Sectioning commands in FoilTEX and conversion to HTML format: FoilHTML Package *†

Boris Veytsman‡

1998/02/17

Abstract

The standard FoilTEX system is based on a visual rather than logical formatting. Thus automatic conversion of FoilTEX documents to HTML format is difficult. In particular, the well-known LATEX2HTML converter gives unsatisfactory results when used on FoilTEX documents.

This package provides integration between FoilTEX and LATEX2HTML. It adds sectioning commands and elements of logical formatting to FoilTEX and provides support for FoilTEX commands in LATEX2HTML. It also supports automatic creation of LATEX2HTML images with proper font sizes and shapes when converting FoilTEX documents.

Contents

*This file has version number v1.2, last revised 1998/02/17.

‡E-mail: boris@plmsc.psu.edu, Home page: http://www.plmsc.psu.edu/~boris

Part I
User Interface

1 Introduction

Many people like to put the contents of their talks and lectures on the World Wide Web using tools like Nikos Drakos' LaTeX2HTML [1]. Unfortunately, when one tries to convert the documents prepared in FoilTeX [2], several problems emerge:

1. FoilTeX does not have proper sectioning commands. Instead it has command like `\foilhead` and `\rotatefoilhead`, that are not recognized by LaTeX2HTML.

2. FoilTeX has commands like `\MyLogo` and `\Restriction` that have no correspondence in the HTML format.

3. The equations and pictures processed by LaTeX2HTML are too large and (for pictures) use Sans Serif fonts. This produces ugly `gif` inserts in HTML files.

This package solves these problems by two methods:

1. It provides a Perl script to teach LaTeX2HTML the basic constructions of FoilTeX and put the correct headers in the LaTeX2HTML `images.tex` file.

2. It provides a LaTeX package that implements sectioning commands in FoilTeX.

2 Installation

Since FoilHTML is an *add-on* to FoilTeX and LaTeX2HTML, you need both these programs installed and working prior to installation of FoilHTML. This package is intended for LaTeX 2$_\varepsilon$ *only*; I don't have enough time to port and test it under LaTeX2.09. It is tested under FoilTeX2.1 and LaTeX2HTML96–LaTeX2HTML98.

The FoilHTML distribution consists of the following files:

- `readme.v12`—Read Me file
- `foilhtml.dtx`—the fully documented code

- `foilhtml.ins`—the LaTeX program extracting the code from the `.dtx` file.

To install the package:

1. Run LaTeX on `foilhtml.ins`. This will create the following files:
   - `foils.perl`—Perl script for LaTeX2HTML98
   - `foilhtml.sty` and `foilhtml.cfg`—LaTeX style file and the configuration file for user-customization,
   - `foilhtml.drv`—documentation driver.
   - `foilhtml-96.perl` and `foils-97.perl`—obsolete versions of Perl scripts required *only* if you have an obsolete installation of LaTeX2HTML.
2. Install the script `foils.perl` in the directory where your LaTeX2HTML scripts reside (usually it is `styles` subdirectory of the main LaTeX2HTML directory, e. g. `/usr/local/lib/latex2html/styles`).
3. Tune the configuration file `foilhtml.cfg` according to your taste and put it together with the file `foilhtml.sty` to the directory read by LaTeX[1].
4. Run LaTeX on `foilhtml.drv` to produce the documentation[2].
5. (Optional) to produce the documentation *and* the code run LaTeX through the file `foilhtml.dtx`

2.1 Obsolete versions

If you have older versions of LaTeX2HTML, your best options is to upgrade. However, if you cannot do this by some reason, you can use the previous versions of the perl scripts. Just rename the files `foilhtml-96.perl` *or* `foils-97.perl` to `foilhtml.perl` or `foils.perl` correspondingly and install them in the `styles` subdirectory.

3 Basic Usage

The usage of FoilHTML could be very simple. Just add to the preamble of your document `\usepackage{foilhtml}`, for example

```
\documentclass[20pt,dvips]{foils}
\usepackage{foilhtml,html}
```

and your FoilTeX documents will be correctly parsed by LaTeX2HTML[3]. The FoilTeX-specific commands `\foilhead` and `\rotatefoilhead` will be translated by

`\foilhead`
`\rotatefoilhead`

[1] If you use `kpathsea` (e. g. if you have teTeX distribution), you will need to run `texhash` to update your search database

[2] The documentation driver `foilhtml.drv` reads the contents of the file `foilhtml.dtx`, so do not delete the latter until you produced the documentation.

[3] Actually, if you use LaTeX2HTML97-1 or later, you don't need even to load the package `foilhtml`: unlike the old versions of LaTeX2HTML, this one parses both `\usepackage` and `\documentclass` commands. However, you will want to load this package if you need the new FoilTeX commands described below.

LaTeX2HTML as starred `\subsection*` commands (this behavior could be changed in your `.latex2html-init` file—see Section 4).

However, this is not the full story. In addition, you will be able to use "normal" sectioning commands like `\section` and `\chapter` and even `\tableofcontents`. By default, commands `\part` and `\chapter` produce new foils containing only titles of the corresponding, and the command `\section` starts a new foil. The starred forms of these commands produce unnumbered sections that are not included in the table of contents.

In the "vanilla" FoilTeX the commands `\foilhead` and `rotatefoilhead` have both logical (they are close to sectioning commands) and visual meanings. If you want to convert your documents to HTML, you need to adhere to logical formatting. Therefore, their usage is discouraged. Most important, you should *not* use the commands `\foilhead{}` and `\rotatefoilhead{}` without arguments to produce blank pages: they will be treated by LaTeX2HTML as `\subsection*{}` commands with blank subsection name. Instead, use `\clearpage` and the commands `\portraitfoils` and `\landscapefoils`, described below.

While the sectioning commands take care of the logical formatting, it is impossible to get rid of visual formatting of foils altogether. FoilHTML package provides some new commands for visual formatting.

`\portraitfoils`
`\landscapefoils`
`\portraitfoils*`
`\landscapefoils*`

The commands `\portraitfoils` and `\landscapefoils` start a new foil with the selected orientation. Their starred forms do not eject the page, and the selected orientation begins only from the next sectioning command at the level of `section` or greater. Note that the explicit commands `\foilhead` and `\rotatefoilhead` might change the orientation of the produced pages. However, the default orientation will be restored by the next sectioning command on the level of `section` or greater.

4 Customization

FoilHTML should work "right out of the box". However, there some ways to customize its behavior for picky users.

4.1 Configuring the LaTeX2HTML behavior

There are three Perl variables that control translation that could be set in your `.latex2html-init` file

`$FOILHEADLEVEL`

The value of `$FOILHEADLEVEL` describes the level of sectioning for explicit `foilhead` and `rotatefoilhead` commands. By default it is 4 (subsection). Other values are 1 (part), 2 (chapter), etc. up to 7 (subparagraph).

`$FOILCLASS`
`$FOILOPTIONS`

The header of the `images.tex` is

```
\documentclass[$FOILOPTIONS]{$FOILCLASS}
```

The defaults are `10pt` and `article`. The variables can be set to any valid LaTeX values.

