The FoilTEX class package*

Jim Hafner
IBM Research Division
Almaden Research Center, K53-B2
650 Harry Road
San Jose, CA 95120-6099
hafner@almaden.ibm.com

Printed on August 3, 1999

Preface

The original FoilTEX was a LATEX-like system for typesetting foils which used its own format file. This new version has converged into a LATEX 2_ε class package. Its features include simplicity of use, compatibility with LATEX, large sans serif font as default, extra macros to start foils with bold headings and special mechanisms to control the footer and header. The primary enhancement, besides the gain obtained by being just a class file for LATEX 2_ε, is the addition of simple tools to rotate either the entire set of foils to landscape mode or to rotate individual foils. Because this is integrated with LATEX 2_ε, the use of color and other fancy stuff comes automatically. No additional utilities are required.

This document is the user guide for FoilTEX and describes its basic features and components.

There are restrictions on the use of FoilTEX. Please refer to Section 8 for more information[1].

TEX is a trademark of the American Mathematical Society. PostScript is a trademark of Adobe Systems, Incorporated.

*This package has version number v2.1.3, last revised 1998/08/15.

[1]Copyright ©1995,1997,1998 by International Business Machines Corporation.

Contents

1 The FoilTEX package **4**
- 1.1 Package differences with original FoilTEX . . . 5
- 1.2 Additional Features in Version 2.1 . . . 5
- 1.3 Additional Features in Version 2.1.3 . . . 5
- 1.4 Related Packages on CTAN . . . 6

2 Getting started: the `\documentclass` command **6**

3 The basic features **7**
- 3.1 The class options . . . 7
  - 3.1.1 Default options and `foiltex.cfg` . . . 8
  - 3.1.2 New Options in Version 2.1 . . . 8
  - 3.1.3 New Options in Version 2.1.3 . . . 9
- 3.2 The `\maketitle` command . . . 9
- 3.3 The `\foilhead` and `\rotatefoilhead` macros . . . 9
- 3.4 The `\MyLogo` and `\Restriction` macros . . . 10
  - 3.4.1 Toggling the logo . . . 11
- 3.5 The other three corners of the page . . . 11
- 3.6 Header and footer rules and `fancyheadings.sty` . . . 11
- 3.7 Predefined `Theorem` and `Proof` environments . . . 12
- 3.8 Mathematics in bold typeface . . . 12
- 3.9 Hyphenation and raggedright . . . 13
- 3.10 Non-floating floats . . . 13
- 3.11 User adjustable dimensions . . . 14
- 3.12 Differences with LATEX . . . 14
- 3.13 Future versions . . . 14

4 Fonts and their sizes **15**

5 Making color foils **16**

6 Using PostScript fonts **16**

7 Installing FoilTEX **16**

8 Usage restrictions **17**
- 8.1 Experimental Software Disclaimer . . . 17
- 8.2 User Agreement . . . 17

9 Acknowledgements, requests and help **18**

10 A sample file **18**

11 Description of the code in the FoilTEX package **26**
- 11.1 The package options . . . 26
- 11.2 Some basic macros . . . 29
- 11.3 List environments: spacing . . . 31
- 11.4 List environments: structure . . . 33
- 11.5 Page layout parameters . . . 34

11.6 The title page . . . 37
11.7 Page style . . . 38
11.8 Bibliography or References . . . 40
11.9 Non-floating floats . . . 41
11.10 The foil title macros and foil rotation . . . 43
11.11 Theorem-like environments . . . 47
11.12 Boldface mathematics . . . 48

12 The class option files **49**
12.1 The `17pt` option . . . 49
12.2 The `20pt` option . . . 50
12.3 The `25pt` option . . . 50
12.4 The `30pt` option . . . 51
12.5 The `shortform` option . . . 52

13 Compatibility mode **53**

14 Font definitions **53**
14.1 Font macro utilities . . . 54
14.2 The font family and shape declarations: `OT1` . . . 55
14.3 The font family and shape declarations: `T1` . . . 56
14.4 The font family and shape declarations: `OM*` . . . 58
14.5 The font selection mechanisms . . . 59
14.6 Math and symbol fonts . . . 59

This document describes the FoilTEX system (version 2.1.3) for making foils (slides, transparencies, overheads, etc.). In contrast to the previous version which required its own format file, this new version is fully integrated into LATEX 2ε as a class package. Like the earlier version, it is much simpler to use than SLITEX (or the SLITEX class package), and should be very easy for typical LATEX users to master. This document tells you about the special features it has and the extra macros that have been added. It also includes installation instructions which are now much easier than version 1.0.1. Other than some differences with font availability (sizes not families), it should work under any implementation of LATEX 2ε. It is assumed that you are already familiar with LATEX. If not, you should get the books *LATEX: A Document Preparation System* 2nd edition, by Leslie Lamport [3], and *The LATEX Companion*, by Michel Goossens, Frank Mittelbach and Alexander Samarin [1].

1 The FoilTEX package

The FoilTEX package consists of the files listed in Table 1 on top of the basic implementation of LATEX 2ε. You will probably get it packaged with just the first three files. The others are generated by running LATEX on the `foiltex.ins` file (see Section 7 for more details). Documentation (both user guide and code description) can be obtained by running LATEX on the `foiltex.dtx` file.

