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:
https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment

```
\setlength{\LWR@quotchapwidth}{#1*2}%
\global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
\global\let\@quotchap\BODY
\global\let\@printcites\@iprintcites%
}
```

The quotation author is placed inside a `<div>` of class `qauthor`:

```
\newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

`\qsetcnfont` is ignored:

```
\newcommand{\qsetcnfont}[1]{}
```

---

File 132 **`lwarp-ragged2e.sty`**

# § 213 Package **ragged2e**

(*Emulates or patches code by* MARTIN SCHRÖDER.)

Pkg `ragged2e`

ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{ragged2e}

\newcommand*{\Centering}{\centering}
\newcommand*{\RaggedLeft}{\raggedleft}
\newcommand*{\RaggedRight}{\raggedright}
\newcommand*{\justifying}{}
\newlength{\CenteringLeftskip}
```

```
\newlength{\RaggedLeftLeftskip}
\newlength{\RaggedRightLeftskip}
\newlength{\CenteringRightskip}
\newlength{\RaggedLeftRightskip}
\newlength{\RaggedRightRightskip}
\newlength{\CenteringParfillskip}
\newlength{\RaggedLeftParfillskip}
\newlength{\RaggedRightParfillskip}
\newlength{\JustifyingParfillskip}
\newlength{\CenteringParindent}
\newlength{\RaggedLeftParindent}
\newlength{\RaggedRightParindent}
\newlength{\JustifyingParindent}
\newenvironment*{Center}{\center}{\endcenter}
\newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
\newenvironment*{FlushRight}{\flushright}{\endflushright}
\newenvironment*{justify}{\justifying}{\endjustifying}
```

File 133 `lwarp-realscripts.sty`

## § 214 Package **realscripts**

(*Emulates or patches code by* Will Robertson.)

Pkg realscripts

realscripts is emulated. See `lwarp.css` for the <span> of class `supsubscript`.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{realscripts}
```

```
\let\realsuperscript\textsuperscript
\let\realsubscript\textsubscript

\let\fakesuperscript\textsuperscript
\let\fakesubscript\textsubscript

\newlength{\subsupersep}

\newcommand*{\LWR@realscriptsalign}{}

\newcommand*{\LWR@setrealscriptsalign}[1]{%
\renewcommand*{\LWR@realscriptsalign}{}%
\ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
\ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
}

\DeclareDocumentCommand \textsubsuperscript {s O{l} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsuperscript{#4}\textsubscript{#3}%
}%
}

\DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsubscript{#4}\textsuperscript{#3}%
}%
}
```

File 134 **lwarp-relsize.sty**

# § 215 Package **relsize**

(*Emulates or patches code by* Donald Arseneau, Bernie Cosell, Matt Swift.)

Pkg relsize

relsize is patched for use by lwarp.

For HTML only the inline macros are supported: \textlarger, \textsmaller, and \textscale. Each becomes an inline span of a modified font-size.

\relsize, \larger, \smaller, and \relscale are ignored.

While creating SVG math for HTML, the original definitions are temporarilty restored, and so should work as expected.

⚠ not small

The HTML browser's setting for minumum font size may limit how small the output will be displayed.

**for HTML output:**

```
\LWR@ProvidesPackagePass{relsize}
```

```
\let\LWR@origrelsize\relsize
\LetLtxMacro\LWR@origlarger\larger
\LetLtxMacro\LWR@origsmaller\smaller
\let\LWR@relscale\relscale
\LetLtxMacro\LWR@origtextlarger\textlarger
\LetLtxMacro\LWR@origtextsmaller\textsmaller
\let\LWR@textscale\textscale

\appto{\LWR@restoreorigformatting}{%
\let\relsize\LWR@origrelsize%
\LetLtxMacro\larger\LWR@origlarger%
\LetLtxMacro\smaller\LWR@origsmaller%
\let\relscale\LWR@relscale%
\LetLtxMacro\textlarger\LWR@origtextlarger%
\LetLtxMacro\textsmaller\LWR@origtextsmaller%
\let\textscale\LWR@textscale%
}

\newcounter{LWR@relsizetemp}

\renewcommand*{\relsize}[1]{}
\renewcommand*{\larger}[1][]{}
\renewcommand*{\smaller}[1][]{}
\renewcommand*{\relscale}[1]{}

\renewcommand*{\textlarger}[2][1]{%
\setcounter{LWR@relsizetemp}{100+(#1*20)}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textlarger}{#2}%
}

\renewcommand*{\textsmaller}[2][1]{%
\setcounter{LWR@relsizetemp}{100-(#1*20)}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textsmaller}{#2}%
}

\renewcommand*{\textscale}[2]{%
\setcounter{LWR@relsizetemp}{100*\real{#1}}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textscale}{#2}%
}
```

File 135 **lwarp-romanbar.sty**

## § 216 Package **romanbar**

(*Emulates or patches code by* H.-Martin Münch.)

Pkg romanbar romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

**for HTML output:**

```
\LWR@ProvidesPackagePass{romanbar}
```

```
\DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
\InlineClass[%
    text-decoration: overline underline ;
]{romanbar}{#1}%
}
```

File 136 **lwarp-romanbarpagenumber.sty**

## § 217 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{romanbarpagenumber}
```

File 137 **lwarp-rotating.sty**

## § 218 Package **rotating**

(*Emulates or patches code by* Robin Fairbairns, Sebastian Rahtz, Leonor Barroca.)

Pkg rotating rotating is emulated.

All rotations are ignored in HTML output.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{rotating}
```

```
\LetLtxMacro\sidewaystable\table
\let\endsidewaystable\endtable

\LetLtxMacro\sidewaysfigure\figure
\let\endsidewaysfigure\endfigure

\newenvironment*{sideways}{}{}
\newenvironment*{turn}[1]{}{}
\newenvironment*{rotate}[1]{}{}
\NewDocumentCommand{\turnbox}{m +m}{#2}
\let\rotcaption\caption
\let\@makerotcaption\@makecaption
```

File 138 **lwarp-rotfloat.sty**

## § 219 Package **rotfloat**

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg rotfloat rotfloat is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{rotfloat}

\RequirePackage{float}
```

`\newfloat` {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the `\newfloat` command from the float package. Sideways floats are `\let` to the same as regular floats.

“placement” is ignored.

```
\RenewDocumentCommand{\newfloat}{m m m o}{%
\IfValueTF{#4}
{
    \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
}
{
    \DeclareFloatingEnvironment[fileext=#3]{#1}
    \DeclareFloatingEnvironment[fileext=#3]{sideways#1}
}
\csletcs{sideways#1}{#1}
\csletcs{endsideways#1}{end#1}
```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
\cslet{listof#1s}\relax
\cslet{listof#1es}\relax
}
```

---

File 139 **`lwarp-savetrees.sty`**

# § 220 Package savetrees

Pkg `savetrees` Emulated.

**for HTML output:** Discard all options for lwarp-savetrees:

```
\LWR@ProvidesPackageDrop{savetrees}
```

File 140 **lwarp-scalefnt.sty**

## § 221 Package **scalefnt**

(*Emulates or patches code by* D. Carlisle.)

Pkg scalefnt scalefnt is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scalefnt}
```

```
\DeclareRobustCommand\scalefont[1]{}
```

File 141 **lwarp-scrextend.sty**

## § 222 Package **scrextend**

Pkg scrextend scrextend is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scrextend}
```

```
\DeclareDocumentCommand{\setkomafont}{m m}{}
\DeclareDocumentCommand{\addkomafont}{m m}{}
\DeclareDocumentCommand{\usekomafont}{m}{}

\DeclareDocumentCommand{\usefontofkomafont}{m}{}
\DeclareDocumentCommand{\useencodingofkomafont}{m}{}
\DeclareDocumentCommand{\usesizeofkomafont}{m}{}
\DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
\DeclareDocumentCommand{\useseriesofkomafont}{m}{}
\DeclareDocumentCommand{\useshapeofkomafont}{m}{}

\AtBeginDocument{
\let\LWR@maketitle\maketitle
\DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
}

\DeclareDocumentCommand{\extratitle}{m}{}
\DeclareDocumentCommand{\titlehead}{m}{}
```

```
\DeclareDocumentCommand{\subject}{m}{}
\DeclareDocumentCommand{\publishers}{m}{\published{#1}}
\DeclareDocumentCommand{\uppertitleback}{m}{}
\DeclareDocumentCommand{\lowertitleback}{m}{}
\DeclareDocumentCommand{\dedication}{m}{}

\DeclareDocumentCommand{\ifthispageodd}{m m}{#1}

\DeclareDocumentCommand{\titlepagestyle}{}{}

\DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleplainpage}{}{}
\DeclareDocumentCommand{\cleardoublestandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleoddemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleoddplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddstandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}

\DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
  \begingroup\let\thefootnotemark\multfootsep\@makefnmark\endgroup
}

\DeclareDocumentCommand{\multfootsep}{}{,}

\DeclareDocumentCommand{\footref}{m}{%
  \begingroup
    \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
  \endgroup
  \@footnotemark
}

\DeclareDocumentCommand{\deffootnote}{o m m m}{}
\DeclareDocumentCommand{\deffootnotemark}{m}{}
\DeclareDocumentCommand{\setfootnoterule}{o m}{}
\DeclareDocumentCommand{\raggedfootnote}{}{}

\DeclareDocumentCommand{\dictum}{o m}{
\begin{LWR@BlockClassWP}{text-align:right}{}{dictum}
    #2
    \IfValueT{#1}
    {
        \ifbool{FormatWP}
        {\begin{BlockClass}[border-top:1px solid gray]{dictumauthor}}
        {\begin{BlockClass}{dictumauthor}}
        \dictumauthorformat{#1}
        \end{BlockClass}
    }
\end{LWR@BlockClassWP}
}

\DeclareDocumentCommand{\dictumwidth}{}{}
\DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
\DeclareDocumentCommand{\dictumrule}{}{}
\DeclareDocumentCommand{\raggeddictum}{}{}
\DeclareDocumentCommand{\raggeddictumtext}{}{}
\DeclareDocumentCommand{\raggeddictumauthor}{}{}

\DeclareDocumentEnvironment{labeling}{o m}
{%
\def\sc@septext{#1}%
\list{}{}%
\let\makelabel\labelinglabel%
}
{
\endlist
}

\DeclareDocumentCommand{\labelinglabel}{m}{%
#1 \qquad \sc@septext%
}

\let\addmargin\relax
\let\endaddmargin\relax
\cslet{addmargin*}{\relax}
\cslet{endaddmargin*}{\relax}

\NewDocumentEnvironment{addmargin}{s O{} m}
{
\uselengthunit{PT}%
\setlength{\LWR@templengthtwo}{#3}
\ifblank{#2}
{
    \begin{BlockClass}[
        margin-left:\rndprintlength{\LWR@templengthtwo} ;
        margin-right:\rndprintlength{\LWR@templengthtwo}
    ]{addmargin}
}
{
    \setlength{\LWR@templengthone}{#2}
    \begin{BlockClass}[
        margin-left:\rndprintlength{\LWR@templengthone} ;
        margin-right:\rndprintlength{\LWR@templengthtwo}
    ]{addmargin}
```

```
}
}
{\end{BlockClass}}
```

Ref to create a starred environment:
https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment

```

\ExplSyntaxOn
\cs_new:cpn {addmargin*} {\addmargin*}
\cs_new_eq:cN {endaddmargin*} \endaddmargin
\ExplSyntaxOff

\DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}
```

File 142 **lwarp-scrhack.sty**

## § 223 Package **scrhack**

Pkg scrhack

scrhack is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scrhack}
```

File 143 **lwarp-scrlayer.sty**

## § 224 Package **scrlayer**

(*Emulates or patches code by* Markus Kohm.)

Pkg scrlayer

scrlayer is emulated.

⚠ Not yet tested! Please send bug reports!

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scrlayer}
```

```
\newcommand*{\DeclareSectionNumberDepth}[2]{}
\newcommand*{\DeclareLayer}[2][]{}
\newcommand*{\DeclareNewLayer}[2][]{}
\newcommand*{\ProvideLayer}[2][]{}
\newcommand*{\RedeclareLayer}[2][]{}
\newcommand*{\ModifyLayer}[2][]{}
\newcommand*{\layerhalign}{}
\newcommand*{\layervalign}{}
```

```
\newcommand*{\layerxoffset}{}
\newcommand*{\layeryoffset}{}
\newcommand*{\layerwidth}{}
\newcommand*{\layerheight}{}
\providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
\newcommand*{\putUL}[1]{}
\newcommand*{\putUR}[1]{}
\newcommand*{\putLL}[1]{}
\newcommand*{\putLR}[1]{}
\newcommand*{\putC}[1]{}
\newcommand*{\GetLayerContents}[1]{}
\newcommand{\IfLayerExists}[3]{#3}
\newcommand*{\DestroyLayer}[1]{}
\newcommand*{\layercontentsmeasure}{}
\newcommand*{\currentpagestyle}{}
\newcommand*{\BeforeSelectAnyPageStyle}[1]{}
\newcommand*{\AfterSelectAnyPageStyle}[1]{}
\newcommand*{\DeclarePageStyleAlias}[2]{}
\newcommand*{\DeclareNewPageStyleAlias}[2]{}
\newcommand*{\ProvidePageStyleAlias}[2]{}
\newcommand*{\RedeclarePageStyleAlias}[2]{}
\newcommand*{\DestroyPageStyleAlias}[1]{}
\newcommand*{\GetRealPageStyle}[1]{}
\newcommand*{\DeclarePageStyleByLayers}[3][]{}
\newcommand*{\DeclareNewPageStyleByLayers}[3][]{}
\newcommand*{\ProvidePageStyleByLayers}[3][]{}
\newcommand*{\RedeclarePageStyleByLayers}[3][]{}
\NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}{}
\newcommand*{\AddLayersToPageStyle}[2]{}
\newcommand*{\AddLayersAtBeginOfPageStyle}[2]{}
\newcommand*{\AddLayersAtEndOfPageStyle}[2]{}
\newcommand*{\RemoveLayersFromPageStyle}[2]{}
\newcommand*{\AddLayersToPageStyleBeforeLayer}[3]{}
\newcommand*{\AddLayersToPageStyleAfterLayer}[3]{}
\newcommand*{\UnifyLayersAtPageStyle}[1]{}
\newcommand*{\ModifyLayerPageStyleOptions}[2]{}
\newcommand*{\AddToLayerPageStyleOptions}[2]{}
\newcommand{\IfLayerPageStyleExists}[3]{#3}
\newcommand{\IfRealLayerPageStyleExists}[3]{#3}
\newcommand{\IfLayerAtPageStyle}[4]{#4}
\newcommand{\IfSomeLayerAtPageStyle}[4]{#4}
\newcommand{\IfLayersAtPageStyle}[4]{#4}
\newcommand*{\DestroyRealLayerPageStyle}[1]{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\@mkleft}[1]{}
\newcommand*{\@mkright}[1]{}
```

```
\newcommand*{\@mkdouble}[1]{}
\newcommand*{\@mkboth}[2]{}
\newcommand*{\scrlayerInitInterface}[1][]{}
\newcommand{\scrlayerAddToInterface}[3][]{}
\newcommand{\scrlayerAddCsToInterface}[3][]{}
\newcommand{\scrlayerOnAutoRemoveInterface}[2][]{}
```

File 144 `lwarp-scrlayer-notecolumn.sty`

## § 225 Package **scrlayer-notecolumn**

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-notecolumn` scrlayer-notecolumn is emulated.

⚠ Not yet tested! Please send bug reports!

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scrlayer-notecolumn}
```

```
\newcommand*{\DeclareNoteColumn}[2][]{}
\newcommand*{\DeclareNewNoteColumn}[2][]{}
\newcommand*{\ProvideNoteColumn}[2][]{}
\newcommand*{\RedeclareNoteColumn}[2][]{}
\NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
\newcommand*{\syncwithnotecolumn}[1][]{}
\newcommand*{\syncwithnotecolumns}[1][]{}
\newcommand*{\clearnotecolumn}[1][]{}
\newcommand*{\clearnotecolumns}[1][]{}
```

File 145 `lwarp-scrlayer-scrpage.sty`

## § 226 Package **scrlayer-scrpage**

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-scrpage` scrlayer-scrpage is emulated.