For example, the following lines in your `.latex2html-init` file:

```
$FOILHEADLEVEL = 2;
$FOILOPTIONS = '12pt';
$FOILCLASS = 'book';
```

produce HTML file with `foilhead` commands translated as Chapters, and the images created as for a 12pt book.

4.2 LaTeX variables and options

There are several ways to change the look of the LaTeX document.

`\contentsname` By default, the Table of Contents is titled "Outline" in the printed document. This could be changed by changing the macro `\contentsname` like this:

```
\renewcommand{\contentsname}{The Outline of Today's Talk}
```

Note that the in the HTML version Table of Contents is still called "Contents". This could be changed separately in your `.latex2html-init` file.

`[pagesintoc]` `[nopagesintoc]` By default, the Table of Contents does not include the page numbers of the corresponding sections (they have little sense for a lecture). However, if you wish to include them in the Table of Contents, use package option `pagesintoc`. The package option `nopagesintoc` (default) suppresses the page numbers.

By default, `foilhtml` does *not* number even the non-starred sections, subsections, etc. This behavior is determined by the variable `secnumdepth` (by default 0). This variable has the same meaning here, as in the standard LaTeX classes. All sectioning units of the level larger than `secnumdepth` are numbered, and the others are not.

4.3 LaTeX configuration file

If the file `foilhtml.cfg` exists, it is loaded by the style `foilhtml.sty`. If you use some customization code often, you can put it there. A sample configuration file is provided in the distribution; feel free to hack it.

5 New versions and bug reports

Feel free to send bug reports to `boris@plmsc.psu.edu`. You can find new versions of the package on my home page (`http://www.plmsc.psu.edu/~boris` or on CTAN.

6 Legalese

The package FoilHTML is provided "as is" and comes with absolutely no warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law will the author of the program be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the program or out of inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or by third parties as a result of a failure

of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

FoilHTML package is covered by the same license as the current LaTeX 2$_\varepsilon$ package (see the files `legal.txt` and `modguide.tex` in the LaTeX 2$_\varepsilon$ distribution).

The program FoilTeX is copyrighted by IBM Corporation and covered by a separate license.

The program LaTeX2`HTML` is copyrighted by Nikos Drakos and covered by a separate license.

Part II
Implementation

7 Perl code

⟨∗foilhtml | foils − 97 | foils⟩

First, the header:

```
##########################################################
#
⟨+foilhtml⟩# This is file `foilhtml.perl',
⟨+foils⟩# This is file `foils.perl',
# generated with the docstrip utility.
#
# Copyright Boris Veytsman 1998
#
#  You are *not* allowed to modify this file.
#
#  You are *not* allowed to distribute this file.
# For distribution of the original source see
# the terms for copying and modification in the file
# foilhtml.dtx
#
```

$FOILHEADLEVEL

Now we define `\foilhead` and `\rotatefoihead` as sectioning commands. By default their level is "section", but this can be used by setting `$FOILHEADLEVEL` to 1 (part), 2 (chapter), 3 (section), etc. up to 7 (subparagraph). Note that this will affect only LaTeX2HTML sectioning commands. The typesetting will the same: `<H1>`-level headings. Note, that article class does not have chapters, so if you want to make `$FOILHEADLEVEL` to be 2, you need to redefine `$FOILCLASS` below. Otherwise TeXing `images.tex` will give an error.

```
$FOILHEADLEVEL = 4 unless defined($FOILHEADLEVEL);
%section_commands=('foilhead',$FOILHEADLEVEL,
  'rotatefoilhead',$FOILHEADLEVEL,%section_commands);

%new_section_headings = ('foilhead', 'H1','rotatefoilhead','H1');
&generate_new_sectioning_subs;

sub generate_new_sectioning_subs{
   local($key, $val);
   while ( ($key, $val) = each %new_section_headings) {
       eval "sub do_cmd_$key {"
           . 'local($after) = @_;'
               . '&do_cmd_section_helper(' . $val . ');}';
       %section_commands=($key,$val,%section_commands);
   }
};
```

Now the redefinition of the `images.tex` headers. We just lift the definition from the original LaTeX2HTML and make some modifications.

$FOILCLASS
$FOILOPTIONS

We substitute the header `\documentclass[`⟨*options*⟩`]{foils}` by the header `\documentclass[$FOILOPTIONS]{$FOILCLASS}` (note that FoilTeX options have

no meaning in the HTML world, so we can safely drop them). The defaults are `10pt` and `article`, but this could be easily changed in the init file.

```
$FOILCLASS = 'article' unless defined $FOILCLASS;
$FOILOPTIONS = '10pt' unless defined $FOILOPTIONS;
```

The processing is different for different versions of LaTeX2HTML. First, the 96 version:

⟨/foilhtml | foils − 97 | foils⟩
⟨∗foilhtml⟩

```
sub make_latex{
# This is the environment in which to process constructs that cannot be
# translated to HTML.
# The environment tex2html_wrap will be wrapped around any shorthand
# environments (e.g. $, \(, \[).
# The tex2html_wrap environment will be treated as an unrecognised
# evironment by the translator and its contents (i.e. the 'shorthand'
# environment) will be passed to latex for processing as usual.
   local($contents) = @_;
   local($preamble) = $preamble;
   # Make the @ character a normal letter ...
   $preamble =~
     s/\\documentclass.*/
     \\documentclass\[$FOILOPTIONS\]{$FOILCLASS}\\makeatletter/;
   # ... and make it special again after the preamble
   ($DEBUG ? "\\nonstopmode" : "\\batchmode") .
   "\n$preamble\n\\makeatother\n" .
   "\\newenvironment{tex2html_wrap}{}{}\n" .
   "\\newwrite\\lthtmlwrite\n" .
   "\\def\\lthtmltypeout#1%\n" .
   "{{\\let\\protect\\string\\immediate\\write\\lthtmlwrite{#1}}}%\n" .
   "\\newbox\\sizebox\n" .
   # "\\textheight 250cm\n" .
   "\\begin{document}\n" .
   "\\pagestyle{empty}\n" .
   "$contents\n".
   "\\end{document}";
}
```

The 97 version is much longer:

⟨/foilhtml⟩
⟨∗foils − 97⟩

```
sub make_latex{
# This is the environment in which to process constructs that cannot be
# translated to HTML.
# The environment tex2html_wrap will be wrapped around any shorthand
# environments (e.g. $, \(, \[).
# The tex2html_wrap environment will be treated as an unrecognised
# evironment by the translator and its contents (i.e. the 'shorthand'
# environment) will be passed to latex for processing as usual.
    local($contents) = @_;
    local($preamble) = $preamble;
    local($preamble_aux) = $preamble_aux;
    # Make the @ character a normal letter ...
    $preamble =~
      s/\\documentclass.*/
```

```
81        \\documentclass\[$FOILOPTIONS\]{$FOILCLASS}\\makeatletter/;
82      # ... and make it special again after the preamble
83      # remove the  \begin/\end  for  tex2html_nowrap and tex2html_deferred environments
84      $preamble =~s/\\(begin|end)\s*\{(tex2html_(nowrap|deferred|nomath)[_a-z]*|imagesonly)\}//g;
85      $preamble = "\\documentclass\{article\}%\\usepackage{html}%\n\\makeatletter"
86          unless ($preamble);
87      $preamble_aux = '' unless (($preamble_aux)&&($contents =~ /\\(hyper)?(ref|cite)/));
88
89      local($paperwidth) = '';
90      if ($PAPERSIZE) { $paperwidth = &adjust_textwidth($PAPERSIZE); }
91      else { $paperwidth = &adjust_textwidth("a5"); }
92      local($kern) = ($EXTRA_IMAGE_SCALE ? $EXTRA_IMAGE_SCALE/2 : ".5" );
93      $kern = $kern * $MATH_SCALE_FACTOR;
94      ($DEBUG ? "\\nonstopmode" : "\\batchmode") .
95      "\n$preamble\n$preamble_aux\\makeatother\n" .
96      "\\ifx\\AtBeginDocument\\undefined \\newcommand{\\AtBeginDocument}[1]{}\\fi\n" .
97      "\\newenvironment{tex2html_wrap}{}{}\n" .
98      "\\newbox\\sizebox\n" . "$paperwidth" .
99      "\\newwrite\\lthtmlwrite\n" . "\\makeatletter\n" .
100     "\\let\\realnormalsize=\\normalsize\n\\topskip=0pt\n\\def\\preveqno{}" .
101     "\\let\\real\@float=\\\@float \\let\\realend\@float=\\end\@float\n" .
102     "\\def\\\@float{\\let\\\@savefreelist\\\@freelist\\real\@float}\n" .
103 #    "\\def\\\@float{\\\@dbflt}\n" .
104     "\\def\\end\@float{\\realend\@float\\global\\let\\\@freelist\\\@savefreelist}\n" .
105     "\\let\\real\@dbflt=\\\@dbflt \\let\\end\@dblfloat=\\end\@float\n" .
106     "\\let\\\@largefloatcheck=\\relax\n" .
107     "\\def\\\@dbflt{\\let\\\@savefreelist\\\@freelist\\real\@dbflt}\n" .
108     "\\def\\adjustnormalsize{\\def\\normalsize{\\mathsurround=0pt \\realnormalsize" .
109     "\\parindent=0pt\\abovedisplayskip=0pt\\belowdisplayskip=0pt}\\normalsize}\n" .
110     "\\def\\lthtmltypeout#1{{\\let\\protect\\string\\immediate\\write\\lthtmlwrite{#1}}}%\n" .
111     "\\newcommand\\lthtmlhboxmathA{\\adjustnormalsize\\setbox\\sizebox=\\hbox\\bgroup}%\n" .
112     "\\newcommand\\lthtmlvboxmathA{\\adjustnormalsize\\setbox\\sizebox=\\vbox\\bgroup%\n".
113     " \\let\\ifinner=\\iffalse }%\n" .
114     "\\newcommand\\lthtmlboxmathZ{\\\@next\\next\\\@currlist{}{\\def\\next{\\voidb\@x}}%\n" .
115 #     " \\expandafter\\box\\next\\edef\\next{\\egroup\\def\\noexpand\\thiseqn{\\theequation}}\\nex
116     " \\expandafter\\box\\next\\egroup}%\n" .
117     "\\newcommand\\lthtmlmathtype[1]{\\def\\lthtmlmathenv{#1}}%\n" .
118 #     "\\newcommand\\lthtmllogmath{\\lthtmltypeout{latex2htmlSize%\n" .
119     "\\newcommand\\lthtmllogmath{\\lthtmltypeout{l2hSize %\n" .
120     ":\\lthtmlmathenv:\\the\\ht\\sizebox::\\the\\dp\\sizebox::\\the\\wd\\sizebox.\\preveqno}}%\n"
121     "\\newcommand\\lthtmlfigureA[1]{\\let\\\@savefreelist\\\@freelist
122       \\lthtmlmathtype{#1}\\lthtmlvboxmathA}%\n" .
123     "\\newcommand\\lthtmlfigureZ{\\lthtmlboxmathZ\\lthtmllogmath\\copy\\sizebox
124       \\global\\let\\\@freelist\\\@savefreelist}%\n" .
125     "\\newcommand\\lthtmldisplayA[1]{\\lthtmlmathtype{#1}\\lthtmlvboxmathA}%\n" .
126     "\\newcommand\\lthtmldisplayB[1]{\\edef\\preveqno{(\\theequation)}%\n" .
127     "  \\lthtmldisplayA{#1}\\let\\\@eqnnum\\relax}%\n" .
128     "\\newcommand\\lthtmldisplayZ{\\lthtmlboxmathZ\\lthtmllogmath\\lthtmlsetmath}%\n" .
129     "\\newcommand\\lthtmlinlinemathA[1]{\\lthtmlmathtype{#1}\\lthtmlhboxmathA" .
130     "  \\vrule height1.5ex width0pt }%\n" .
131     "\\newcommand\\lthtmlinlinemathZ{\\egroup\\expandafter\\ifdim\\dp\\sizebox>0pt %\n" .
132     "  \\expandafter\\centerinlinemath\\fi\\lthtmllogmath\\lthtmlsetmath}\n" .
133     "\\def\\lthtmlsetmath{\\hbox{\\vrule width.5pt\\vtop{\\vbox{%\n" .
134     "  \\kern.5pt\\kern$kern pt\\hbox{\\hglue.5pt\\copy\\sizebox\\hglue$kern pt}\\kern.5pt%\n" .
```

```
    "  \\ifdim\\dp\\sizebox>0pt\\kern$kern pt\\fi}%\n" .
    "  \\ifdim\\hsize>\\wd\\sizebox \\hrule depth1pt\\fi}}}\n" .
    "\\def\\centerinlinemath{\\dimen1=\\ht\\sizebox\n" .
    "  \\ifdim\\dimen1<\\dp\\sizebox \\ht\\sizebox=\\dp\\sizebox\n" .
    "  \\else \\dp\\sizebox=\\ht\\sizebox \\fi}\n\n" .
    "\\def\\lthtmlcheckvsize{\\ifdim\\ht\\sizebox<\\vsize\\expandafter\\vfill\n" .
    "  \\else\\expandafter\\vss\\fi}%\n" .
#    "\\def\\\@enddocumenthook{\\ifnum\\count0>1 \\ifvoid\\\@cclv\\penalty-\\\@MM\\fi\\fi}\n" .
    "\\makeatletter\n" .
    $LaTeXmacros . "\n" .  # macros defined in extension files
#    "\\usepackage{lthimages}\n" .
    "\n\\begin{document}\n" .
    "\\pagestyle{empty}\\thispagestyle{empty}%\n" .
    "\\lthtmltypeout{latex2htmlLength hsize=\\the\\hsize}%\n" .
    "\\lthtmltypeout{latex2htmlLength vsize=\\the\\vsize}%\n" .
    "\\lthtmltypeout{latex2htmlLength hoffset=\\the\\hoffset}%\n" .
    "\\lthtmltypeout{latex2htmlLength voffset=\\the\\voffset}%\n" .
    "\\lthtmltypeout{latex2htmlLength topmargin=\\the\\topmargin}%\n" .
    "\\lthtmltypeout{latex2htmlLength topskip=\\the\\topskip}%\n" .
    "\\lthtmltypeout{latex2htmlLength headheight=\\the\\headheight}%\n" .
    "\\lthtmltypeout{latex2htmlLength headsep=\\the\\headsep}%\n" .
    "\\lthtmltypeout{latex2htmlLength parskip=\\the\\parskip}%\n" .
    "\\lthtmltypeout{latex2htmlLength oddsidemargin=\\the\\oddsidemargin}%\n" .
    "\\makeatletter\n" .
    "\\if\@twoside\\lthtmltypeout{latex2htmlLength evensidemargin=\\the\\evensidemargin}%\n" .
    "\\else\\lthtmltypeout{latex2htmlLength evensidemargin=\\the\\oddsidemargin}\\fi%\n" .
    "\\makeatother\n" .
    "$contents\n".
#    "\\clearpage\n" .
    "\\end{document}";
}
```
⟨/foils − 97⟩