⚠ Not yet tested! Please send bug reports!

**for HTML output:**

```
\LWR@ProvidesPackageDrop{scrlayer-scrpage}
```

```
\@ifundefined{footheight}{\newlength\footheight}{}
\NewDocumentCommand{\lehead}{s o m}{}
\NewDocumentCommand{\cehead}{s o m}{}
```

```
\NewDocumentCommand{\rehead}{s o m}{}
\NewDocumentCommand{\lohead}{s o m}{}
\NewDocumentCommand{\cohead}{s o m}{}
\NewDocumentCommand{\rohead}{s o m}{}
\NewDocumentCommand{\lefoot}{s o m}{}
\NewDocumentCommand{\cefoot}{s o m}{}
\NewDocumentCommand{\refoot}{s o m}{}
\NewDocumentCommand{\lofoot}{s o m}{}
\NewDocumentCommand{\cofoot}{s o m}{}
\NewDocumentCommand{\rofoot}{s o m}{}
\NewDocumentCommand{\ohead}{s o m}{}
\NewDocumentCommand{\chead}{s o m}{}
\NewDocumentCommand{\ihead}{s o m}{}
\NewDocumentCommand{\ofoot}{s o m}{}
\NewDocumentCommand{\cfoot}{s o m}{}
\NewDocumentCommand{\ifoot}{s o m}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\defpairofpagestyles}[3][]{}
\newcommand*{\newpairofpagestyles}[3][]{}
\newcommand*{\renewpairofpagestyles}[3][]{}
\newcommand*{\providepairofpagestyles}[3][]{}
\newcommand*{\clearmainofpairofpagestyles}
\newcommand*{\clearplainofpairofpagestyles}
\newcommand*{\clearpairofpagestyles}
\NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\providetriplepagestyle}{m o o m m m m m m}{}
\newcommand*{\defpagestyle}[3]{}
\newcommand*{\newpagestyle}[3]{}
\newcommand*{\providepagestyle}[3]{}
\newcommand*{\renewpagestyle}[3]{}
```

---

File 146 **lwarp-section.sty**

## § 227 Package **section**

Pkg section

section is ignored.

(*Emulates or patches code by* Oliver Pretzel.)

**for HTML output:**

```
\LWR@ProvidesPackageDrop{section}
```

```
\ifx\chapter\undefined
```

```
 \def\chsize{\Large}\def\hdsize{\huge}\else
 \def\chsize{\huge}\def\hdsize{\Huge}
\fi
\let\ttsize\LARGE
\let\ausize\large
\let\dasize\large
\let\secsize\Large
\let\subsize\large
\let\hdpos\raggedright
\newcounter{hddepth}
\let\fpind\relax
\def\ttfnt{}
\def\hdfnt{}
\def\fefnt{}
\def\thfnt{}
\def\pgfnt{}
\def\hmkfnt{}
\let\mkcse\uppercase
\def\hddot{}
\def\cpdot{:}
\def\nmdot{}
\ifx\secindent\undefined
 \newdimen\secindent
 \newskip\secpreskp
 \newskip\secpstskp
 \newdimen\subindent
 \newskip\subpreskp
 \newskip\subpstskp
 \newskip\parpstskp
 \newcount\c@hddepth
\fi
```

File 147 **lwarp-sectsty.sty**

## § 228 Package **sectsty**

(*Emulates or patches code by* Rowland McDonnell.)

Pkg sectsty

sectsty is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{sectsty}
```

```
\newcommand*{\partfont}          [1] {}
\newcommand*{\partnumberfont}    [1] {}
\newcommand*{\parttitlefont}     [1] {}
\newcommand*{\chapterfont}       [1] {}
\newcommand*{\chapternumberfont} [1] {}
```

```
\newcommand*{\chaptertitlefont}  [1] {}
\newcommand*{\sectionfont}       [1] {}
\newcommand*{\subsectionfont}    [1] {}
\newcommand*{\subsubsectionfont} [1] {}
\newcommand*{\paragraphfont}     [1] {}
\newcommand*{\subparagraphfont}  [1] {}
\newcommand*{\minisecfont}  [1] {}
\newcommand*{\allsectionsfont}[1] {}
\newcommand{\nohang}{}
```

\sectionrule is only to be used in *font commands, thus it is ignored.

```
\newcommand*{\sectionrule}[5]{}

\def\ulemheading#1#2{}
```

File 148 **lwarp-setspace.sty**

## § 229 Package **setspace**

(*Emulates or patches code by* Robin Fairbairns.)

Pkg setspace

setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{setspace}

\newcommand*{\setstretch}[1]{}
\newcommand*{\SetSinglespace}[1]{}
\newcommand*{\singlespacing}{}
\newcommand*{\onehalfspacing}{}
\newcommand*{\doublespacing}{}

\newenvironment*{singlespace}
{
\LWR@forcenewpage
\BlockClass{singlespace}
}
{\endBlockClass}

\newenvironment*{singlespace*}
{
\LWR@forcenewpage
\BlockClass{singlespace}
}
{\endBlockClass}
```

```

\newenvironment*{spacing}[1]{

}{

}

\newenvironment*{onehalfspace}
{
\LWR@forcenewpage
\BlockClass{onehalfspace}
}
{\endBlockClass}

\newenvironment*{doublespace}
{
\LWR@forcenewpage
\BlockClass{doublespace}
}
{\endBlockClass}
```

File 149 **lwarp-shadow.sty**

## § 230 Package **shadow**

(*Emulates or patches code by* Mauro Orlandini.)

Pkg shadow

shadow is emulated.

**for HTML output:** Discard all options for lwarp-shadow:

```
\LWR@ProvidesPackageDrop{shadow}
```

```
\newdimen\sboxsep
\newdimen\sboxrule
\newdimen\sdim

\newcommand{\shabox}[1]{%
\InlineClass{shabox}{#1}%
}
```

File 150 **lwarp-showidx.sty**

## § 231 Package **showidx**

Pkg showidx showidx is ignored.

**for HTML output:** Discard all options for lwarp-showidx:

```
\LWR@ProvidesPackageDrop{showidx}
```

File 151 **lwarp-showkeys.sty**

## § 232 Package **showkeys**

(*Emulates or patches code by* David Carlisle, Morten Høgholm.)

Pkg showkeys showkeys is ignored.

**for HTML output:** Discard all options for lwarp-showkeys:

```
\LWR@ProvidesPackageDrop{showkeys}
```

```
\NewDocumentCommand{\showkeys}{s}{}
```

File 152 **lwarp-sidecap.sty**

## § 233 Package **sidecap**

(*Emulates or patches code by* Rolf Niepraschk, Hubert Gässlein.)

Pkg sidecap sidecap is emulated.

**for HTML output:** Discard all options for lwarp-sidecap.

```
\LWR@ProvidesPackageDrop{sidecap}
```

See:
http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment
regarding the creation of starred environments with xparse.

```
\NewDocumentEnvironment{SCtable}{soo}
{\IfValueTF{#3}{\table[#3]}{\table}}
{\endtable}

\ExplSyntaxOn
\cs_new:cpn {SCtable*} {\SCtable*}
\cs_new_eq:cN {endSCtable*} \endSCtable
\ExplSyntaxOff


\NewDocumentEnvironment{SCfigure}{soo}
{\IfValueTF{#3}{\figure[#3]}{\figure}}
{\endfigure}

\ExplSyntaxOn
\cs_new:cpn {SCfigure*} {\SCfigure*}
\cs_new_eq:cN {endSCfigure*} \endSCfigure
\ExplSyntaxOff


\newenvironment*{wide}{}{}
```

File 153 **lwarp-sidenotes.sty**

## § 234 Package **sidenotes**

(*Emulates or patches code by* Andy Thomas, Oliver Schebaum.)

Pkg sidenotes Patched for lwarp.

**for HTML output:** Load the original package:

```
\LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
\RenewDocumentCommand \sidecaption {s o o m}
{
    \LWR@stoppars
  \captionsetup{style=sidecaption}
  \IfBooleanTF{#1}
  { % starred
    \IfNoValueOrEmptyTF{#2}
    {\marginnote{\caption*{#4}}}
    {\marginnote{\caption*{#4}}[#2]}
  }
  { % unstarred
  \IfNoValueOrEmptyTF{#2}
    {\def\@sidenotes@sidecaption@tof{#4}}
    {\def\@sidenotes@sidecaption@tof{#2}}
  \IfNoValueOrEmptyTF{#3}
    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
  }
    \LWR@startpars
}
```

Borrowed from the lwarp version of keyfloat:

```
\NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
{% start
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}

\RenewDocumentEnvironment{marginfigure}{o}
  {\begin{KFLTsidenotes@marginfloat}{figure}}
  {\end{KFLTsidenotes@marginfloat}}

\RenewDocumentEnvironment{margintable}{o}
  {\begin{KFLTsidenotes@marginfloat}{table}}
  {\end{KFLTsidenotes@marginfloat}}
```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the LaTeX 2ε `article.cls` source:

```
\renewenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}

\renewenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

File 154 **lwarp-siunitx.sty**

## § 235 Package **siunitx**

(*Emulates or patches code by* Joseph Wright.)

Pkg siunitx siunitx is patched for use by lwarp.

⚠ per-mode Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ math mode required Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the siunitx extension for MathJax is not currently hosted at any public CDN, thus siunitx is not usable with MathJax unless a local copy of this extension is created first.

**for HTML output:**

```
\LWR@ProvidesPackagePass{siunitx}
```

```
\AtBeginDocument{% in case textcomp was not loaded
\DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
\DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
\DeclareSIUnit\elementarycharge{\textit{e}}
\DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
\DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
}% AtBeginDocument
```

The following is executed by siunitx at the end of `\document`. `\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original.

```
\ExplSyntaxOn
\cs_undefine:N \__siunitx_set_math_fam:n
\cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
  \LWR@traceinfo{sunitx set math fam}
 \int_new:c { c__siunitx_math #1 _int }
 \group_begin:
  \LetLtxMacro\@ensuredmath\LWR@origensuredmath
  \hbox_set:Nn \l__siunitx_tmp_box
   {
    \ensuremath
     {
      \use:c { math #1 }
       {
        \int_gset:cn { c__siunitx_math #1 _int } { \fam }
       }
```

```
          }
      }
  \group_end:
    \LWR@traceinfo{sunitx set math fam: done}
}

\cs_undefine:N \__siunitx_combined_output:n
\cs_new_protected:Npn \__siunitx_combined_output:n #1 {
    \group_begin:
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath
  \bool_if:NTF \l__siunitx_number_parse_bool
    {
      \tl_clear:N \l__siunitx_number_out_tl
      \bool_set_false:N \l__siunitx_number_compound_bool
      \__siunitx_number_output_parse:n {#1}
    }
    {
      \__siunitx_unit_output_pre_print:
      \__siunitx_print:nn { number } { \ensuremath {#1} }
      \__siunitx_unit_output_print:
    }
    \group_end:
}

\ExplSyntaxOff
```

File 155 **lwarp-soul.sty**

## § 236 Package **soul**

(*Emulates or patches code by* Melchior FRANZ.)

Pkg soul Emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{soul}[2003/11/17]
\RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
\newcommand*{\LWR@soululcolor}{}

\newcommand*{\LWR@soulstcolor}{}

% \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
% \newcommand*{\LWR@soulhlcolor}{LWR@soulhlcolordefault}
\newcommand*{\LWR@soulhlcolor}{}
```

\so {⟨*text*⟩}

Basic markup with CSS:

```
\newcommand{\so}[1]{%
\LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
}
```

\caps {⟨*text*⟩}

```
\newcommand{\caps}[1]{%
\LWR@HTMLtextstyle%
    {font-variant:small-caps;letter-spacing:.1ex}%
    {capsspacing}{#1}%
}
```

\LWR@soulcolor {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
\newcommand{\LWR@soulcolor}[5]{%
\ifcsempty{#2}%
{\LWR@HTMLtextstyle{#5}{#3}{#1}}%
{%
    \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
    \LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}%
}%
}
```

```
\newcommand{\ul}[1]{%
\LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
    {text-decoration:underline;text-decoration-skip;}%
}

\newcommand{\st}[1]{
\LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
    {text-decoration:line-through}%
}

\newcommand{\hl}[1]{
\LWR@soulcolor{#1}{LWR@soulhlcolor}{highlight}{background-color}%
    {background:\#F8E800}
}
```

Nullified:

```
\newcommand*{\soulaccent}[1]{}
\newcommand*{\soulregister}[2]{}
\newcommand{\sloppyword}[1]{#1}
```

```
\newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
\newcommand*{\resetso}{}
\newcommand*{\capsdef}[5]{}
\newcommand*{\capsreset}{}
\newcommand*{\capssave}[1]{}
\newcommand*{\capsselect}[1]{}
\newcommand*{\setul}[2]{}
\newcommand*{\resetul}{}
\newcommand*{\setuldepth}[1]{}
\newcommand*{\setuloverlap}[1]{}
```

Set colors:

```
\newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
\newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
\newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
\let\textso\so
\let\textul\ul
\let\texthl\hl
\let\textcaps\caps
```

---

# File 156 `lwarp-soulpos.sty`

## § 237 Package **soulpos**

(*Emulates or patches code by* Javier Bezos.)

Pkg soulpos

soulpos is emulated.

**for HTML output:**

```
\RequirePackage{soul}
\RequirePackage{soulutf8}
\LWR@ProvidesPackageDrop{soulpos}
```

```
\NewDocumentCommand{\ulposdef}{m o m}{}

\newdimen\ulwidth

\newcommand\ifulstarttype[1]{%
\expandafter\@secondoftwo%
}

\newcommand\ifulendtype[1]{%
\expandafter\@secondoftwo%
}
```

```

\newcommand{\ulstarttype}{0}
\newcommand{\ulendtype}{0}
\newcommand\ulpostolerance{0}%
```

File 157 **`lwarp-soulutf8.sty`**

## § 238 Package **soulutf8**

Pkg `soulutf8`

soulutf8 is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{soulutf8}
```

File 158 **`lwarp-stabular.sty`**

## § 239 Package **stabular**

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `stabular`

stabular is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{stabular}
```

Env `stabular` [⟨*vpos*⟩] {⟨*colspec*⟩}

```
\newenvironment{stabular}[2][c]
{
\renewcommand{\noalign}[1]{}
\begin{tabular}[#1]{#2}
}
{\end{tabular}}
```

Env `stabular` {⟨*width*⟩} [⟨*vpos*⟩] {⟨*colspec*⟩}

```
\NewDocumentEnvironment{stabular*}{m o m}
{
\renewcommand{\noalign}[1]{}
\begin{tabular}[#2]{#3}
}
{\end{tabular}}
```

File 159 **lwarp-subfig.sty**

## § 240 Package **subfig**

(*Emulates or patches code by* Steven Douglas Cochran.)

Pkg subfig

subfig is supported and patched by lwarp.

⚠ lof/lotdepth

At present, the package options for lofdepth and lotdepth are not working. These counters must be set separately after the package has been loaded.

horizontal spacing

In the document source, use \hfill and \hspace* between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

**for HTML output:** Accept all options for lwarp-subfig:

```
\LWR@ProvidesPackagePass{subfig}
```

\sf@@@subfloat {⟨*1 type*⟩} [⟨*2 lof entry*⟩] [⟨*3 caption*⟩] {⟨*4 contents*⟩}

The outer minipage allows side-by-side subfloats with \hfill between.