And 98 version is even longer...

⟨∗foils⟩
```
sub make_latex{
# This is the environment in which to process constructs that cannot be
# translated to HTML.
# The environment tex2html_wrap will be wrapped around any shorthand
# environments (e.g. $, \(, \[).
# The tex2html_wrap environment will be treated as an unrecognised
# evironment by the translator and its contents (i.e. the 'shorthand'
# environment) will be passed to latex for processing as usual.
   local($contents) = @_;
   local($preamble) = $preamble;
   local($aux_preamble) = $aux_preamble;
   while ($preamble =~ s/^(\@.*\n)/$prelatex .= $1;''/e) {}
   print "\nPRE-LATEX: $prelatex" if (($prelatex)&&($VERBOSITY > 1));

   # Make the @ character a normal letter ...
   $preamble =~ s/\\par([^A-Za-z])/\n$1/g;
    $preamble =~
      s/\\documentclass.*/
      \\documentclass\[$FOILOPTIONS\]{$FOILCLASS}\\makeatletter/;
   # ... and make it special again after the preamble
```

```
    # remove the  \begin/\end  for  tex2html_nowrap and tex2html_deferred environments
    $preamble =~s/\\(begin|end)\s*\{(tex2html_(nowrap|deferred|nomath|preform)[_a-z]*|imagesonly)\
    $preamble = "\\documentclass\{article\}%\n\\usepackage{html,color}%\n\\makeatletter"
unless ($preamble);
    if (($LATEX_DUMP)&&(!($preamble =~ /\\usepackage\{ldump\}/))) {
$*=1;
$preamble =~ s/(\\document(class|style)[^\n]*\n)/$1\\usepackage\{ldump\}\n/;
$*=0;
    }
    $LOAD_LATEX_COLOR = "\\usepackage[dvips]{color}" unless $LOAD_LATEX_COLOR;
    $LATEX_COLOR = "\\pagecolor[gray]{.85}" unless $LATEX_COLOR;
    if ($preamble =~ /(^|\s*[^%])\s*\\documentstyle/) {
# \usepackage is invalid in LaTeX 2.09 and LaTeX-2e compatibility mode
$LATEX_COLOR = ''; $LOAD_LATEX_COLOR = '';
    }

    $preamble .= $LOAD_LATEX_COLOR."\n" unless ($preamble =~ /[,\{]color[,\}]/);
    $preamble .= $LATEX_COLOR."\n" unless ($preamble =~ /\\pagecolor/);

    $aux_preamble = '' unless (($aux_preamble)&&($contents =~ /\\(hyper)?(ref|cite)/));

    local($paperwidth) = '';
    if ($PAPERSIZE) { $paperwidth = &adjust_textwidth($PAPERSIZE); }
    else { $paperwidth = &adjust_textwidth("a5"); }
    local($kern) = ($EXTRA_IMAGE_SCALE ? $EXTRA_IMAGE_SCALE/2 : ".5" );
    $kern = $kern * $MATH_SCALE_FACTOR;
    $prelatex . ($DEBUG ? "\\nonstopmode" : "\\batchmode") .
    "\n$preamble\n$aux_preamble\n" .
    "\\count\@=\\the\\catcode`\\_ \\catcode`\\_=8 \n" .
    "\\newenvironment{tex2html_wrap}{}{} \\catcode`\\_=\\count\@\n" .
    "\\makeatother\n" .
    "\\ifx\\AtBeginDocument\\undefined \\newcommand{\\AtBeginDocument}[1]{}\\fi\n" .
    "\\newbox\\sizebox\n" . "$paperwidth" .
    "\\newwrite\\lthtmlwrite\n" . "\\makeatletter\n" .
    "\\let\\realnormalsize=\\normalsize\n\\global\\topskip=2sp\n\\def\\preveqno{}" .
    "\\let\\real\@float=\\\@float \\let\\realend\@float=\\end\@float\n" .
    "\\def\\\@float{\\let\\\@savefreelist\\\@freelist\\real\@float}\n" .
#    "\\def\\\@float{\\\@dbflt}\n" .
    "\\def\\end\@float{\\realend\@float\\global\\let\\\@freelist\\\@savefreelist}\n" .
    "\\let\\real\@dbflt=\\\@dbflt \\let\\end\@dblfloat=\\end\@float\n" .
    "\\let\\\@largefloatcheck=\\relax\n" .
    "\\def\\\@dbflt{\\let\\\@savefreelist\\\@freelist\\real\@dbflt}\n" .
    "\\def\\adjustnormalsize{\\def\\normalsize{\\mathsurround=0pt \\realnormalsize\n" .
    " \\parindent=0pt\\abovedisplayskip=0pt\\belowdisplayskip=0pt}\\normalsize}%\n" .
    "\\def\\lthtmltypeout#1{{\\let\\protect\\string\\immediate\\write\\lthtmlwrite{#1}}}%\n" .
    "\\newcommand\\lthtmlhboxmathA{\\adjustnormalsize\\setbox\\sizebox=\\hbox\\bgroup}%\n" .
    "\\newcommand\\lthtmlvboxmathA{\\adjustnormalsize\\setbox\\sizebox=\\vbox\\bgroup%\n".
    " \\let\\ifinner=\\iffalse }%\n" .
    "\\newcommand\\lthtmlboxmathZ{\\\@next\\next\\\@currlist{}{\\def\\next{\\voidb\@x}}%\n" .
#    " \\expandafter\\box\\next\\edef\\next{\\egroup\\def\\noexpand\\thiseqn{\\theequation}}\\next
    " \\expandafter\\box\\next\\egroup}%\n" .
    "\\newcommand\\lthtmlmathtype[1]{\\def\\lthtmlmathenv{#1}}%\n" .
    "\\newcommand\\lthtmllogmath{\\lthtmltypeout{l2hSize %\n" .
    ":\\lthtmlmathenv:\\the\\ht\\sizebox::\\the\\dp\\sizebox::\\the\\wd\\sizebox.\\preveqno}}%\n"
```

```
242     "\\newcommand\\lthtmlfigureA[1]{\\let\\\\@savefreelist\\\\@freelist