```
\long\def\sf@@@subfloat#1[#2][#3]#4{%
\begin{minipage}{\linewidth}% new
\LWR@stoppars% new
    \@ifundefined{FBsc@max}{}%
        {\FB@readaux{\let\FBsuboheight\relax}}%
    \@tempcnta=\@ne
    \if@minipage
      \@tempcnta=\z@
    \else\ifdim \lastskip=\z@ \else
      \@tempcnta=\tw@
    \fi\fi
    \ifmaincaptiontop
      \sf@top=\sf@nearskip
      \sf@bottom=\sf@farskip
    \else
      \sf@top=\sf@farskip
      \sf@bottom=\sf@nearskip
    \fi
    \leavevmode
    \setbox\@tempboxa \hbox{#4}%
    \@tempdima=\wd\@tempboxa
    \@ifundefined{FBsc@max}{}%
```

```
        {\global\advance\Xhsize-\wd\@tempboxa
         \dimen@=\ht\@tempboxa
         \advance\dimen@\dp\@tempboxa
         \ifdim\dimen@>\FBso@max
           \global\FBso@max\dimen@
         \fi}%
    \vtop\bgroup
      \vbox\bgroup
        \ifcase\@tempcnta
          \@minipagefalse
        \or
          \vskip\sf@top
        \or
          \ifdim \lastskip=\z@ \else
            \@tempskipb\sf@top\relax\@xaddvskip
          \fi
        \fi
        \sf@ifpositiontop{%
          \ifx \@empty#3\relax \else
            \sf@subcaption{#1}{#2}{#3}%
            \vskip\sf@capskip
            \vskip\sf@captopadj
          \fi\egroup
          \hrule width0pt height0pt depth0pt
            \LWR@startpars% new
%  \box\@tempboxa
            #4
            \LWR@stoppars% new
        }{%
        \LWR@startpars% new
        \@ifundefined{FBsc@max}%
            {
%  \box\@tempboxa
                #4
            }%
            {\ifx\FBsuboheight\relax
%                \box\@tempboxa
                #4
             \else
%                \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
                #4
             \fi}%
        \LWR@stoppars% new
          \egroup
          \ifx \@empty#3\relax \else
            \vskip\sf@capskip
            \hrule width0pt height0pt depth0pt
            \sf@subcaption{#1}{#2}{#3}%
         \fi
        }%
      \vskip\sf@bottom
    \egroup
    \@ifundefined{FBsc@max}{}%
        {\addtocounter{FRobj}{-1}%
         \ifnum\c@FRobj=0\else
           \subfloatrowsep
         \fi}%
    \ifmaincaptiontop\else
      \global\advance\@nameuse{c@\@captype}\m@ne
    \fi
\end{minipage}% new
\LWR@startpars% new
   \endgroup\ignorespaces%
}%
```

\sf@subcaption {⟨*1 type*⟩} {⟨*2 lof entry*⟩} {⟨*3 caption*⟩}

```
\long\def\sf@subcaption#1#2#3{%
\LWR@stoppars% new
  \ifx \relax#2\relax \else
    \bgroup
      \let\label=\@gobble
      \let\protect=\string
      \def\@subcaplabel{%
        \caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
      \sf@updatecaptionlist{#1}{#2}{\the\value{\@captype}}{\the\value{#1}}%
    \egroup
  \fi
  \bgroup
    \ifx \relax#3\relax
      \let\captionlabelsep=\relax
    \fi
%     \setbox0\vbox{%
%        \hb@xt@\the\@tempdima{%
%
% %          \hss
% %         \parbox[t]{\the\@tempdima}{%
%           \caption@make
%               {\@nameuse{sub\@captype name}}%
%               {\@nameuse{thesub\@captype}}%
%               {#3}
% %  }%
% %            \hss
%   }
%  }%
    \@ifundefined{FBsc@max}%
%         {\box0}%
        {
%  \parbox[t]{\the\@tempdima}{%
\LWR@traceinfo{sfsubcap B1}% new
            \LWR@figcaption% new
            \caption@make
                {\@nameuse{sub\@captype name}}%
                {\@nameuse{thesub\@captype}}%
                {#3}
            \LWR@figcaption% new
\LWR@traceinfo{sfsubcap B2}% new
%  }%
        }%
        {\dimen@\ht0%
         \advance\dimen@\dp0%
         \ifdim\dimen@>\FBsc@max
           \global\FBsc@max\dimen@
         \fi
         \FB@readaux{\let\FBsubcheight\relax}%
         \ifx\FBsubcheight\relax
           \def\next{
%  \parbox[t]{\the\@tempdima}
            }%
         \else
           \def\next{
%  \parbox[t][\FBsubcheight][t]{\the\@tempdima}
            }%
         \fi
         \vbox{%
%            \hb@xt@\the\@tempdima{%

%              \hss
%              \next{%
\LWR@traceinfo{sfsubcap C1}% new
               \caption@make
                   {\@nameuse{sub\@captype name}}%
                   {\@nameuse{thesub\@captype}}%
                   {#3}
\LWR@traceinfo{sfsubcap C1}% new
%  }%
%              \hss

%  }
          }
        }%
  \egroup
\LWR@startpars% new
}
```

\caption@@@make {⟨*caption label*⟩} {⟨*caption text*⟩}

```
\renewcommand\caption@@@make[2]{%
\LWR@startpars% new
  \sbox\@tempboxa{#1}%
  \ifdim\wd\@tempboxa=\z@
    \let\caption@lsep\relax
  \fi
  \caption@ifempty{#2}{%
    \let\caption@lsep\@empty
    \let\caption@tfmt\@firstofone
  }%
%   \@setpar{\@@par\caption@@par}\caption@@par
\renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}% new
  \caption@applyfont
  \caption@fmt
    {\ifcaption@star\else
       \begingroup
         \captionlabelfont
         #1%
       \endgroup
     \fi}%
    {\ifcaption@star\else
       \begingroup
         \caption@iflf\captionlabelfont
         \relax\caption@lsep
       \endgroup
     \fi}%
    {{\captiontextfont
      \caption@ifstrut
        {\vrule\@height\ht\strutbox\@width\z@}%
        {}%
      \nobreak\hskip\z@skip % enable hyphenation
      \caption@tfmt{#2}
    \LWR@ensuredoingapar% new
      \caption@ifstrut
        {\ifhmode\@finalstrut\strutbox\fi}%
        {}%
      \par}}
\LWR@stoppars% new
}
```

\subfloat@label Patches for \sf@sub@label:

```
\def\subfloat@label{%
\LWR@ensuredoingapar% new
  \@ifnextchar(%   %) match left parenthesis
    {\sf@sub@label}
    {\sf@sub@label(Sub\@captype\space
                   \@ifundefined{thechapter}{}{\@nameuse{thechapter}\space}%
                   \@nameuse{p@sub\@captype}%
                   \@nameuse{thesub\@captype}.)}}
```

Patches for `\subref`.

`\sf@subref` {⟨*label*⟩}

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
\renewcommand{\sf@subref}[1]{%
\LWR@subnewref{#1}{sub@#1}%
}
```

`\sf@@subref` {⟨*label*⟩}

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
\renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

`\@newsubfloat` [⟨*keys/values*⟩] {⟨*float name*⟩}

```
\LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat

\def\@newsubfloat[#1]#2{%
\LWR@orig@newsubfloat[#1]{#2}%
\renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
}
```

Pre-defined for figures and tables:

`\l@subfigure` {⟨*text*⟩} {⟨*pagenum*⟩}

```
\renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
```

`\l@subtable` {⟨*text*⟩} {⟨*pagenum*⟩}

```
\renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

---

File 160 **`lwarp-supertabular.sty`**

## § 241 Package **supertabular**

(*Emulates or patches code by* Johannes Braams, Theo Jurriens.)

Pkg `supertabular` supertabular is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{supertabular}
```

⚠ misplaced alignment alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.7.

```
\newcommand{\LWRST@firsthead}{}

\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRST@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}
\newcommand{\tabletail}[1]{}

\newcommand{\LWRST@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRST@lasttail{#1}%
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRST@caption{\caption[#1]{#2}}%
}

\let\topcaption\tablecaption
\let\bottomcaption\tablecaption

```

```
\newcommand*{\LWRST@caption}{}

\newcommand*{\shrinkheight}[1]{}

\NewDocumentEnvironment{supertabular}{s o m}
{
\LWR@traceinfo{supertabular}
\table
\LWRST@caption
\begin{tabular}{#3}
\TabularMacro\ifdefvoid{\LWRST@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
}
{%
\ifdefvoid{\LWRST@lasttail}%
{}%
{%
\TabularMacro\ResumeTabular%
\LWRST@lasttail%
}%
\end{tabular}
\endtable
\LWR@traceinfo{supertabular done}
}

\NewDocumentEnvironment{mpsupertabular}{s o m}
{\minipage{\linewidth}\supertabular{#3}}
{\endsupertabular\endminipage}
```

File 161 `lwarp-syntonly.sty`

## § 242 Package **syntonly**

(*Emulates or patches code by* Frank Mittelbach, Rainer Schöpf.)

Pkg `syntonly` Emulated.

**for HTML output:** Discard all options for lwarp-syntonly:

```
\LWR@ProvidesPackageDrop{syntonly}
```

```
\newif\ifsyntax@
\syntax@false

\newcommand*{\syntaxonly}{}

\@onlypreamble\syntaxonly
```

File 162 `lwarp-tabls.sty`

## § 243 Package **tabls**

(*Emulates or patches code by* Donald Arseneau.)

Pkg `tabls` tabls is emulated. `\LWR@hline` is used to handle the optional argument when tabls is loaded.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{tabls}
```

```
\newdimen\tablinesep
```

```
\newdimen\arraylinesep
\newdimen\extrarulesep
```

## File 163 lwarp-tabularx.sty

## § 244 Package tabularx

(*Emulates or patches code by* DAVID CARLISLE.)

Pkg tabularx

tabularx is emulated by lwarp.

**for HTML output:** Discard all options for lwarp-tabularx:

```
\LWR@ProvidesPackageDrop{tabularx}
```

```
\NewDocumentEnvironment{tabularx}{m o m}
{\tabular{#3}}
{\endtabular}

\NewDocumentEnvironment{tabularx*}{m o m}
{\tabular{#3}}
{\endtabular}
```

## File 164 lwarp-tabulary.sty

## § 245 Package tabulary

(*Emulates or patches code by* DAVID CARLISLE.)

Pkg tabulary

tabulary is emulated by lwarp.

**for HTML output:** Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```
\LWR@ProvidesPackageDrop{tabulary}
```

```
\NewDocumentEnvironment{tabulary}{m o m}
{\tabular{#3}}
{\endtabular}

\NewDocumentEnvironment{tabulary*}{m o m}
{\tabular{#3}}
{\endtabular}

```

```
\newdimen\tymin
\newdimen\tymax
\def\tyformat{}
```

File 165 `lwarp-textarea.sty`

# § 246 Package **textarea**

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg `textarea` textarea is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{textarea}
```

```
\newcommand\StartFromTextArea{}
\newcommand\StartFromHeaderArea{}
\newcommand*\RestoreTextArea{}
\newcommand*\ExpandTextArea[1][*]{}
\let\NCC@restoretextarea\@empty
```

File 166 `lwarp-textcomp.sty`

# § 247 Package **textcomp**

(*Emulates or patches code by* Frank Mittelbach, Robin Fairbairns, Werner Lemberg.)

Pkg `textcomp` textcomp is patched for use by lwarp.

## § 247.1 Limitations

Some textcomp symbols do not have Unicode equivalents, and thus are not supported.

⚠ Missing symbols Many textcomp symbols are not supported by many fonts. Try using more complete fonts in the css, but expect to see gaps in coverage.

## § 247.2 Package loading

**for HTML output:**

```
\LWR@ProvidesPackagePass{textcomp}
```

## § 247.3 Remembering original defintions

The following are restored for print when inside a `lateximage`:

```
\let\LWR@origtextdegree\textdegree
\let\LWR@origtextcelsius\textcelsius
\let\LWR@origtextohm\textohm
\let\LWR@origtextmu\textmu
\let\LWR@origtextlquill\textlquill
\let\LWR@origtextrquill\textrquill
\let\LWR@origtextcircledP\textcircledP
\let\LWR@origtexttwelveudash\texttwelveudash
\let\LWR@origtextthreequartersemdash\textthreequartersemdash
\let\LWR@origtextmho\textmho
\let\LWR@origtextnaira\textnaira
\let\LWR@origtextpeso\textpeso
\let\LWR@origtextrecipe\textrecipe
\let\LWR@origtextinterrobangdown\textinterrobangdown
\let\LWR@origtextpertenthousand\textpertenthousand
\let\LWR@origtextbaht\textbaht
\let\LWR@origtextdiscount\textdiscount
\let\LWR@origtextservicemark\textservicemark
\LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
\LetLtxMacro\LWR@origcapitalogonek\capitalogonek
\LetLtxMacro\LWR@origcapitalgrave\capitalgrave
\LetLtxMacro\LWR@origcapitalacute\capitalacute
\LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
\LetLtxMacro\LWR@origcapitaltilde\capitaltilde
\LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
\LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
\LetLtxMacro\LWR@origcapitalring\capitalring
\LetLtxMacro\LWR@origcapitalcaron\capitalcaron
\LetLtxMacro\LWR@origcapitalbreve\capitalbreve
\LetLtxMacro\LWR@origcapitalmacron\capitalmacron
\LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
\LetLtxMacro\LWR@origtextcircled\textcircled
```

## § 247.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

\AtBeginDocument improves support for LuaLaTeX and XeLaTeX.

### § 247.4.1 pdfLaTeX symbols

```
\AtBeginDocument{
\ifPDFTeX
\renewcommand*{\textdegree}{\HTMLentity{deg}}
\renewcommand*{\textcelsius}{\HTMLunicode{2103}}
\renewcommand*{\textohm}{\HTMLunicode{2126}}
\renewcommand*{\textmu}{\HTMLunicode{00B5}}
\renewcommand*{\textlquill}{\HTMLunicode{2045}}
\renewcommand*{\textrquill}{\HTMLunicode{2046}}
\renewcommand*{\textcircledP}{\HTMLunicode{2117}}
\renewcommand*{\texttwelveudash}{\HTMLunicode{2014}}% emdash
\renewcommand*{\textthreequartersemdash}{\HTMLunicode{2014}}% emdash
\renewcommand*{\textmho}{\HTMLunicode{2127}}
\renewcommand*{\textnaira}{\HTMLunicode{20A6}}
\renewcommand*{\textpeso}{\HTMLunicode{20B1}}
\renewcommand*{\textrecipe}{\HTMLunicode{211E}}
\renewcommand*{\textinterrobangdown}{\HTMLunicode{2E18}}
\renewcommand*{\textpertenthousand}{\HTMLunicode{2031}}
\renewcommand*{\textbaht}{\HTMLunicode{0E3F}}
\renewcommand*{\textdiscount}{\%}
\renewcommand*{\textservicemark}{\HTMLunicode{2120}}
\else
```

### § 247.4.2 XeLaTeX and LuaLaTeX symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```
\renewcommand*{\textdegree}{°}
\renewcommand*{\textcelsius}{℃}
\renewcommand*{\textohm}{Ω}
\renewcommand*{\textmu}{µ}
\renewcommand*{\textlquill}{⁅}
\renewcommand*{\textrquill}{⁆}
\renewcommand*{\textcircledP}{℗}
\renewcommand*{\texttwelveudash}{--}% emdash
\renewcommand*{\textthreequartersemdash}{--}% emdash
\renewcommand*{\textmho}{℧}
\renewcommand*{\textnaira}{₦}
\renewcommand*{\textpeso}{₱}
```

```
\renewcommand*{\textrecipe}{℞}
\renewcommand*{\textinterrobangdown}{⸘}
\renewcommand*{\textpertenthousand}{‱}
\renewcommand*{\textbaht}{฿}
\renewcommand*{\textdiscount}{\%}
\renewcommand*{\textservicemark}{}
\fi
```

## § 247.5 HTML dicritics

For HTML, Unicode diacritical marks are used:

```
\renewcommand*{\capitalcedilla}[1]{#1\HTMLunicode{0327}}
\renewcommand*{\capitalogonek}[1]{#1\HTMLunicode{0328}}
\renewcommand*{\capitalgrave}[1]{#1\HTMLunicode{0300}}
\renewcommand*{\capitalacute}[1]{#1\HTMLunicode{0301}}
\renewcommand*{\capitalcircumflex}[1]{#1\HTMLunicode{0302}}
\renewcommand*{\capitaltilde}[1]{#1\HTMLunicode{0303}}
\renewcommand*{\capitaldieresis}[1]{#1\HTMLunicode{0308}}
\renewcommand*{\capitalhungarumlaut}[1]{#1\HTMLunicode{30B}}
\renewcommand*{\capitalring}[1]{#1\HTMLunicode{30A}}
\renewcommand*{\capitalcaron}[1]{#1\HTMLunicode{30C}}
\renewcommand*{\capitalbreve}[1]{#1\HTMLunicode{306}}
\renewcommand*{\capitalmacron}[1]{#1\HTMLunicode{304}}
\renewcommand*{\capitaldotaccent}[1]{#1\HTMLunicode{307}}
```

`\textcircled` becomes a span with a rounded border:

```
\renewcommand*{\textcircled}[1]{%
\InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
}
}% AtBeginDocument
```

## § 247.6 Inside a `lateximage`

When a `lateximage` is begun:

```
\appto{\LWR@restoreorigformatting}{%
\let\textdegree\LWR@origtextdegree%
\let\textcelsius\LWR@origtextcelsius%
\let\textohm\LWR@origtextohm%
\let\textmu\LWR@origtextmu%
\let\textlquill\LWR@origtextlquill%
\let\textrquill\LWR@origtextrquill%
\let\textcircledP\LWR@origtextcircledP%
\let\texttwelveudash\LWR@origtexttwelveudash%
\let\textthreequartersemdash\LWR@origtextthreequartersemdash%
\let\textmho\LWR@origtextmho%
\let\textnaira\LWR@origtextnaira%
\let\textpeso\LWR@origtextpeso%
\let\textrecipe\LWR@origtextrecipe%
\let\textinterrobangdown\LWR@origtextinterrobangdown%
\let\textpertenthousand\LWR@origtextpertenthousand%
\let\textbaht\LWR@origtextbaht%
\let\textdiscount\LWR@origtextdiscount%
\let\textservicemark\LWR@origtextservicemark%
\LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
\LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
\LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
\LetLtxMacro\capitalacute\LWR@origcapitalacute%
\LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
\LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
\LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
\LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
\LetLtxMacro\capitalring\LWR@origcapitalring%
\LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
\LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
\LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
\LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
\LetLtxMacro\textcircled\LWR@origtextcircled%
}
```

## File 167 lwarp-textpos.sty

# § 248 Package **textpos**

(*Emulates or patches code by* Norman Gray.)

Pkg textpos

textpos is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{textpos}
```

```
\NewDocumentEnvironment{textblock}{m r()}{}{}
\NewDocumentEnvironment{textblock*}{m o r()}{}{}
\newcommand*{\TPGrid}[3][]{}
\NewDocumentCommand{\TPMargin}{s o}{}
\newcommand*{\textblockcolour}[1]{}
\newcommand*{\textblockrulecolour}[1]{}
\newcommand*{\textblockcolor}[1]{}
\newcommand*{\textblockrulecolor}[1]{}
\newcommand*{\tekstblokkulur}[1]{}
\newcommand*{\tekstblokrulekulur}[1]{}
\newlength{\TPHorizModule}
\newlength{\TPVertModule}
\newlength{\TPboxrulesize}
\newcommand{\textblocklabel}[1]{}
```

```
\newcommand*{\showtextsize}{}
\newcommand{\textblockorigin}[2]{}
```

File 168 **lwarp-theorem.sty**

# § 249 Package **theorem**

(*Emulates or patches code by* Frank Mittelbach.)

Pkg theorem theorem is patched for use by lwarp.

Table 14: Theorem package — CSS styling of theorems and proofs

**Theorem:** `<div>` of class `theorembody<theoremstyle>`

**Theorem Header:** `<span>` of class `theoremheader`

where `<theoremstyle>` is `plain`, `break`, etc.

**for HTML output:**

```
\LWR@ProvidesPackagePass{theorem}
```

## § 249.1 Remembering the theorem style

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
   \@ifundefined{th@#1}{\@warning
          {Unknown theoremstyle ‘#1’. Using ‘plain’}%
          \theorem@style{plain}%
           \renewcommand{\LWR@newtheoremstyle}{plain}% new
          }%
      {%
          \theorem@style{#1}%
          \renewcommand{\LWR@newtheoremstyle}{#1}% new
      }%
      \begingroup
        \csname th@\the\theorem@style \endcsname
      \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
\gdef\@xnthm#1#2[#3]{%
   \expandafter\@ifdefinable\csname #1\endcsname
   {%
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
    \@definecounter{#1}\@newctr{#1}[#3]%
    \expandafter\xdef\csname the#1\endcsname
      {\expandafter \noexpand \csname the#3\endcsname
       \@thmcountersep \@thmcounter{#1}}%
    \def\@tempa{\global\@namedef{#1}}%
    \expandafter \@tempa \expandafter{%
      \csname th@\the \theorem@style
            \expandafter \endcsname \the \theorem@bodyfont
     \@thm{#1}{#2}}%
    \global \expandafter \let \csname end#1\endcsname \@endtheorem
    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
   }}

\gdef\@ynthm#1#2{%
    \expandafter\@ifdefinable\csname #1\endcsname
   {
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
    \@definecounter{#1}%
    \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
    \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
     \expandafter{\csname th@\the \theorem@style \expandafter
     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
    \global \expandafter \let \csname end#1\endcsname \@endtheorem
    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
   }}

\gdef\@othm#1[#2]#3{%
  \expandafter\ifx\csname c@#2\endcsname\relax
   \@nocounterr{#2}%
  \else
   \expandafter\@ifdefinable\csname #1\endcsname
   {
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
    \expandafter \xdef \csname the#1\endcsname
     {\expandafter \noexpand \csname the#2\endcsname}%
    \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
     \expandafter{\csname th@\the \theorem@style \expandafter
     \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
    \global \expandafter \let \csname end#1\endcsname \@endtheorem
    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
    }%
  \fi}
```

## § 249.2 CSS patches

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```
\gdef\th@plain{%\normalfont\itshape
 \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
   \BlockClass{theorembody\LWR@thisthmstyle}% new
       \item[\hskip\labelsep
           \InlineClass{theoremheader}{##1\ ##2}
       ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
   \BlockClass{theorembody\LWR@thisthmstyle}% new
  \item[\hskip\labelsep
       \InlineClass{theoremheader}{##1\ ##2\ (##3)}
   ]}
}

\gdef\th@break{%\normalfont\slshape
 \def\@begintheorem##1##2{%
\LWR@forcenewpage% new
   \BlockClass{theorembody\LWR@thisthmstyle}% new
   \item[\hskip \labelsep
       \InlineClass{theoremheader}{##1\ ##2}\newline%
   ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
   \BlockClass{theorembody\LWR@thisthmstyle}% new
 \item[\hskip \labelsep
       \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
   ]}
}

\gdef\th@marginbreak{%\normalfont\slshape
 \def\@begintheorem##1##2{
\LWR@forcenewpage% new
   \BlockClass{theorembody\LWR@thisthmstyle}% new
   \item[\hskip\labelsep %
       \InlineClass{theoremheader}{##2 \qquad ##1}\newline
   ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
```

```
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep %
        \InlineClass{theoremheader}{##2 \qquad ##1\ %
        (##3)}\newline
    ]}
}

\gdef\th@changebreak{%\normalfont\slshape
  \def\@begintheorem##1##2{
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
        \InlineClass{theoremheader}{##2\ ##1}\newline
    ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
        \InlineClass{theoremheader}{ ##2\ ##1\ %
        (##3)}\newline
    ]}
}

\gdef\th@change{%\normalfont\slshape
  \def\@begintheorem##1##2{
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
        \InlineClass{theoremheader}{##2\ ##1}
    ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
        \InlineClass{theoremheader}{##2\ ##1\ (##3)}
    ]}
}

\gdef\th@margin{%\normalfont\slshape
  \def\@begintheorem##1##2{
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
        \InlineClass{theoremheader}{##2 \qquad ##1}
    ]}%
\def\@opargbegintheorem##1##2##3{%
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
    \item[\hskip\labelsep
            \InlineClass{theoremheader}{##2 \qquad ##1\ (##3)}
    ]}
}
```

Patched for CSS:

```
\gdef\@endtheorem{\endBlockClass\endtrivlist}
```

File 169 `lwarp-threeparttable.sty`

## § 250 Package **threeparttable**

(*Emulates or patches code by* Donald Arseneau.)

Pkg `threeparttable` threeparttable is emulated.

Table note are contained inside a CSS `<div>` of class `tnotes`. If enumitem is used, the note item labels are also individually highlighted with an additional CSS `<span>` of class `tnoteitemheader`, otherwise they are plain text.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{threeparttable}
```

`\LWR@printtablenote` {⟨*text*⟩}

Prints the table note item header inside a CSS class of `tnoteitemheader`.

```
\newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env `threeparttable` [⟨*alignment*⟩] To emulate threeparttable:

```
\newenvironment*{threeparttable}[1][b]{}{}
```

Env `tablenotes` [⟨*options*⟩]

```
\newenvironment*{tablenotes}[1][]
{%
\LWR@forcenewpage
\BlockClass{tnotes}%
\ltx@ifpackageloaded{enumitem}{%
\setlist[description]{format=\LWR@printtablenote}%
}{}%
\description%
}
{%
\enddescription%
\endBlockClass%
}
```

\tnote {⟨*text*⟩}

```
\newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

File 170 **lwarp-tikz.sty**

# § 251 Package **tikz**

(*Emulates or patches code by* Till Tantau.)

Pkg tikz tikz is supported.

Accept all options for lwarp-tikz:

```
\LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp changes the catcode of $ for its own use. The Ti*k*z babel library temporarily changes catcodes back to normal for Ti*k*z's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, lwarp must change $'s catcode itself.

Also see:
https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded

**for HTML output:**

```
\begin{warpHTML}
```

```
\newboolean{LWR@tikzbabel}

\@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
{\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
{\boolfalse{LWR@tikzbabel}}
```

Env tikzpicture tikzpicture environment is enclosed inside a \lateximage. May be used as-is, and its contents will be converted to an image.

```
\BeforeBeginEnvironment{tikzpicture}{%
\begin{lateximage}%
\ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
{}%
{\catcode`\$=3} % dollar sign is math shift
}

\AfterEndEnvironment{tikzpicture}{%
\end{lateximage}%
\ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
{}%
{\catcode`\$=\active}%
}
```

```
\end{warpHTML}
```

---

## File 171 `lwarp-titleps.sty`

## § 252 Package **titleps**

(*Emulates or patches code by* Javier Bezos.)

Pkg `titleps`

titleps is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titleps:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{titleps}
```

`\pagestyle` and `\thispagestyle` are already disabled in the lwarp code.

`\newpagestyle` {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\newpagestyle}{m o m}{}
```

`\renewpagestyle` {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\renewpagestyle}{m o m}{}
```

`\sethead` [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\sethead}{o o o m m m}{}
```

`\setfoot` [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\setfoot}{o o o m m m}{}
```

`\settitlemarks` * {⟨*names*⟩}

```
\NewDocumentCommand{\settitlemarks}{s m}{}
```

`\headrule`

```
\newcommand*{\headrule}{}
```

\footrule

```
\newcommand*{\footrule}{}
```

\setheadrule {⟨*length*⟩}

```
\newcommand*{\setheadrule}[1]{}
```

\setfootrule {⟨*length*⟩}

```
\newcommand*{\setfootrule}[1]{}
```

\makeheadrule

```
\newcommand*{\makeheadrule}{}
```

\makefootrule

```
\newcommand*{\makefootrule}{}
```

\setmarkboth {⟨*code*⟩}

```
\newcommand{\setmarkboth}[1]{}
```

\widenhead

```
\NewDocumentCommand{\widenhead}{s o o m m}{}
```

\bottitlemarks

```
\newcommand*{\bottitlemarks}{}
```

\toptitlemarks

```
\newcommand*{\toptitlemarks}{}
```

\firsttitlemarks

```
\newcommand*{\firsttitlemarks}{}
```

\nexttitlemarks

```
\newcommand*{\nexttoptitlemarks}{}
```

\outertitlemarks

```
\newcommand*{\outertitlemarks}{}
```

\innertitlemarks

```
\newcommand*{\innertitlemarks}{}
```

\newtitlemark * {⟨*name*⟩}

```
\NewDocumentCommand{\newtitlemark}{s m}{}
```

\pretitlemark * {⟨*section*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\pretitlemark}{s m m}{}
```

\ifsamemark {⟨*group*⟩} {⟨*command*⟩} {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\ifsamemark}[4]{}
```

\setfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}
```

\setfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}
```

\nextfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}
```

\nextfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}
```

\newmarkset {⟨*markset*⟩}

```
\newcommand{\newmarkset}[1]{}
```

\newextramark * {⟨*markset*⟩} {⟨*macro-name*⟩}

```
\NewDocumentCommand{\newextramarkset}{s m m}{}
```

\botextramarks {⟨*markset*⟩}

```
\newcommand{\botextramarks}[1]{}
```

`\topextramarks` {⟨*markset*⟩}

```
\newcommand{\topextramarks}[1]{}
```

`\firstextramarks` {⟨*markset*⟩}

```
\newcommand{\firstextramarks}[1]{}
```

`\nextextramarks` {⟨*markset*⟩}

```
\newcommand{\nexttopextramarks}[1]{}
```

`\outerextramarks` {⟨*markset*⟩}

```
\newcommand{\outerextramarks}[1]{}
```

`\innerextramarks` {⟨*markset*⟩}

```
\newcommand{\innerextramarks}[1]{}
```

---

File 172 **`lwarp-titleref.sty`**

## § 253 Package **titleref**

Pkg `titleref` titleref is superceded by hyperref and nameref.