243        \\lthtmlmathtype{#1}\\lthtmlvboxmathA}%\n" .
244     "\\newcommand\\lthtmlfigureZ{\\lthtmlboxmathZ\\lthtmllogmath\\copy\\sizebox
245        \\global\\let\\\\@freelist\\\\@savefreelist}%\n" .
246     "\\newcommand\\lthtmldisplayA[1]{\\lthtmlmathtype{#1}\\lthtmlvboxmathA}%\n" .
247     "\\newcommand\\lthtmldisplayB[1]{\\edef\\preveqno{(\\theequation)}%\n" .
248     "  \\lthtmldisplayA{#1}\\let\\\\@eqnnum\\relax}%\n" .
249     "\\newcommand\\lthtmldisplayZ{\\lthtmlboxmathZ\\lthtmllogmath\\lthtmlsetmath}%\n" .
250     "\\newcommand\\lthtmlinlinemathA[1]{\\lthtmlmathtype{#1}\\lthtmlhboxmathA" .
251     "  \\vrule height1.5ex width0pt }%\n" .
252     "\\newcommand\\lthtmlinlineA[1]{\\lthtmlmathtype{#1}\\lthtmlhboxmathA}%\n" .
253     "\\newcommand\\lthtmlinlineZ{\\egroup\\expandafter\\ifdim\\dp\\sizebox>0pt %\n" .
254     "  \\expandafter\\centerinlinemath\\fi\\lthtmllogmath\\lthtmlsetinline}\n" .
255     "\\newcommand\\lthtmlinlinemathZ{\\egroup\\expandafter\\ifdim\\dp\\sizebox>0pt %\n" .
256     "  \\expandafter\\centerinlinemath\\fi\\lthtmllogmath\\lthtmlsetmath}\n" .
257     "\\def\\lthtmlsetinline{\\hbox{\\vrule width.1em\\vtop{\\vbox{%\n" .
258     "  \\kern.1em\\copy\\sizebox}\\ifdim\\dp\\sizebox>0pt\\kern.1em\\else\\kern.3pt\\fi\n" .
259     "  \\ifdim\\hsize>\\wd\\sizebox \\hrule depth1pt\\fi}}}\n" .
260     "\\def\\lthtmlsetmath{\\hbox{\\vrule width.1em\\setbox1=\\vtop{\\vbox{%\n" .
261     "  \\kern.1em\\kern$kern pt\\hbox{\\hglue.17em\\copy\\sizebox\\hglue$kern pt}}\\kern.3pt%\n"
262 #   "  \\ifdim\\dp\\sizebox>0pt\\kern.1em \\kern$kern pt\\fi}%\n" .
263     "  \\ifdim\\dp\\sizebox>0pt\\kern.1em\\fi \\kern$kern pt%\n" .
264     "  \\ifdim\\hsize>\\wd\\sizebox \\hrule depth1pt\\fi}\\message{ht\\the\\ht1: dp\\the\\dp1}\\b
265     "\\def\\centerinlinemath{%\\dimen1=\\ht\\sizebox\n" .
266 #   "  \\ifdim\\dimen1<\\dp\\sizebox \\ht\\sizebox=\\dp\\sizebox\n" .
267     "  \\dimen1=\\ifdim\\ht\\sizebox<\\dp\\sizebox \\dp\\sizebox\\else\\ht\\sizebox\\fi\n" .
268 #    "   \\vrule width0pt depth\\dp\\sizebox \n" .
269 #   "  \\else \\dp\\sizebox=\\ht\\sizebox \\fi}\n\n" .
270     "  \\advance\\dimen1by.5pt \\vrule width0pt height\\dimen1 depth\\dimen1 \n".
271     " \\dp\\sizebox=\\dimen1\\ht\\sizebox=\\dimen1\\relax}\n\n" .
272     "\\def\\lthtmlcheckvsize{\\ifdim\\ht\\sizebox<\\vsize\\expandafter\\vfill\n" .
273     "  \\else\\expandafter\\vss\\fi}%\n" .
274 #    "\\def\\\\@enddocumenthook{\\ifnum\\count0>1 \\ifvoid\\\\@cclv\\penalty-\\\\@MM\\fi\\fi}\n" .
275     "\\makeatletter \\tracingstats = 1 \n" .
276     $LaTeXmacros . "\n"  # macros defined in extension files
277 #    "\\usepackage{lthimages}\n" .
278     . (($LATEX_DUMP)? "\\latexdump\n" : '')
279     . "\n\\begin{document}\n" .
280     "\\pagestyle{empty}\\thispagestyle{empty}%\n" .
281     "\\lthtmltypeout{latex2htmlLength hsize=\\the\\hsize}%\n" .
282     "\\lthtmltypeout{latex2htmlLength vsize=\\the\\vsize}%\n" .
283     "\\lthtmltypeout{latex2htmlLength hoffset=\\the\\hoffset}%\n" .
284     "\\lthtmltypeout{latex2htmlLength voffset=\\the\\voffset}%\n" .
285     "\\lthtmltypeout{latex2htmlLength topmargin=\\the\\topmargin}%\n" .
286     "\\lthtmltypeout{latex2htmlLength topskip=\\the\\topskip}%\n" .
287     "\\lthtmltypeout{latex2htmlLength headheight=\\the\\headheight}%\n" .
288     "\\lthtmltypeout{latex2htmlLength headsep=\\the\\headsep}%\n" .
289     "\\lthtmltypeout{latex2htmlLength parskip=\\the\\parskip}%\n" .
290     "\\lthtmltypeout{latex2htmlLength oddsidemargin=\\the\\oddsidemargin}%\n" .
291     "\\makeatletter\n" .
292     "\\if\\@twoside\\lthtmltypeout{latex2htmlLength evensidemargin=\\the\\evensidemargin}%\n" .
293     "\\else\\lthtmltypeout{latex2htmlLength evensidemargin=\\the\\oddsidemargin}\\fi%\n" .
294     "\\makeatother\n"
295     . "$contents\n"
```

```
#    "\\clearpage\n" .
    . "\\end{document}";
}
```
⟨/foils⟩

Since they are different, we must prevent loading `foilhtml.perl` if we use `foils.perl`. Of course, we asked in the user guide *not* to install `foilhtml.perl` for 97-1, but one cannot be sure anybody will RTFM...

⟨∗foils − 97 | foils⟩
```
 $DONT_INCLUDE = "foilhtml:".$DONT_INCLUDE;
```

And now the ignored FoilTEX commands. They are common for both versions of LATEX2HTML.