**for HTML output:**

```
\LWR@loadnever{titleref}{hyperref and nameref}
```

---

File 173 **`lwarp-titlesec.sty`**

## § 254 Package **titlesec**

(*Emulates or patches code by* Javier Bezos.)

Pkg `titlesec` titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{titlesec}
```

`\titlelabel` {⟨*label-format*⟩}

```
\newcommand*{\titlelabel}[1]{}
```

`\titleformat*` {⟨*command*⟩} {⟨*format*⟩}

`\titleformat` {⟨*command*⟩} [⟨*shape*⟩] {⟨*format*⟩} {⟨*label*⟩} {⟨*sep*⟩} {⟨*begfore*⟩} [⟨*after*⟩]

```
\newcommand\titleformat{%
  \@ifstar{\ttl@format@s}%
          {\ttl@format@i}}
\newcommand{\ttl@format@s}[1]{}
\NewDocumentCommand{\ttl@format@i}{m o m m m m o}{}
```

`\chaptertitlename`

```
\@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
\newcommand\chaptertitlename{\@chapapp}
```

`\titlespacing` * {⟨*command*⟩} {⟨*left*⟩} {⟨*before*⟩} {⟨*after*⟩} [⟨*right*⟩]

```
\NewDocumentCommand{\titlespacing}{s m m m m o}{}
```

`\filright`

```
\newcommand*{\filright}{}
```

`\filcenter`

```
\newcommand*{\filcenter}{}
```

`\filleft`

```
\newcommand*{\filleft}{}
```

`\fillast`

```
\newcommand*{\fillast}{}
```

`\filinner`

```
\newcommand*{\filinner}{}
```

`\filouter`

```
\newcommand*{\filouter}{}
```

`\wordsep`

```
\newcommand\wordsep{\fontdimen\tw@\font \@plus
  \fontdimen\thr@@\font \@minus \fontdimen4\font}
```

`\titleline` * [⟨*align*⟩] {⟨*material*⟩}

```
\NewDocumentCommand{\titleline}{s o m}{}
```

`\titlerule` [⟨*height*⟩]

```
\providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
\newcommand*{\ttl@rule}[1][]{}
\newcommand*{\ttl@row}[2][]{}
```

`\iftitlemeasuring` {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\iftitlemeasuring}[2]{#2}
```

`\assignpagestyle` {⟨*command*⟩} {⟨*pagestyle*⟩}

```
\newcommand{\assignpagestyle}[2]{#2}
```

`\titleclass` {⟨*name*⟩} [⟨*startlevel*⟩] {⟨*class*⟩} [⟨*cmd*⟩]

```
\NewDocumentCommand{\titleclass}{m o m o}
```

File 174 **`lwarp-titletoc.sty`**

## § 255 Package **titletoc**

(*Emulates or patches code by* Javier Bezos.)

Pkg `titletoc` titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{titletoc}
```

`\dottedcontents` {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*label*⟩} {⟨*leader*⟩}

```
\NewDocumentCommand{\dottedcontents}{m o m m m}{}
```

`\titlecontents` * {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*numbered*⟩} {⟨*numberless*⟩} {⟨*filler*⟩} [⟨*below or begin*⟩] [⟨*separator*⟩] [⟨*end*⟩]

```
\newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
\NewDocumentCommand{\ttl@tcstar}{m o m m m m o o o}{}
```

```
\NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}
```

`\contentsmargin` [⟨*correction*⟩] {⟨*right*⟩}

```
\newcommand{\contentsmargin}[2][]{}
```

`\thecontentslabel`

```
\newcommand*{\thecontentslabel}{thecontentslabel}
```

`\thecontentspage`

```
\newcommand*{\thecontentspage}{thecontentspage}
```

`\contentslabel` [⟨*format*⟩] {⟨*space*⟩}

```
\newcommand{\contentslabel}[2][]{\thecontentslabel}
```

`\contentspage` [⟨*format*⟩]

```
\newcommand{\contentspage}[1][]{\thecontentspage}
```

`\contentspush` {⟨*text*⟩}

```
\newcommand{\contentspush}[1]{}
```

`\contentsuse` {⟨*name*⟩} {⟨*text*⟩}

```
\newcommand{\contentsuse}[2]{}
```

`\startcontents` [⟨*name*⟩]

```
\newcommand*{\startcontents}[1][]{}
```

`\stopcontents` [⟨*name*⟩]

```
\newcommand*{\stopcontents}[1][]{}
```

`\resumecontents` [⟨*name*⟩]

```
\newcommand*{\resumecontents}[1][]{}
```

`\printcontents` [⟨*name*⟩] {⟨*prefix*⟩} {⟨*start*⟩} {⟨*code*⟩}

```
\newcommand{\printcontents}[4][]{}
```

`\startlist` [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\startlist}[2][]{}
```

`\stoplist` [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\stoplist}[2][]{}
```

`\resumelist` [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\resumelist}[2][]{}
```

`\printlist` [⟨*name*⟩] {⟨*list*⟩} {⟨*prefix*⟩} {⟨*code*⟩}

```
\newcommand{\printlist}[4][]{}
```

File 175 **`lwarp-titling.sty`**

## § 256 Package **titling**

(*Emulates or patches code by* Peter Wilson.)

Pkg `titling`

package support
⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

`\published` and `\subtitle`

If using the titling package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 55.7.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

**for HTML output:**

```
\LWR@ProvidesPackagePass{titling}
```

`\@bsmtitlempty` Patch `\@bsmtitlempty`:

```
\let\LWR@orig@bsmtitlempty\@bsmtitlempty
\renewcommand*{\@bsmtitlempty}{%
\LWR@orig@bsmtitlempty%
}
```

`\keepthetitle` Patch `\keepthetitle`:

```
\let\LWR@origkeepthetitle\keepthetitle
\renewcommand*{\keepthetitle}{%
\LWR@orig@keepthetitle%
}
```

\killtitle Patch \killtitle:

```
\let\LWR@origkilltitle\killtitle
\renewcommand*{\killtitle}{%
\LWR@orig@killtitle%
}
```

Env titlingpage

```
\renewenvironment*{titlingpage}
{%
```

Start an HTML titlepage div:

```
\LWR@printpendingfootnotes
\begin{titlepage}
```

Prepare for a custom version of \maketitle inside the titlingpage:

```
\LWR@maketitlesetup
\let\maketitle\LWR@titlingmaketitle
}
{
```

At the end of the environment, end the HTML titlepage div:

```
\end{titlepage}
}
```

Patch the pre/post title/author/date to add HTML tags, then initilize:

```

\pretitle{}
\posttitle{}

\preauthor{}
\postauthor{}

\predate{}
\postdate{}
```

\LWR@maketitlesetup Patches \thanks macros.

```
\renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\def\@makefnmark{\textsuperscript{\@thefnmark}}
```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\makethanksmark \LWR@orighspace{1in}
```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

Print the text:

```
##1%
}%
}
```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
\begin{titlepage}
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Close the HTML titlepage div:

```
\end{titlepage}
```

Reset the footnote counter:

```
\@bscontmark
}
```

\@maketitle Typesets the title, etc. Patched for HTML.

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \maketitlehooka
    {
        \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
        \@bspretitle \@title \@bsposttitle
        \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    }
    \maketitlehookb
    {
        \begin{BlockClass}{author}
        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
        \@bspreauthor \@author  \@bspostauthor
        \end{BlockClass}
        \end{BlockClass}
    }
    \maketitlehookc
    {
        \begin{BlockClass}{titledate}
        \@bspredate \@date \@bspostdate
        \end{BlockClass}
    }
    \maketitlehookd
}
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
\renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
\@thanks
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any `\thanks` footnotes:

```
\@thanks
```

Reset the footnote counter:

```
\@bscontmark
}
```

`\thanksmarkseries` {⟨*series*⟩}

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
\renewcommand{\thanksmarkseries}[1]{%
\def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
}
```

Set default titlepage thanks footnote marks. See section 55.6.

```
\if@titlepage
   \thanksmarkseries{arabic}
\else
   \thanksmarkseries{fnsymbol}
\fi
```

File 176 **lwarp-tocbasic.sty**

## § 257 Package **tocbasic**

(*Emulates or patches code by* Markus Kohm.)

Pkg tocbasic tocbasic is patched for use by lwarp.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

**for HTML output:**

```
\LWR@ProvidesPackagePass{tocbasic}

\DeclareDocumentCommand{\usetocbasicnumberline}{o}{}
\DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{}
\DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{}
\DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{}
\DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{}
\DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{}
\DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{}
\DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{}
```

File 177 **lwarp-tocbibind.sty**

## § 258 Package **tocbibind**

(*Emulates or patches code by* Peter Wilson.)

Pkg tocbibind tocbibind is patched for use by lwarp.

Opt IndexLanguage The lwarp package takes an option IndexLanguage=english to set the language used by xindy. This is passed to xindy using its -L option, and is used for both index and glossary generation.

⚠ tocloft & other packages If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its titles option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

placement and TOC options

An index may be placed inline with other HTML text, or on its own HTML page:

**Inline, with a manual TOC entry:**

A commonly-used method to introduce an index in a LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

**On its own HTML page, with a manual TOC entry:**

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

**Inline, with an automatic TOC entry:**

Pkg tocbibind

The tocbibind package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

**On its own HTML page, with an automatic TOC entry:**

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt tocbibind numindex
numbered index section

Use the tocbibind numindex option to generate a numbered index. Without this option, the index heading has no number.

**for HTML output:**

```
\let\simplechapterdelim\relax

\LWR@ProvidesPackagePass{tocbibind}
```

```
\renewenvironment{theindex}%
{%
    \if@bibchapter
```

```
        \if@donumindex
            \chapter{\indexname}
        \else
          \if@dotocind
            \chapter*{\indexname}
            \addcontentsline{toc}{chapter}{\indexname}
          \else
            \chapter*{\indexname}
          \fi
        \fi
     \else
        \if@donumindex
            \section{\indexname}
        \else
          \if@dotocind
            \section*{\indexname}
            \addcontentsline{toc}{\@tocextra}{\indexname}
          \else
            \section*{\indexname}
          \fi
        \fi
     \fi
\let\item\LWR@indexitem%
\let\subitem\LWR@indexsubitem%
\let\subsubitem\LWR@indexsubsubitem%
}{}
```

The following code is shared by anonchap.

```
\renewcommand{\simplechapter}[1][\@empty]{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\renewcommand{\restorechapter}{%
\let\@chapcntformat\@seccntformat%
}
```

## File 178 `lwarp-tocloft.sty`

# § 259 Package **tocloft**

(*Emulates or patches code by* Peter Wilson.)

Pkg `tocloft`

tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

Discard all options for lwarp-tocloft:

**for HTML output:**

```
\LWR@ProvidesPackageDrop{tocloft}
```

\tocloftpagestyle {⟨*style*⟩}

```
\newcommand{\tocloftpagestyle}[1]{}
```

\cftmarktoc

```
\newcommand*{\cftmarktoc}{}
```

\cfttoctitlefont

```
\newcommand*{\cfttoctitlefont}{}
```

\cftaftertoctitle

```
\newcommand*{\cftaftertoctitle}{}
```

```
\newlength{\cftbeforetoctitleskip}
\newlength{\cftaftertoctitleskip}
```

\cftmarklof

```
\newcommand*{\cftmarklof}{}
```

\cftloftitlefont

```
\newcommand*{\cftloftitlefont}{}
```

\cftafterloftitle

```
\newcommand*{\cftafterloftitle}{}
```

```
\newlength{\cftbeforeloftitleskip}
\newlength{\cftafterloftitleskip}
```

\cftmarklot

```
\newcommand*{\cftmarklot}{}
```

\cftlottitlefont

```
\newcommand*{\cftlottitlefont}{}
```

\cftafterlottitle

```
\newcommand*{\cftafterlottitle}{}
```

```
\newlength{\cftbeforelottitleskip}
\newlength{\cftafterlottitleskip}
```