⟨/foils − 97 | foils⟩
⟨∗foilhtml | foils − 97 | foils⟩
```
&ignore_commands( <<_IGNORED_CMDS_);
portraitfoils
lanscapefoils
MyLogo # {}
Restriction # {}
LogoOff
LogoOn
rightfooter # {}
rightheader # {}
leftheader # {}
newnonfloat #{} #{}
foilheadskip # &ignore_numeric_argument
abovefloatskip # &ignore_numeric_argument
captionwidth # &ignore_numeric_argument
titleauthorskip # &ignore_numeric_argument
authorauthorskip # &ignore_numeric_argument
authordateskip # &ignore_numeric_argument
dateabstractskip # &ignore_numeric_argument
zerolistvertdimens
_IGNORED_CMDS_
```

The famous last line:
```
 1; # This should be the last line
```
⟨/foilhtml | foils − 97 | foils⟩

8 LATEX style

8.1 Initialization and processing options

First, let us ask (not so) nicely for LATEX

⟨∗package⟩
```
\NeedsTeXFormat{LaTeX2e}
```

`\@pagesintoc` Now let us check whether we need pages in toc:

```
\newif\if@pagesintoc
\DeclareOption{pagesintoc}{\@pagesintoctrue}
\DeclareOption{nopagesintoc}{\@pagesintocfalse}
\ExecuteOptions{nopagesintoc}
```

`\c@secnumdepth` We need to set `secnumdepth` before loading the configuration file:

```
\setcounter{secnumdepth}{0}
```

Now let us check the configuration file:

```
\InputIfFileExists{foilhtml.cfg}{%
  \typeout{Loading configuration file foilhtml.cfg}}{%
  \typeout{Configuration file foilhtml.cfg is not found. Using default
  options.}}
```

And now we construct a sample configuration file

```
⟨/package⟩
⟨*cfg⟩
%%
%% The default options go here
%%
\ExecuteOptions{nopagesintoc}
%%
\AtEndOfPackage{\setcounter{secnumdepth}{0} % Customization goes here
}
⟨/cfg⟩
⟨*package⟩
```

`\if@portraitfoils` The flag `\if@portraitfoils` controls what kind of foils we produce for sectioning commands: portrait or rotated. We put it here because we want the global option `landscape` to work intelligently...

```
\newif\if@portraitfoils
\@portraitfoilstrue
\DeclareOption{landscape}{\@portraitfoilsfalse}
```

Processing options:

```
\ProcessOptions
```

8.2 Checking the caller

Let us check where we are. The package could be loaded either from `foils` document class or from `article` (or other) documentclass (e. g. from `images.tex`). We don't want to do anything in the latter case. So let us check whether some FoilTEX command is defined:

```
\@ifundefined{foilhead}{%
```

If we are *not* called by FoilTEX, we should disable FoilTEX-specific commands. The LaTeX2HTML script does not allow them in the document body, but, unfortunately, leaves them in the preamble. So we take care of them:

```
 \def\portraitfoils{\relax}%
 \def\lanscapefoils{\relax}%
 \def\MyLogo{\@gobble}%
 \def\Restriction{\@gobble}%
 \def\LogoOff{\relax}%
 \def\LogoOn{\relax}%
 \def\rightfooter{\@gobble}%
 \def\rightheader{\@gobble}%
 \def\leftheader{\@gobble}%
 \def\newnonfloat{\@gobbletwo}%
 \newdimen\foilheadskip
 \newdimen\abovefloatskip
 \newdimen\captionwidth
 \newdimen\titleauthorskip
```

```
\newdimen\authorauthorskip
\newdimen\authordateskip
\newdimen\dateabstractskip
\def\zerolistvertdimens{\relax}%
}{%
```

8.3 Rotating commands

\portraitfoils
\portraitfoils*
\landscapefoils
\landscapefoils*

The rotating commands set `\if@portraitfoils` and (for unstarred variants) eject the foil using `\foilhead`

```
\newcommand{\portraitfoils}{%
  \@ifstar{\@portraitfoilstrue}{%
  \@portraitfoilstrue\foilhead[-\foilheadskip]{}}}
\newcommand{\landscapefoils}{%
  \@ifstar{\@portraitfoilsfalse}{%
  \@portraitfoilsfalse\rotatefoilhead[-\foilheadskip]{}}}
```

8.4 Sectioning commands

The stuff here is mostly stolen from `classes.dtx` [3]—however, we implement the high level sectioning commands through `\foilhead`.

\chaptermark

First, mark commands (anybody uses marks for foils out here?) The file `latex.ltx` defines all of them but `\chaptermark`, so we will oblige...

```
\newcommand*\chaptermark[1]{}
```

\c@part
\c@chapter
\c@section
\c@subsection
\c@subsubsection
\c@paragraph
\c@subparagraph

Now the counters: (almost verbatim from `classes.dtx`):

```
\newcounter {part}
\newcounter {chapter}
\newcounter {section}
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
```

\thepart
\thechapter
\thesection
\thesubsection
\thesubsubsection
\theparagraph
\thesubparagraph

```
\renewcommand \thepart {\@Roman\c@part}
\renewcommand \thechapter {\@arabic\c@chapter}
\renewcommand \thesection {\@arabic\c@section}
\renewcommand\thesubsection   {\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection .\@arabic\c@subsubsection}
\renewcommand\theparagraph    {\thesubsubsection.\@arabic\c@paragraph}
\renewcommand\thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
```

\part
\part*

The commands `\part` and `\part*` are implemented through `\secdef` macro:

```
\newcommand\part{%
  \@afterindentfalse\clearpage
  \secdef\@part\@spart}
```

\@part
\@spart

The actual typesetting is done by `\@part` and `\@spart` commands. They produce foils with just the title on them in `\Huge\bfseries` font. By the way, we still print *Logo* and *Restriction* on them. Maybe we should set them to null?

```
\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >-2\relax
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
      \if@portraitfoils
         \foilhead{\huge\bfseries\partname~\thepart\\[1ex]%
         \Huge\bfseries #2}%
      \else
         \rotatefoilhead{\huge\bfseries\partname~\thepart\\[1ex]%
         \Huge\bfseries #2}%
      \fi
    \else
      \addcontentsline{toc}{part}{#1}%
      \if@portraitfoils
         \foilhead{\Huge\bfseries #2}%
      \else
         \rotatefoilhead{\Huge\bfseries #2}%
      \fi
    \fi
   \markboth{}{}%
   \clearpage}
\def\@spart#1{%
   \if@portraitfoils
      \foilhead{\Huge\bfseries #1}%
   \else
      \rotatefoilhead{\Huge\bfseries #1}%
   \fi
   \markboth{}{}%
   \clearpage}
```

`\chapter` `\chapter*` The commands `\chapter` and `\chapter*` are implemented through `\secdef` macro:

```
\newcommand\chapter{%
   \@afterindentfalse\clearpage
   \secdef\@chapter\@schapter}
```

`\@chapter` `\@schapter` The `\@chapter` and `\@schapter` commands produce foils with just the title on them in `\huge\bfseries` font. Once again, we still print *Logo* and *Restriction* on them. Maybe we should set them to null?

```
\def\@chapter[#1]#2{%
    \ifnum \c@secnumdepth >-1\relax
      \refstepcounter{chapter}%
      \addcontentsline{toc}{chapter}{\thechapter\hspace{1em}#1}%
      \if@portraitfoils
         \foilhead{\LARGE\bfseries\chaptername~\thechapter\\[1ex]%
         \huge\bfseries #2}%
      \else
         \rotatefoilhead{\LARGE\bfseries\chaptername~\thechapter\\[1ex]%
         \huge\bfseries #2}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{#1}%
      \if@portraitfoils
         \foilhead{\huge\bfseries #2}%
```

```
      \else
         \rotatefoilhead{\huge\bfseries #2}%
      \fi
    \fi
   \markboth{}{}%
   \clearpage}
\def\@schapter#1{%
   \if@portraitfoils
      \foilhead{\huge\bfseries #1}%
   \else
      \rotatefoilhead{\huge\bfseries #1}%
   \fi
   \markboth{}{}%
   \clearpage}
```

`\section` `\section*` Once again, `\secdef`:

```
\newcommand\section{%
   \@afterindentfalse\clearpage
   \secdef\@section\@ssection}
```

`\if@subnewfoil` All previous commands had titles on separate foils. Now the situation is different: we want the text follow the heading. The only problem we have is with subsection: should it start a new foil or not? The decision is the following: subsection does not start a new foil only if it follows a section title. To implement this we use the mechanism similar to `\@nobreak` in standard classes: a flag `\if@subnewfoil` is initially set to true. Each section command sets it to false, but any new paragraph will set it to true.

```
\newif\if@subnewfoil
\@subnewfoiltrue
```

`\@section` `\@ssection` The commands `\section` and `\section*` produce new foils with the section titles on them. Also, they set `\if@subnewfoil` to false, so new subsection will start on the same foil if no new text is present.

```
\def\@section[#1]#2{%
    \ifnum \c@secnumdepth >0\relax
      \refstepcounter{section}%
      \addcontentsline{toc}{section}{\thesection\hspace{1em}#1}%
      \markboth{\MakeUppercase{\thesection~#2}}{%
         \MakeUppercase{\thesection~#2}}%
      \if@portraitfoils
         \foilhead{\Large\bfseries\thesection~#2}%
      \else
         \rotatefoilhead{\Large\bfseries\thesection~#2}%
      \fi
    \else
      \addcontentsline{toc}{section}{#1}%
      \markboth{\MakeUppercase{#2}}{\MakeUppercase{#2}}%
      \if@portraitfoils
         \foilhead{\Large\bfseries #2}%
      \else
         \rotatefoilhead{\Large\bfseries #2}%
      \fi
    \fi
```

```
    \global\@subnewfoilfalse\everypar{\global\@subnewfoiltrue\everypar{}}}
\def\@ssection#1{%
    \if@portraitfoils
       \foilhead{\Large\bfseries #1}%
    \else
       \rotatefoilhead{\Large\bfseries #1}%
    \fi
    \markboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}%
    \global\@subnewfoilfalse\everypar{\global\@subnewfoiltrue\everypar{}}}
```

\subsection
\subsection*

We start a new page unless immediately after a section title:

```
\newcommand\subsection{%
   \@afterindentfalse\if@subnewfoil\clearpage\fi\global\@subnewfoiltrue
   \secdef\@subsection\@ssubsection}
```

\@subsection
\@ssubsection

```
\def\@subsection[#1]#2{%
     \ifnum \c@secnumdepth >1\relax
       \refstepcounter{subsection}%
       \addcontentsline{toc}{subsection}{\thesubsection\hspace{1em}#1}%
       \begin{center}\large\bfseries\thesubsection~#2\end{center}%
     \else
       \addcontentsline{toc}{subsection}{#1}%
       \begin{center}\large\bfseries #2\end{center}%
     \fi}
\def\@ssubsection#1{%
     \begin{center}\large\bfseries #1\end{center}}%
```

All lower lever headings are shamelessly copied from `classes.dtx` [3] with one addition: they all set `\@subnewfoiltrue`. The quotes are from `classes.dtx`.

\subsubsection

> This gives a normal heading with white space above and below the heading, the title set in `\normalsize\bfseries`, and no indentation on the first paragraph.

```
\newcommand\subsubsection{\global\@subnewfoiltrue%
                                \@startsection{subsubsection}{3}{\z@}%
                                {-3.25ex\@plus -1ex \@minus -.2ex}%
                                {1.5ex \@plus .2ex}%
                                {\normalfont\normalsize\bfseries}}
```

\paragraph

> This gives a run-in heading with white space above and to the right of the heading, the title set in `\normalsize\bfseries`.

```
\newcommand\paragraph{\global\@subnewfoiltrue%
                               \@startsection{paragraph}{4}{\z@}%
                               {3.25ex \@plus1ex \@minus.2ex}%
                               {-1em}%
                               {\normalfont\normalsize\bfseries}}
```

\subparagraph

> This gives an indented run-in heading with white space above and to the right of the heading, the title set in `\normalsize\bfseries`.

```
\newcommand\subparagraph{\global\@subnewfoiltrue%
                              \@startsection{subparagraph}{5}{\parindent}%
                              {3.25ex \@plus1ex \@minus .2ex}%
```

```
518                                 {-1em}%
519                                 {\normalfont\normalsize\bfseries}}
```

8.5 Table of Contents

First, the typesetting parameters. Once again, from `classes.dtx`:

`\@pnumwidth` `\@tocrmarg` `\@dotsep`

This command uses the following three parameters, which are set with a `\newcommand` (so em's can be used to make them depend upon the font).

`\@pnumwidth` The width of a box in which the page number is put.

`\@tocrmarg` The right margin for multiple line entries. One wants `\@tocrmarg` $\geq$ `\@pnumwidth`

`\@dotsep` Separation between dots, in mu units. Should be defined as a number like 2 or 1.7

```
520 \newcommand\@pnumwidth{1.55em}
521 \newcommand\@tocrmarg{2.55em}
522 \newcommand\@dotsep{4.5}
523 \setcounter{tocdepth}{3}
```

`\@nopagetocline` The standard LaTeX kernel [4] defines `\@dottedtocline` for formatting the Table of Contents entries. We define an analogous command for contents line *without* page number. Its parameters are the same as for `\@dottedtocline` but the last one (page).

```
524 \def\@nopagetocline#1#2#3#4{%
525   \ifnum #1>\c@tocdepth \else
526     \vskip \z@ \@plus.2\p@
527     {\leftskip #2\relax
528      \parindent #2\relax\@afterindenttrue
529      \interlinepenalty\@M
530      \leavevmode
531      \@tempdima #3\relax
532      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
533      {#4}\hfil%
534      \par}%
535   \fi}
```

`\tableofcontents` We implement table of contents as a new `\section*`:

```
536 \newcommand\tableofcontents{%
537 \section*{\contentsname}%
538 \@starttoc{toc}%
539 \clearpage\global\@subnewfoiltrue}%
```

The actual typesetting is done by commands `\l@`⟨*name*⟩. Here we define them. Once again, verbatim from `classes.dtx` plus check for `\if@pagesintoc`.