\cftdot

```
\newcommand*{\cftdot}{.}
```

\cftdotsep

```
\providecommand*{\cftdotsep}{1}
```

\cftnodots

```
\newcommand*{\cftnodots}{5000}
```

\cftdotfill {⟨*sep*⟩}

```
\providecommand{\cftdotfill}[1]{}
```

\cftsetpnumwidth {⟨*length*⟩}

```
\newcommand*{\cftsetpnumwidth}[1]{}
```

\cftsetrmarg {⟨*length*⟩}

```
\newcommand*{\cftsetrmarg}[1]{}
```

\cftpnumalign {⟨*alignment*⟩}

```
\newcommand*{\cftpnumalign}[1]{}
```

```
\newlength{\cftparskip}
```

```
\newlength{\cftbeforepartskip}
\newlength{\cftpartindent}
\newlength{\cftpartnumwidth}
\newcommand*{\cftpartfont}{}
```

```
\newcommand*{\cftpartpresnum}{}
\newcommand*{\cftpartaftersnum}{}
\newcommand*{\cftpartaftersnumb}{}
\newcommand*{\cftpartleader}{}
\newcommand*{\cftpartdotsep}{1}
\newcommand*{\cftpartpagefont}{}
\newcommand*{\cftpartafterpnum}{}
```

```
\newlength{\cftbeforechapskip}
\newlength{\cftchapindent}
\newlength{\cftchapnumwidth}
\newcommand*{\cftchapfont}{}
\newcommand*{\cftchappresnum}{}
\newcommand*{\cftchapaftersnum}{}
\newcommand*{\cftchapaftersnumb}{}
\newcommand*{\cftchapleader}{}
\newcommand*{\cftchapdotsep}{1}
\newcommand*{\cftchappagefont}{}
\newcommand*{\cftchapafterpnum}{}
```

```
\newlength{\cftbeforesecskip}
\newlength{\cftsecindent}
\newlength{\cftsecnumwidth}
\newcommand*{\cftsecfont}{}
\newcommand*{\cftsecpresnum}{}
\newcommand*{\cftsecaftersnum}{}
\newcommand*{\cftsecaftersnumb}{}
\newcommand*{\cftsecleader}{}
\newcommand*{\cftsecdotsep}{1}
\newcommand*{\cftsecpagefont}{}
\newcommand*{\cftsecafterpnum}{}
```

```
\newlength{\cftbeforesubsecskip}
\newlength{\cftsubsecindent}
\newlength{\cftsubsecnumwidth}
\newcommand*{\cftsubsecfont}{}
\newcommand*{\cftsubsecpresnum}{}
\newcommand*{\cftsubsecaftersnum}{}
\newcommand*{\cftsubsecaftersnumb}{}
\newcommand*{\cftsubsecleader}{}
\newcommand*{\cftsubsecdotsep}{1}
\newcommand*{\cftsubsecpagefont}{}
\newcommand*{\cftsubsecafterpnum}{}
```

```
\newlength{\cftbeforesubsubsecskip}
\newlength{\cftsubsubsecindent}
\newlength{\cftsubsubsecnumwidth}
\newcommand*{\cftsubsubsecfont}{}
\newcommand*{\cftsubsubsecpresnum}{}
```

```
\newcommand*{\cftsubsubsecaftersnum}{}
\newcommand*{\cftsubsubsecaftersnumb}{}
\newcommand*{\cftsubsubsecleader}{}
\newcommand*{\cftsubsubsecdotsep}{1}
\newcommand*{\cftsubsubsecpagefont}{}
\newcommand*{\cftsubsubsecafterpnum}{}

\newlength{\cftbeforeparaskip}
\newlength{\cftparaindent}
\newlength{\cftparanumwidth}
\newcommand*{\cftparafont}{}
\newcommand*{\cftparapresnum}{}
\newcommand*{\cftparaaftersnum}{}
\newcommand*{\cftparaaftersnumb}{}
\newcommand*{\cftparaleader}{}
\newcommand*{\cftparadotsep}{1}
\newcommand*{\cftparapagefont}{}
\newcommand*{\cftparaafterpnum}{}

\newlength{\cftbeforesubparaskip}
\newlength{\cftsubparaindent}
\newlength{\cftsubparanumwidth}
\newcommand*{\cftsubparafont}{}
\newcommand*{\cftsubparapresnum}{}
\newcommand*{\cftsubparaaftersnum}{}
\newcommand*{\cftsubparaaftersnumb}{}
\newcommand*{\cftsubparaleader}{}
\newcommand*{\cftsubparadotsep}{1}
\newcommand*{\cftsubparapagefont}{}
\newcommand*{\cftsubparaafterpnum}{}

\newlength{\cftbeforefigskip}
\newlength{\cftfigindent}
\newlength{\cftfignumwidth}
\newcommand*{\cftfigfont}{}
\newcommand*{\cftfigpresnum}{}
\newcommand*{\cftfigaftersnum}{}
\newcommand*{\cftfigaftersnumb}{}
\newcommand*{\cftfigleader}{}
\newcommand*{\cftfigdotsep}{1}
\newcommand*{\cftfigpagefont}{}
\newcommand*{\cftfigafterpnum}{}

\newlength{\cftbeforesubfigskip}
\newlength{\cftsubfigindent}
\newlength{\cftsubfignumwidth}
\newcommand*{\cftsubfigfont}{}
\newcommand*{\cftsubfigpresnum}{}
\newcommand*{\cftsubfigaftersnum}{}
```

```
\newcommand*{\cftsubfigaftersnumb}{}
\newcommand*{\cftsubfigleader}{}
\newcommand*{\cftsubfigdotsep}{1}
\newcommand*{\cftsubfigpagefont}{}
\newcommand*{\cftsubfigafterpnum}{}

\newlength{\cftbeforetabskip}
\newlength{\cfttabindent}
\newlength{\cfttabnumwidth}
\newcommand*{\cfttabfont}{}
\newcommand*{\cfttabpresnum}{}
\newcommand*{\cfttabaftersnum}{}
\newcommand*{\cfttabaftersnumb}{}
\newcommand*{\cfttableader}{}
\newcommand*{\cfttabdotsep}{1}
\newcommand*{\cfttabpagefont}{}
\newcommand*{\cfttabafterpnum}{}

\newlength{\cftbeforesubtabskip}
\newlength{\cftsubtabindent}
\newlength{\cftsubtabnumwidth}
\newcommand*{\cftsubtabfont}{}
\newcommand*{\cftsubtabpresnum}{}
\newcommand*{\cftsubtabaftersnum}{}
\newcommand*{\cftsubtabaftersnumb}{}
\newcommand*{\cftsubtableader}{}
\newcommand*{\cftsubtabdotsep}{1}
\newcommand*{\cftsubtabpagefont}{}
\newcommand*{\cftsubtabafterpnum}{}

\newcommand{\cftsetindents}[3]{}

\newcommand{\pagenumbersoff}[1]{}
\newcommand{\pagenumberson}[1]{}
```

`\newlistentry` [⟨*within*⟩] {⟨*counter*⟩} {⟨*ext*⟩} {⟨*level-1*⟩}

```
\NewDocumentCommand{\newlistentry}{o m m m}
{
\IfValueTF{#1}
{
    \newcounter{#2}[#1]
    \expandafter\edef\csname the#2\endcsname{
        \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}
    }
}
{\newcounter{#2}}
\@namedef{l@#2}##1##2{\hypertocfloat{1}{#2}{#3}{##1}{##2}}
\expandafter\newlength\csname cftbefore#2skip\endcsname
\expandafter\newlength\csname cft#2indent\endcsname
\expandafter\newlength\csname cft#2numwidth\endcsname
\@namedef{cft#2font}{}
\@namedef{cft#2presnum}{}
\@namedef{cft#2aftersnum}{}
\@namedef{cft#2aftersnumb}{}
\@namedef{cft#2leader}{}
\@namedef{cft#2dotsep}{1}
\@namedef{cft#2pagefont}{}
\@namedef{cft#2afterpnum}{}
\@namedef{toclevel@#2}{#4}
\@namedef{cft#2fillnum}##1{}
}
```

`\newlistof` [⟨*within*⟩] {⟨*type*⟩} {⟨*ext*⟩} {⟨*listofname*⟩}

Emulated through the `\newfloat` mechanism.

```
\NewDocumentCommand{\newlistof}{o m m m}
{%
\IfValueTF{#1}
{\newlistentry[#1]{#2}{#3}{0}}
{\newlistentry{#2}{#3}{0}}
\@namedef{ext@#2}{#3}
\newcounter{#3depth}
\setcounter{#3depth}{1}
\@namedef{cftmark#3}{}
\@namedef{listof#2}{\listof{#2}{#4}}
\@namedef{@cftmake#3title}{}
\expandafter\newlength\csname cftbefore#3titleskip\endcsname
\expandafter\newlength\csname cftafter#3titleskip\endcsname
\@namedef{cft#3titlefont}{}
\@namedef{cftafter#3title}{}
\@namedef{cft#3prehook}{}
\@namedef{cft#3posthook}{}
}
```

`\cftchapterprecis` {⟨*text*⟩}

```
\newcommand{\cftchapterprecis}[1]{%
  \cftchapterprecishere{#1}
  \cftchapterprecistoc{#1}}
\newcommand{\cftchapterprecishere}[1]{%
  \begin{quote}\textit{#1}\end{quote}}
\newcommand{\cftchapterprecistoc}[1]{
  \addtocontents{toc}{%
  {
     \protect\begin{quote}#1\protect\end{quote}}
  }
}
```

File 179 **lwarp-tocstyle.sty**

## § 260 Package **tocstyle**

Pkg tocstyle tocstyle is ignored.

⚠ Not yet tested! Please send bug reports!

**for HTML output:**

```
\LWR@ProvidesPackageDrop{tocstyle}
```

```
\newcommand*{\usetocstyle}[2][]{}
\newcommand*{\deactivatetocstyle}[1][]{}
\newcommand*{\reactivatetocstyle}[1][]{}
\NewDocumentCommand{\settocfeature}{o o m m}{}
\NewDocumentCommand{\settocstylefeature}{o m m}{}
\NewDocumentCommand{\newtocstyle}{o o m m}{}
\newcommand*{\aliastoc}[2]{}
\newcommand*{\showtoc}[2][]{}
\newcommand{\iftochasdepth}[4]{}
```

File 180 **lwarp-todo.sty**

## § 261 Package **todo**

(*Emulates or patches code by* Federico Garcia.)

Pkg todo todo is patched for use by lwarp.

**for HTML output:**

```
\LWR@ProvidesPackagePass{todo}
```

```
\renewcommand\todoitem[2]{%
    \refstepcounter{todo}%
    \item[%
        \HTMLunicode{2610} \quad
        \ref{todopage:\thetodo}
     ] : {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
    \label{todolbl:\thetodo}%
}%

\renewcommand\doneitem[2]{%
    \stepcounter{todo}%
    \item[%
```

```
        \HTMLunicode{2611} \quad
        \ref{todopage:\thetodo}
    ] \@nameuse{@done\the\c@todo}:
        {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
}

\xpatchcmd{\@displaytodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displaytodo.}}

\xpatchcmd{\@displayfulltodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displayfulltodo.}}

\patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch todoenv.}}

\patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch astodos.}}

\AtBeginDocument{
\crefname{todo}{todo}{todos}
\Crefname{todo}{Todo}{Todos}
}
```

File 181 **lwarp-todonotes.sty**

## § 262 Package **todonotes**

(*Emulates or patches code by* Henrik Skov Midtiby.)

Pkg todonotes

todonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

**for HTML output:**

```
\LWR@ProvidesPackagePass{todonotes}
```

```
\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}


```

```
\renewcommand{\@todonotes@drawMarginNoteWithLine}{
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
        {\@todonotes@author:\,}%
        \fi%
        \@todonotes@text%
    }%
}

\renewcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

\renewcommand{\@todonotes@drawLineToRightMargin}{}

\renewcommand{\@todonotes@drawLineToLeftMargin}{}

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

```

```
\LetLtxMacro\LWRTODONOTES@orig@todo\@todo

\RenewDocumentCommand{\@todo}{o m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\IfValueTF{#1}{%
    \LWRTODONOTES@orig@todo[#1]{#2}%
}{%
    \LWRTODONOTES@orig@todo{#2}%
}
\endgroup%
}

\fi% \if@todonotes@disabled
```

## File 182 `lwarp-transparent.sty`

# § 263 Package **transparent**

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `transparent` Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

⚠ Not XƎLaTeX! Note that transparent does not work with XƎLaTeX.

**for HTML output:** Discard all options for lwarp-transparent:

```
\LWR@ProvidesPackageDrop{transparent}
```

```
\newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}

\newcommand*{\texttransparent}[2]{%
\begingroup%
\transparent{#1}%
\InlineClass[opacity: #1]{transparent}{#2}%
\endgroup%
}
```

## File 183 `lwarp-trivfloat.sty`

# § 264 Package **trivfloat**

(*Emulates or patches code by* Joseph Wright.)

Pkg trivfloat

trivfloat is forced to use the built-in lwarp emulation for floats.

Discard all options for lwarp-trivfloat. This tells trivfloat not to use floatrow or memoir.

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
\LWR@ProvidesPackageDrop{trivfloat}
\LWR@origRequirePackage{trivfloat}
```

\tfl@chapter@fix

Nullified at the beginning of the document. Is used by trivfloat to correct float chapter numbers, but is not needed for lwarp.

**for HTML output:**

```
\begin{warpHTML}
```

```
\AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
```

```
\end{warpHTML}
```

## § 264.1 Combining \newfloat, \trivfloat, and algorithmicx

**for HTML & PRINT:**

```
\begin{warpall}
```

For both print and HTML output:

⚠ When using float, trivfloat, or algorithmicx at the same time, be aware of conflicting file usage. algorithmicx uses .loa. trivfloat by default starts with .loa and goes up for additional floats, skipping .lof and .lot.

⚠ When using \newfloat, be sure to manually assign higher letters to the \newfloat files to avoid .loa used by algorithmicx, and any files used by trivfloat. Also avoid using .lof and .lot.

⚠ When using \trivfloat, you may force it to avoid conflicting with algorithmicx by starting trivfloat's file extensions with .lob:

```
\makeatletter
\setcounter{tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter
```

```
\end{warpall}
```

File 184 **lwarp-typearea.sty**

# § 265 Package typearea

(*Emulates or patches code by* Markus Kohm.)

Pkg typearea

typearea is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{typearea}
```

```
\DeclareDocumentCommand{\typearea}{o m}{}
\DeclareDocumentCommand{\recalctypearea}{}{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\areaset}{o m m}{}
\DeclareDocumentCommand{\activateareas}{}{}
\DeclareDocumentCommand{\storeareas}{m}{}
\DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}
\DeclareDocumentCommand{\AfterSettingArea}{s m}{}
```

File 185 **lwarp-ulem.sty**

# § 266 Package ulem

(*Emulates or patches code by* Donald Arseneau.)

Pkg ulem

Emulated.

**for HTML output:**

Original lwarp definitions:

```
\LetLtxMacro\LWR@ulemorigemph\emph
\LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using css:

```
\NewDocumentCommand{\uline}{+m}{%
\LWR@HTMLtextstyle%
```

```
    {text-decoration:underline;text-decoration-skip}%
    {uline}{#1}%
}

\NewDocumentCommand{\uuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;text-decoration-skip;%
        text-decoration-style:double%
    }%
    {uuline}{#1}%
}

\NewDocumentCommand{\uwave}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;text-decoration-skip;%
        text-decoration-style:wavy%
    }%
    {uwave}{#1}%
}

\NewDocumentCommand{\sout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {sout}{#1}%
}

\NewDocumentCommand{\xout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {xout}{#1}%
}

\NewDocumentCommand{\dashuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip;%
        text-decoration-style:dashed%
    }%
    {dashuline}{#1}%
}

\NewDocumentCommand{\dotuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip;%
        text-decoration-style:dotted%
    }%
    {dotuline}{#1}%
}
```

Nullified parameters:

```
\NewDocumentCommand{\ULthickness}{}{}
\newlength{\ULdepth}
```

Nullified/emulated macros:

```
\NewDocumentCommand{\markoverwith}{m}{}
\NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
\NewDocumentCommand{\useunder}{m m m}{%
\relax%
\ifx\relax#3\relax\else % argumentative command
    \def#3{#1}\MakeRobust{#3}\fi
}
```

Triggered by package options, also available for the users:

```
\newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
\newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
\ULforem% default
```

Package options:

```
\DeclareOption{normalem}{\normalem}
\DeclareOption{ULforem}{\ULforem}
\DeclareOption{normalbf}{}
\DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}
```

Emulate the original package:

```
\LWR@ProvidesPackageDrop{ulem}
```

File 186 **`lwarp-upref.sty`**

## § 267 Package **upref**

Pkg `upref` Ignored.

**for HTML output:**

Discard all options for lwarp-upref:

```
\LWR@ProvidesPackageDrop{upref}
```

---

File 187 **lwarp-verse.sty**

# § 268 Package **verse**

(*Emulates or patches code by* Peter Wilson.)

Pkg verse

verse is supported and patched by lwarp.

**for HTML output:** Pass all options for lwarp-verse:

```
\LWR@ProvidesPackagePass{verse}
```

\attrib

The documentation for the verse and memoir packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. lwarp provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len \leftskip
Len \leftmargini
Len \TMLvleftskip
Len \TMLleftmargini

These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

Horizontal spacing relies on `pdftotext`’s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

Env verse

The `verse` environment will be placed inside a HTML `<pre>`.

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verse.}
```

At the beginning of the verse environment:

```
\AtBeginEnvironment{verse}
{%
```

Use the original list environment inside a <pre> to attempt to preserve formatting.

```
\LWR@restoreoriglists%
```

Pkg verse
Pkg memoir
\flagverse
Len \leftskip

The verse or memoir packages can place stanza numbers to the left with their \flagverse command. Do not allow them to go into the left margin, which would cause pdfcrop to crop the entire page further to the left:

```
\ifdef{\vleftskip}{%
\setlength{\vleftskip}{\HTMLvleftskip}
\setlength{\leftmargini}{\HTMLleftmargini}
}{}
\LWR@forcenewpage
\LWR@atbeginverbatim{verse}
\unskip\LWR@origvspace{-\baselineskip}
}
```

After the end of the verse environment, which places the <pre> tag at the regular left margin:

```
\AtEndEnvironment{verse}{
\LWR@afterendverbatim
}
```

Patch to place poemtitle inside an HTML <span> of class poemtitle:

```
\ifdef{\poemtitle}{
\DeclareDocumentCommand{\@vstypeptitle}{m}{%
   \vspace{\beforepoemtitleskip}%
   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
   \vspace{\afterpoemtitleskip}%
 }
}{}

\LWR@traceinfo{Finished patching verse.}
}% AfterEndPreamble
```

File 188 **lwarp-wallpaper.sty**

## § 269 Package **wallpaper**

(*Emulates or patches code by* Michael H.F. Wilkinson.)

Pkg wallpaper

wallpaper is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{wallpaper}
```

```
\newcommand*{\CenterWallPaper}[2]{}
\newcommand*{\ThisCenterWallPaper}[2]{}
\newcommand*{\TileWallPaper}[3]{}
\newcommand*{\ThisTileWallPaper}[3]{}
\newcommand*{\TileSquareWallPaper}[2]{}
\newcommand*{\ThisTileSquareWallPaper}[2]{}
\newcommand*{\ULCornerWallPaper}[2]{}
\newcommand*{\ThisULCornerWallPaper}[2]{}
\newcommand*{\LLCornerWallPaper}[2]{}
\newcommand*{\ThisLLCornerWallPaper}[2]{}
\newcommand*{\URCornerWallPaper}[2]{}
\newcommand*{\ThisURCornerWallPaper}[2]{}
\newcommand*{\LRCornerWallPaper}[2]{}
\newcommand*{\ThisLRCornerWallPaper}[2]{}
\newcommand*{\ClearWallPaper}{}
\newlength{\wpXoffset}
\newlength{\wpYoffset}
```

File 189 **lwarp-watermark.sty**

## § 270 Package **watermark**

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg watermark

watermark is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{watermark}
```

```
\newcommand{\watermark}[1]{}
\newcommand{\leftwatermark}[1]{}
\newcommand{\rightwatermark}[1]{}
\newcommand{\thiswatermark}[1]{}
\newcommand{\thispageheading}[1]{}
```

## File 190 lwarp-wrapfig.sty

# § 271 Package **wrapfig**

(*Emulates or patches code by* DONALD ARSENEAU.)

Pkg wrapfig

wrapfig is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{wrapfig}
```

```
\newcommand*{\LWR@wrapposition}{}

\newcommand*{\LWR@subwrapfigure}[2]{%
\renewcommand*{\LWR@wrapposition}{}%
\ifthenelse{%
    \equal{#1}{r}\OR\equal{#1}{R}\OR%
    \equal{#1}{o}\OR\equal{#1}{O}%
}%
{\renewcommand*{\LWR@wrapposition}{float:right}}%
{\renewcommand*{\LWR@wrapposition}{float:left}}%
\setlength{\LWR@templengthone}{#2}%
\uselengthunit{PT}%
\LWR@BlockClassWP{%
    width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition; %
    margin:10pt%
}%
{%
    width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition%
}%
{marginblock}%
}


\NewDocumentEnvironment{wrapfigure}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
\captionsetup{type=figure}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wraptable}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
\captionsetup{type=table}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wrapfloat}{m o m o m}
{%
\LWR@subwrapfigure{#3}{#5}%
\captionsetup{type=#1}%
}
{%
\endLWR@BlockClassWP%
}

\newlength{\wrapoverhang}
```

File 191 `lwarp-xcolor.sty`

# § 272 Package **xcolor**

(*Emulates or patches code by* DR. UWE KERN.)

Pkg xcolor

xcolor is supported by lwarp.

## § 272.1 Limitations

\colorboxBlock and \fcolorboxBlock

`\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by lwarp's core if xcolor is loaded.

background: none

`\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

color support

Color definitions, models, and mixing are fully supported without any changes required.

tables

Colored tables are ignored so far. Use CSS to style tables.

colored text and boxes

`\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

\color and \pagecolor \color and \pagecolor are ignored. Use CSS or \textcolor where possible.

## § 272.2 Xcolor definitions: location and timing

The lwarp core and its lwarp-xcolor package are tightly integrated to allow comparable results for print, HTML and print inside an HTML lateximage. This requires a number of definitions and redefintions depending on whether each of xcolor and lateximage is being used, and whether print or HTML is being generated. Some of these actions are one-time when xcolor is loaded, and others are temporary as lateximage is used.

**When xcolor is loaded in print mode:** No special actions are taken at the time that xcolor is loaded in print mode, but see \AtBeginDocument below.

**When lwarp-xcolor is loaded in HTML mode:** xcolor's original definitions are saved for later restoration. \LWR@restoreorigformatting is appended to restore these definitions for use inside a lateximage. New HTML-mode definitions are created for \textcolor, \pagecolor, \nopagecolor, \colorbox, \colorboxBlock, \fcolorbox, \fcolorboxBlock, and fcolorminipage.

**\AtBeginDocument in print or HTML mode:** See Section 73. If xcolor has been loaded, the print-mode \fcolorbox is modified to accept a background color of none, and additional definitions are created for lwarp's new macros print-mode macros \colorboxBlock, \fcolorboxBlock, and fcolorminipage. The HTML versions of these macros will already have been created by lwarp-xcolor if it has been loaded.

For use inside an HTML lateximage, \LWR@restoreorigformatting is appended to temporarily set these functions to their print-mode versions.

**In a lateximage in HTML mode:** \LWR@restoreorigformatting temporarily restores the print-mode definitions of xcolor's functions. See \LWR@restoreorigformatting on page 365.

\color:

**Print:** Used as-is.

**HTML:** Ignored by pdftotext, and will not appear.

**HTML lateximage:** Colors will appear in a lateximage.

\textcolor:

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 647.

**HTML lateximage:** Remembers and reuses the print version.

\pagecolor:

**Print:** Used as-is.

**HTML:** Ignored.

**HTML `lateximage`:** Colors will be picked up in a `lateximage`.

**`\nopagecolor`:**

**Print:** Used as-is.

**HTML:** Ignored.

**HTML `lateximage`:** Colors will be picked up in a `lateximage`.

**`\colorbox`:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 647.

**HTML `lateximage`:** Remembers and reuses the print version.

**`\colorboxBlock`:**

**Print:** Becomes `\colorbox`.

**HTML:** Newly defined by lwarp-xcolor to use a `<div>`, page 648.

**HTML `lateximage`:** Remembers and reuses the print version `\colorbox`.

**`\fcolorbox`:**

**Print:** Modified to allow a background of `none`.
`\LWRprint@fcolorbox` at section 73

**HTML:** Redefined by lwarp-xcolor, page 648.

**HTML `lateximage`:** Remembers and reuses the print version.

**`\fcolorboxBlock`:**

**Print:** Becomes `\fcolorbox`. Section 73

**HTML:** Newly defined by lwarp-xcolor to use a `<div>`, page 649.

**HTML `lateximage`:** Remembers and reuses the print version `\fcolorbox`.

**`fcolorminipage`:**

**Print:** Newly defined in the lwarp core.
`LWRprint@fcolorminipage` at section 73

**HTML:** Newly defined by lwarp-xcolor, page 649.

**HTML `lateximage`:** Uses the print version.

**`\boxframe`:**

**Print:** Used as-is.

**HTML:** Redefined by lwarp-xcolor, page 650.

**HTML `lateximage`:** Remembers and reuses the print version.

## § 272.3 Package loading

**for HTML output:**

```
\LWR@ProvidesPackagePass{xcolor}
```

```
\begin{warpHTML}
```

## § 272.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
\LetLtxMacro\LWRprint@textcolor\textcolor
\LetLtxMacro\LWRprint@pagecolor\pagecolor
\LetLtxMacro\LWRprint@nopagecolor\nopagecolor
\LetLtxMacro\LWRprint@colorbox\colorbox
\LetLtxMacro\LWRprint@colorboxBlock\colorbox
\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
\LetLtxMacro\LWRorigprint@boxframe\boxframe
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\textcolor\LWRprint@textcolor%
\LetLtxMacro\pagecolor\LWRprint@pagecolor%
\LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
\LetLtxMacro\colorbox\LWRprint@colorbox%
\LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
\LetLtxMacro\boxframe\LWRorigprint@boxframe%
}
```

## § 272.5 HTML color style

`\LWR@tempcolor` The color converted to HTML colorspace.

```
\newcommand*{\LWR@tempcolor}{}
\newcommand*{\LWR@tempcolortwo}{}
```

`\LWR@findcurrenttextcolor` Sets `\LWR@tempcolor` to the current color.

```
\newcommand*{\LWR@findcurrenttextcolor}{%
\protect\colorlet{LWR@current@color}{.}%
\protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
}
```

`\LWR@currenttextcolorstyle` Prints a color style for the current color.

```
\newcommand*{\LWR@currenttextcolorstyle}{%
\LWR@findcurrenttextcolor%
\ifdefstring{\LWR@tempcolor}{000000}%
{}%
{color: \#\LWR@tempcolor ; }%
}
```

`\LWR@textcurrentcolor` {⟨*text*⟩} Like `\textcolor` but uses the current `\color` instead.

```
\newcommand*{\LWR@textcurrentcolor}[1]{%
\begingroup%
\LWR@FBcancel%
\LWR@findcurrenttextcolor%
\InlineClass[color:\#\LWR@tempcolor]{textcolor}{%
    \renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
    #1%
}%
\endgroup%
}
```

`\LWR@colorstyle` {⟨*1: styletext*⟩} {⟨*2: model*⟩} {⟨*3: color*⟩}

For a color style, prints the color converted to HTML colors.

```
\NewDocumentCommand{\LWR@colorstyle}{m m}{%
\begingroup%
\LWR@FBcancel%
```

Use the xcolor package to convert to an HTML color space:

```
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
\#\LWR@tempcolor%
\endgroup%
}
```

`\LWR@backgroundcolor` [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```
\NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
#3%
}%
\endgroup%
}
```

## § 272.6 HTML border

\LWR@borderpadding {⟨*colorstyle*⟩} {⟨*color*⟩} Prints the HTML attributes for a black border and padding. \LWR@forceminwidth must be used first in order to set the border width.

```
\newcommand*{\LWR@borderpadding}[2]{%
\uselengthunit{PT}%
border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
}
```

## § 272.7 High-level macros

\color \color appears in the LaTeX PDF output, but is ignored by pdftotext and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is \let to the final name once the HTML conversion starts.

\textcolor [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex color span.

```
\RenewDocumentCommand{\textcolor}{O{named} m m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
#3%
}%
\endgroup%
}
```

\pagecolor [⟨*model*⟩] {⟨*color*⟩}

Ignored. Use \CSSFilename instead.

```
\renewcommand*{\pagecolor}[2][named]{}
```

\nopagecolor Ignored.

```
\renewcommand*{\nopagecolor}{}
```

\colorbox [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex background color <span>.

```
\RenewDocumentCommand{\colorbox}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\InlineClass[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
]{colorbox}{#3}%
\endgroup%
}
```

\colorboxBlock [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex background color `<div>`.

```
\NewDocumentCommand{\colorboxBlock}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\begin{BlockClass}[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
]{colorboxBlock}
#3
\end{BlockClass}%
\endgroup%
}
```

\fcolorbox [⟨*framemodel*⟩] {⟨*framecolor*⟩} [⟨*boxmodel*⟩] {⟨*boxcolor*⟩} {⟨*text*⟩}

Converted into a framed HTML hex background color span.

A background color of `none` creates a colored frame without a background color.

```
\RenewDocumentCommand{\fcolorbox}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorbox #2 #4}%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \InlineClass[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorbox}{#5}%
}%
{% yes background color
```

```
    \InlineClass[%
    \LWR@borderpadding{#1}{#2} ; %
    background:\LWR@colorstyle{#3}{#4}%
    ]{fcolorbox}{#5}%
}%
\endgroup%
}
```

\fcolorboxBlock [⟨*framemodel*⟩] {⟨*framecolor*⟩} [⟨*boxmodel*⟩] {⟨*boxcolor*⟩} {⟨*text*⟩}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```
\NewDocumentCommand{\fcolorboxBlock}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \begin{BlockClass}[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
    \end{BlockClass}%
}%
{% yes background color
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \begin{BlockClass}[%
    background:\#\LWR@tempcolortwo\ ; %
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
    \end{BlockClass}%
}%
\endgroup%
\LWR@traceinfo{HTML fcolorboxBlock done}%
}
```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section 73

\LWR@subfcolorminipage {⟨*framemodel*⟩} {⟨*framecolor*⟩} {⟨*background tag*⟩} {⟨*height*⟩}

```
\NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
\begin{BlockClass}[%
#3%
\LWR@borderpadding{#1}{#2} ; %
\IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
width:\rndprintlength{\LWR@tempwidth}%
]{fcolorminipage}%
}
```

Env fcolorminipage [⟨*1:framemodel*⟩] {⟨*2:framecolor*⟩} [⟨*3:boxmodel*⟩] {⟨*4:boxcolor*⟩} [⟨*5:align*⟩] [⟨*6:height*⟩] [⟨*7:inner-align*⟩] {⟨*8:width*⟩}

```
\NewDocumentEnvironment{fcolorminipage}{O{named} m O{named} m O{c} o o m}
{%
\LWR@FBcancel%
\setlength{\LWR@tempwidth}{#8}%
\IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
\ifthenelse{\equal{#4}{none}}%
{\LWR@subfcolorminipage{#1}{#2}{}{#6}}%
{%
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
}%
}
{\end{BlockClass}}
```

\boxframe {⟨*width*⟩} {⟨*height*⟩} {⟨*depth*⟩}

The depth is added to the height, but the box is not decended below by the depth. `\textcolor` is honored.

```
\renewcommand*{\boxframe}[3]{%
{%
\setlength{\LWR@tempwidth}{#1}%
\setlength{\LWR@tempheight}{#2}%
\addtolength{\LWR@tempheight}{#3}%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\InlineClass[%
display:inline-block ; %
border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
width:\rndprintlength{\LWR@tempwidth} ; %
height:\rndprintlength{\LWR@tempheight}%
]{boxframe}{}%
}%
}
```

```
\end{warpHTML}
```

File 192 **`lwarp-xfrac.sty`**

## § 273 Package **xfrac**

(*Emulates or patches code by* THE LATEX3 PROJECT.)