`\l@part`

First we make sure that if a pagebreak should occur, it occurs *before* this entry. Also a little whitespace is added and a group begun to keep changes local.

```
540 \newcommand*\l@part[2]{%
541   \ifnum \c@tocdepth >-2\relax
```

```
    \addpenalty\@highpenalty
    \addvspace{2.25em \@plus\p@}%
    \begingroup
```

The macro `\numberline` requires that the width of the box that holds the part number is stored in LaTeX's scratch register `\@tempdima`. Therefore we put it there.

```
      \setlength\@tempdima{3em}%
```

The we set `\parindent` to 0pt and use `\rightskip` to leave enough room for the pagenumbers. To prevent overfull box messages the `\parfillskip` is set to a negative value.

```
      \parindent \z@
      \if@pagesintoc
        \rightskip \@pnumwidth
        \parfillskip -\@pnumwidth
      \fi
```

Unlike standard classes, the entry is set in `\Large` bold font (they use `\large`)

We make sure to leave vertical mode, set the part title and add the pagenumber, set flush right.

```
      {\leavevmode
        \Large \bfseries #1\hfil
        \if@pagesintoc\hb@xt@\@pnumwidth{\hss #2}\fi
      }\par
```

Prevent a pagebreak immediately after this entry, but use `\everypar` to reset the `\if@nobreak` switch. Finally we close the group.

```
       \nobreak
        \global\@nobreaktrue
        \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
```

`\l@chapter`

This macro formats the entries in the table of contents for chapters. It is very similar to `\l@part`

First we make sure that if a pagebreak should occur, it occurs *before* this entry. Also a little whitespace is added and a group begun to keep changes local.

```
\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip 1.0em \@plus\p@
```

The macro `\numberline` requires that the width of the box that holds the part number is stored in LaTeX's scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@
      \if@pagesintoc
        \rightskip \@pnumwidth
        \parfillskip -\@pnumwidth
      \fi
```

Then we leave vertical mode and switch to a large bold font.

```
      \leavevmode \large\bfseries
```

Because we do not use `\numberline` here, we have do some fine tuning 'by hand', before we can set the entry. We discourage but not disallow a pagebreak immediately after a chapter entry.

```
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \if@pagesintoc\nobreak\hb@xt@\@pnumwidth{\hss #2}\fi\par
      \penalty\@highpenalty
    \endgroup
  \fi}
```

`\l@section` Our class is a sort of a cross between article and book/report. Therefore we want the special formatting used in the article class...

In the article document class the entry in the table of contents for sections looks much like the chapter entries for the report and book document classes.

First we make sure that if a pagebreak should occur, it occurs *before* this entry. Also a little whitespace is added and a group begun to keep changes local.

```
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
```

The macro `\numberline` requires that the width of the box that holds the part number is stored in LaTeX's scratch register `\@tempdima`. Therefore we put it there. We begin a group, and change some of the paragraph parameters.

```
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@
      \if@pagesintoc
         \rightskip \@pnumwidth
         \parfillskip -\@pnumwidth
      \fi
```

Then we leave vertical mode and switch to a bold font.

```
     \leavevmode \bfseries
```

Because we do not use `\numberline` here, we have do some fine tuning 'by hand', before we can set the entry. We discourage but not disallow a pagebreak immediately after a chapter entry.

```
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \if@pagesintoc\nobreak\hb@xt@\@pnumwidth{\hss #2}\fi\par
    \endgroup
  \fi}
```

`\l@subsection` `\l@subsubsection` `\l@paragraph` `\l@subparagraph`

Lower level headings are made through `\@dottedtocline` and `\@nopagetocline`:

```
\newcommand*\l@subsection[2]{%
  \if@pagesintoc
    \@dottedtocline{2}{1.5em}{2.3em}{#1}{#2}%
  \else
    \@nopagetocline{2}{1.5em}{2.3em}{#1}%
  \fi}
\newcommand*\l@subsubsection[2]{%
  \if@pagesintoc
    \@dottedtocline{3}{3.8em}{3.2em}{#1}{#2}%
  \else
    \@nopagetocline{3}{3.8em}{3.2em}{#1}%
  \fi}
\newcommand*\l@paragraph[2]{%
  \if@pagesintoc
    \@dottedtocline{4}{7.0em}{4.1em}{#1}{#2}%
  \else
    \@nopagetocline{4}{7.0em}{4.1em}{#1}%
  \fi}
\newcommand*\l@subparagraph[2]{%
  \if@pagesintoc
    \@dottedtocline{5}{10em}{5em}{#1}{#2}%
  \else
    \@nopagetocline{5}{10em}{5em}{#1}%
  \fi}
```

8.6 Bibliography

`\thebibliography` Jim Hafner [2] reimplemented `\thebibliography`. He deleted the `\section*` command. Now we put it back!

```
\renewenvironment{thebibliography}[1]{
 \section*{\refname}
 \list{\@biblabel{\arabic{enumiv}}}%
 {\settowidth\labelwidth{\@biblabel{#1}}%
 \leftmargin\labelwidth
 \advance\leftmargin\labelsep
 \if@openbib
   \advance\leftmargin\bibindent
   \itemindent -\bibindent
   \listparindent \itemindent
   \parsep \z@
 \fi
```

```
  \usecounter{enumiv}%
  \let\p@enumiv\@empty
  \renewcommand\theenumiv{\arabic{enumiv}}}
  \if@openbib
    \renewcommand\newblock{\par}
  \else
    \renewcommand\newblock{\hskip .11em \@plus .33em \@minus -.07em}
  \fi
  \sloppy\clubpenalty4000\widowpenalty4000%
  \sfcode`\.=\@m\relax}%
  {\def\@noitemerr{\@latex@warning{Empty `thebibliography' environment}}%
  \endlist}
```

8.7 Names

\contentsname
\partname
\chaptername

We add names for Table of Contents, chapters and parts.

```
\newcommand\contentsname{Outline}%
\newcommand\partname{Part}%
\newcommand\chaptername{Chapter}%
```

8.8 Wrapping up

And now let us finish this `\if@undefined` stuff

```
}
```

References

[1] Nikos Drakos, Jens Lippmann, and Ross Moore The LaTeX2HTML translator. CTAN, 1996, 1997.

[2] Jim Hafner. The FoilTeX class package. CTAN, 1997.

[3] Leslie Lamport, Frank Mittelbach, and Johannes Braams. Standard document classes for LaTeX version 2e. CTAN, 1997.

[4] Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Chris Rowley, and Rainer Schöpf. ltdefns.dtx. CTAN, 1997.