Pkg `xfrac` Supported by adding xfrac instances.

**for HTML output:**

```
\LWR@ProvidesPackagePass{xfrac}
```

⚠ font size In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by `pdftotext` to a plain text file. While the text may be in any font which `pdftotext` can read, the math is directly converted into SVG images using this same user-selected font. `xfrac` below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` `[`⟨*instance*⟩`]` `{`⟨*num*⟩`}` `[`⟨*sep*⟩`]` `{`⟨*denom*⟩`}`

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

**for HTML & PRINT:**

```
\begin{warpall}
```

`\xfracHTMLfontsize` User-redefinable macro which controls the font size of the fraction.

```
\newcommand*{\xfracHTMLfontsize}{.6em}
```

```
\end{warpall}
```

**for HTML output:**

```
\begin{warpHTML}
```

font size A span for a small font, used in the numerator and denominator:

```
\newcommand*{\LWR@htmlsmallfontstart}{%
\LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"{}}%
\LWR@nestspan%
%
}

\newcommand*{\LWR@htmlsmallfontend}{%
\LWR@htmltagc{/span}%
\endLWR@nestspan%
}
```

\scalebox A nullified \scalebox command, to avoid introducing HTML scaling tags:

```
\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.

```
\DeclareInstance{xfrac}{default}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmr}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmss}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmtt}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}
```

```
\end{warpHTML}
```

File 193 **lwarp-xltxtra.sty**

## § 274 Package **xltxtra**

(*Emulates or patches code by* WILL ROBERTSON, JONATHAN KEW.)

Pkg xltxtra xltxtra is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{xltxtra}
```

```
\RequirePackage{realscripts}
\RequirePackage{metalogo}
\newcommand*\TeX@logo@spacing[6]{}

\newcommand*{\vfrac}[2]{%
\textsuperscript{#1}/\textsubscript{#2}%
}

\newcommand\namedglyph[1]{%
  \@tempcnta=\XeTeXglyphindex "#1"\relax
  \ifnum\@tempcnta>0
    \XeTeXglyph\@tempcnta
  \else
    \xxt@namedglyph@fallback{#1}%
  \fi}

  \newcommand\xxt@namedglyph@fallback[1]{[#1]}

\DeclareDocumentCommand{\showhyphens}{m}{}
```

File 194 `lwarp-xmpincl.sty`

## § 275 Package **xmpincl**

(*Emulates or patches code by* Maarten Sneep.)

Pkg `xmpincl` Emulated.

**for HTML output:** Discard all options for lwarp-xmpincl:

```
\LWR@ProvidesPackageDrop{xmpincl}
```

```
\newcommand*{\includexmp}[1]{}
```

File 195 `lwarp-xtab.sty`

## § 276 Package **xtab**

(*Emulates or patches code by* Peter Wilson.)

Pkg `xtab` xtab is emulated.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{xtab}
```

⚠ misplaced alignment alignment tab character & For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.7.

```
\newcommand{\LWRXT@firsthead}{}

```

```
\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRXT@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}

\newcommand{\tablelasthead}[1]{}

\newcommand{\notablelasthead}{}

\newcommand{\tabletail}[1]{}

\newcommand{\LWRXT@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRXT@lasttail{#1}%
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRXT@caption{\caption[#1]{#2}}%
}

\let\topcaption\tablecaption
\let\bottomcaption\tablecaption

```

```
\newcommand*{\LWRXT@caption}{}

\newcommand*{\shrinkheight}[1]{}

\newcommand*{\xentrystretch}[1]{}

\NewDocumentEnvironment{xtabular}{s o m}
{
\LWR@traceinfo{xtabular}
\table
\LWRXT@caption
\begin{tabular}{#3}
\TabularMacro\ifdefvoid{\LWRXT@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
}
{%
\ifdefvoid{\LWRXT@lasttail}%
{}%
{%
\TabularMacro\ResumeTabular%
\LWRXT@lasttail%
}%
\end{tabular}
\endtable
\LWR@traceinfo{xtabular done}
}

\NewDocumentEnvironment{mpxtabular}{s o m}
{\minipage{\linewidth}\xtabular{#3}}
{\endxtabular\endminipage}
```

File 196 **lwarp-zwpagelayout.sty**

# § 277 Package **zwpagelayout**

(*Emulates or patches code by* Zdeněk Wagner.)

Pkg zwpagelayout zwpagelayout is ignored.

**for HTML output:**

```
\LWR@ProvidesPackageDrop{zwpagelayout}
```

```
\def\noBboxes{}
\@onlypreamble\noBboxes

\expandafter\ifx\csname definecolor\endcsname\relax \else
  \definecolor{cmykblack}{cmyk}{0,0,0,1}
  \definecolor{grblack}{gray}{0}
%   \ifzwpl@redefineblack
%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
%   \fi
  \definecolor{cmykred}{cmyk}{0,1,1,0}
  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
  \definecolor{cmykblue}{cmyk}{1,1,0,0}
  \definecolor{rgbred}{rgb}{1,0,0}
  \definecolor{rgbgreen}{rgb}{0,1,0}
  \definecolor{rgbblue}{rgb}{0,0,1}
%   \ifzwpl@redefinetocmyk
%     \definecolor{red}{cmyk}{0,1,1,0}
%     \definecolor{green}{cmyk}{1,0,1,0}
%     \definecolor{blue}{cmyk}{1,1,0,0}
%   \fi
\fi

\let\OverprintXeTeXExtGState\relax

\DeclareRobustCommand\SetOverprint{\ignorespaces}
\DeclareRobustCommand\SetKnockout{\ignorespaces}
\DeclareRobustCommand\textoverprint[1]{{\SetOverprint#1}}
\DeclareRobustCommand\textknockout[1]{{\SetKnockout#1}}
```

```

\def\SetPDFminorversion#1{}
\@onlypreamble\SetPDFminorversion

\newcommand*\Vcorr{}

\DeclareRobustCommand\vb[1][]{}
\NewDocumentCommand{\NewOddPage}{* o}{}
\NewDocumentCommand{\NewEvenPage}{* o}{}
\def\SetOddPageMessage#{\gdef\ZW@oddwarning}
\def\SetEvenPageMessage#{\gdef\Z@@evenwarning}
\def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning

\def\clap#1{#1}

\def\CropFlap{2in}
\def\CropSpine{1in}
\def\CropXSpine{1in}
\def\CropXtrim{.25in}
\def\CropYtrim{.25in}
\def\UserWidth{5in}
\def\UserLeftMargin{1in}
\def\UserRightMargin{1in}
\def\UserTopMargin{1in}
\def\UserBotMargin{1in}
\def\thePageNumber{\#\,\arabic{page}}
\ifXeTeX
\def\ifcaseZWdriver{\ifcase2}
\else
\def\ifcaseZWdriver{\ifcase1}
\fi
\DeclareRobustCommand\ZWifdriver[2]{}
```

# Change History and Index

## § 277 Change History

glossaries: Prevent error with `\glo@name` not defined. . . . . . 365

`lwarpmk`: Fix: `io.lines()` changed to `file:lines()` due to `luatex` changes. . . . . . . . . . . 199

v0.33

`\HTMLAuthor`: Fix: Provides empty default author if none given. . . 239

`\LWR@loadbefore`: Fix: No `\PackageError` if already loaded. . . . . . . . . . . . . . . . . . 135

`\LWR@parseatcolumn`: Fix: Column alignment with leftmost `@`. . . . 301

`\LWR@tabledatasinglecolumntag`: Fix: Macros in tabular could cause extra data cell. . . . . . . . 312

`\LWR@vspace`: Add: `\vspace` nullified. . . . . . . . . . . . . . . . . 425

`\StartDefiningTabulars`: Add: Avoids error: `misplaced alignment tab character &`. 296

General: 2017/07/10 . . . . . . . . . . . 1

amsmath: Removed `fleqn` option. 144

fancyhdr: Fix: Optional args for `\lhead`, etc. . . . . . . . . . . . . . . 467

Add: Tabular at and bang columns now have their own HTML columns. . . . . . . . . . . . . . . . . 292

cleveref: Fix: Loaded `\AtEndPreamble`. . . . . . . . . . 402

Fix: Incorrectly-inline math environments. . . . . . . . . . . . . 379

New handling of `&` to localize catcode changes. . . . . . . . . . . 292

v0.34

`\@fnsymbol`: Text symbols instead of math. . . . . . . . . . . . . . . . . . . 275

`\InlineClass`: Moved optional argument in front of mandatory. 229

`\LWR@htmldivclass`: Moved optional argument in front of mandatory. . . . . . . . . . . . . . . 227

`\LWR@htmlelementclass`: Moved optional argument in front of mandatory. . . . . . . . . . . . . . . 227

`\LWR@htmlelementclassline`: Moved optional argument in front of mandatory. . . . . . . . . 227

`\LWR@htmlspanclass`: Moved optional argument in front of mandatory. . . . . . . . . . . . . . . 224

`\LWR@nullfonts`: Improved font control. . . . . . . . . . . . . . . . . . 417

`\LWR@restoreorigformatting`: booktabs: Works inside `lateximage`. . . . . . . . . . . . . . 365

Improved font control. . . . . . . . 365

`\LWR@subhtmlelementclass`: Moved optional argument in front of mandatory. . . . . . . . . 226

`\LWR@tabledatacolumntag`: booktabs: Works inside `lateximage`. . . . . . . . . . . . . . 331

`\fboxBlock`: Added. . . . . . . . . . . . 412

`\makebox`: Fix: Handles paren arg. . 410

General: 2017/08/08 . . . . . . . . . . . 1

babel-french: Adds fixed-width HTML spaces to punctuation. . 222

balance: Added. . . . . . . . . . . . . 446

booktabs: Works inside `lateximage`. . . . . . . . . . . 334, 449

boxedminipage2e: Added. . . . . . 449

boxedminipage: Prevented. . . . . 449

crop: Added. . . . . . . . . . . . . . . 453

enumerate: Added. . . . . . . . . . . 459

enumitem: Added, no longer required. . . . . . . . . . . . . . . . . 459

everyshi: Added. . . . . . . . . . . . . 462

fancybox: Added. . . . . . . . . . . . 464

fancyvrb: Added, no longer required. . . . . . . . . . . . . . . . . 469

figcaps: Added. . . . . . . . . . . . . . 474

filecontents: Required. Patched for morewrites. . . . . . . . . . . . . . . 143

floatpag: Added. . . . . . . . . . . . . 479

flushend: Added. . . . . . . . . . . . 485

fullpage: Added. . . . . . . . . . . . . 492

hyperxmp: Added. . . . . . . . . . . 513

idxlayout: Added. . . . . . . . . . . . 513

marginfit: Added. . . . . . . . . . . . 527

mdframed: Improved `mdtheorem` patch. . . . . . . . . . . . . . . . . . . 535

moreverb: Added. . . . . . . . . . . . 539

paralist: Added. . . . . . . . . . . . . 561

pdflscape: Added. . . . . . . . . . . . 562

pdfsync: Added. . . . . . . . . . . . . 562

prelim2e: Added. . . . . . . . . . . . 563

rotfloat: Added. . . . . . . . . . . . . 569

Added vertical rules. . . . . . . . . . 544
Fix: < spec. . . . . . . . . . . . . . . . 544
LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french. . 335
Improved rules. . . . . . . . . . . . . 336

v0.42
\@ensuredmath: Improved \ensuremath. . . . . . . . . . . . . 372
\@textsubscript: Added. . . . . . . 419
\@textsuperscript: Added. . . . . . 419
\LWR@HTMLtextstyle: Added. . . . . 415
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment. . . . . . . . . . . . . 318
\LWR@addrulewidth: If FormatWP force explicit border. . . . . . . . 317
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments. . . . . . . . . . . . . 390
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments. . . . . . . . . 390
\LWR@domulticolumn: If FormatWP add cell alignment. . . . . . . . . 323
\LWR@doubledollar: If FormatWP print LaTeX expression. . . . . . 371
Improved \ensuremath. . . . . . . 371
Improved line spacing with mathjax. . . . . . . . . . . . . . . . 371
\LWR@floatbegin: If FormatWP add a text frame. . . . . . . . . . . . . 350
\LWR@floatend: If FormatWP add a text frame. . . . . . . . . . . . . . 351
\LWR@hline: If FormatWP force explicit border. . . . . . . . . . . 334
\LWR@hspace: If FormatWP add \quads. . . . . . . . . . . . . . . . . 424
\LWR@htmlmathlabel: If FormatWP print LaTeX expression. . . . . . 377
\LWR@includegraphicsb: Fix: Filename expansion. . . . . . . . 499
If FormatWP, use explicit size. . . 501
\LWR@remembertag: Fix: Numbering and naming AMS math environments. . . . . . . . . . . . . 389
\LWR@restoreorigformatting: Improved \ensuremath. . . . . . 367
\LWR@rule: If FormatWP add \quads. . . . . . . . . . . . . . . . . 427
\LWR@singledollar: If FormatWP print LaTeX expression. . . . . . 371
\LWR@subaddcmidruletrim: 0pt if no rule given. . . . . . . . . . . . 316
\LWR@tabledatasinglecolumntag: If FormatWP add cell alignment. 313
\LaTeX: If FormatWP use explicit style. . . . . . . . . . . . . . . . . . 429
\TeX: If FormatWP use explicit style. 429
\listoffigures: Added boolean WPMarkLOFT. . . . . . . . . . . . . 358
\listoftables: Added boolean WPMarkLOFT. . . . . . . . . . . . . 358
\marginpar: If FormatWP emulate a wrapfig. . . . . . . . . . . . . . . . 244
\marginparBlock: If FormatWP emulate a wrapfig. . . . . . . . . . 245
\tableofcontents: Added boolean WPMarkTOC. . . . . . . . . . . . . . 358
\underline: If FormatWP, use explicit styles for \underline, etc. . . . . . . . . . . . . . . . . . . . 420
General: 2017/10/30 . . . . . . . . . . . 1
\textbf and related: If FormatWP, use explicit styles for \textsc, etc. . . . . . . . . . . . . . . . . . . . 415
algorithmicx: If FormatWP add \quads. . . . . . . . . . . . . . . . . 439
epigraph: If FormatWP add HTML styles. . . . . . . . . . . . . . . . . . 460
fancybox: If FormatWP add HTML styles. . . . . . . . . . . . . . . . . . 464
floatflt: Added width. . . . . . . . . 478
includegraphics: Fix: Class key. . . 499
keyfloat: If FormatWP add explicit HTML style. . . . . . . . . . . . . 516
moreverb: Simplified formatting of listings. . . . . . . . . . . . . . . . . 539
morewrites: Added. . . . . . . . . . . 541
multirow: If FormatWP add cell alignment. . . . . . . . . . . . . . 544
overpic: Added. . . . . . . . . . . . . 560
realscripts: Fix for subscripts in a lateximage. . . . . . . . . . . . . 566
sidenotes: If FormatWP add explicit HTML style. . . . . . . . . 583
siunitx: Improved \ensuremath. . 584
soul: If FormatWP, add explicit styles. . . . . . . . . . . . . . . . . . 585

# § 277 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.