Table 1: The files in the FoilTEX package

`readme.flt`	a simple readme file
`foiltex.ins`	driver file for unpacking
`foiltex.dtx`	user guide and documented source
`foils.cls`	main class file
`fltfonts.def`	font definition file
`foils.sty`	compatibility mode style file
`foil17.clo`	17pt class option file
`foil20.clo`	20pt class option file
`foil25.clo`	25pt class option file
`foil30.clo`	30pt class option file
`foilshrt.clo`	shortform class option file (new in 2.1.3)
`sampfoil.tex`	sample FoilTEX file

The main class file, `foils.cls`, together with `fltfonts.def` form the heart of FoilTEX. The former defines the basic set of macros that are used (and includes a request to input the latter) and the latter defines all the fonts used by FoilTEX. Font selection is done via the LATEX 2ε NFSS scheme. The `foils.sty` file is just a wrapper for `foils.cls` in compatibility mode.

The next group of files are the class option files that change the default font sizes and some list spacing parameters. See Sections 2 and 4 for more information about these files.

The last file is a relatively extensive sample file. It demonstrates some of the basic features and is self-documented. If you don't like reading documentation (like this) you can probably do pretty well with FoilTEX simply by browsing through

the sample file and looking at its output. To see its output (after installation), simply type

```
latex sampfoil
```

This creates the `.dvi` and `.aux` files. You will need to rerun this step to get the cross-referencing right. You can then preview or print this as you would any other `.dvi` file.

1.1 Package differences with original FoilTEX

The original FoilTEX package (version 1.0.1) contained a number of similar files plus many more. Because FoilTEX is now integrated with LATEX, the additional files are no longer necessary. For example, we no longer include the `colordvi.sty` files since this is handled in a much cleaner and more integrated way with the `color.sty` of the `graphics` package. We also don't include anything for PostScript fonts since that is nicely handled with the PSNFSS packages. Because we are integrated with LATEX, you can even use the `t1enc.sty` package to access the EC fonts (see below). (This has not been extensively tested because the author doesn't use them, but it did work on some smaller test cases, e.g., `sampfoil.tex`.) There are no more system-specific files since that is handled directly by the LATEX installation. Similarly, we don't need to include a file for using the $\mathcal{AMS}$-TEX fonts. Finally, the "This is *not* LATEX" warning has been removed as it is completely irrelevant now.

1.2 Additional Features in Version 2.1

There are very few new features in this version. The biggest change is the support for the new EC fonts (and backlevel support for the DC fonts version 1.3 which was not included in version 2.0.1 of FoilTEX). An alternative to the `t1enc.sty` package for accessing the EC fonts (with the `T1` encoding) is via the line

```
\usepackage[T1]{fontenc}
```

in the preamble. See Section 3.1.2 for more information about using these fonts. Note, that we have no support for the TC (TEX Companion) fonts, since we don't understand how they are used.

The only other changes are either small bug fixes (most notably with the footline — thanks to Angelika Schneider for finding two!) and a re-encoding of the `fltfonts.def` file so that it is smaller. Version 2.1.1 has a small bug fix to the order of list environment definitions.

1.3 Additional Features in Version 2.1.3

Version 2.1.2 probably never made it outside of IBM, but it included a number of other bug fixes. Version 2.1.3 has the following additional features:

- Addition of a `\captionfraction` command to set the width of the captions as a fraction of the text width. With these, we have deprecated `\captionwidth`. See Section 3.11. This was found by John Wu.
- Rotation support for DVIPSONE from Y&Y, Inc. My thanks to them for passing on the necessary code changes.

- Addition of `shortform` class option, which can be used to generate a document which takes up much less space (paper, mostly). This was added in response to some requests by Neal Beck who asked for something like this over a year ago.
- Bug fix for the `figure*` and `table*` and related environments (these were supposed to be unnumbered but always had a number). Thanks to Scott Berger for finding this problem.
- Bug fix for the `\big` and related operators. This bug was reported by Hans-Peter Zehrfeld (thanks).
- Additional support for other paper sizes, as provided by Owen Kelly. These new sizes are `ledgerpaper`, `legalpaper`, `a3paper`, `a2paper`, `a1paper`. The second of these had been suppressed in earlier versions. It is restored here on request.

1.4 Related Packages on CTAN

A number of people have contributed to the FoilTEX family (in some ways indirectly). For example, check out `epslatex.ps` by Keith Reckdahl which describes a way to put a EPS Logo on every page and still manage to keep the size of the reluting PostScript file small (effectively only loading the real EPS logo only once as a header file). You can find this document in the `info` directory on your favorite CTAN.

Another package you can find on CTAN is the FoilHTML package by Boris Veytsman. It provides extensions to `latex2html` to handle FoilTEX's extra macross.

If you want to put multiple foils (reduced) on a single page, you can use the `psnup` utility (found in most good TEX bundles).

2 Getting started: the `\documentclass` command

`\documentclass{foils}`

To create a FoilTEX document, you edit a LATEX file. Instead of the standard LATEX options specified in the `\documentclass` command, you should use

`\documentclass[`*opts*`]{foils}`

Here, the *opts* list can include most of the standard LATEX class file options plus a few more FoilTEX-specific ones that are documented in Section 3.1.

`17pt,20pt,25pt,30pt`
`shortform`

By default, `foils.cls` loads `foil20.clo` and sets up the normal size fonts at 20pt. Analogous to LATEX's `11pt` and `12pt` style options, FoilTEX has `25pt`, `30pt` and `17pt` options. For example, to make normal size at 25pt the command

`\documentclass[25pt,`*opts*`]{foils}`

will do the trick. The default `20pt` *is* an acceptable option, though it is redundant. Besides the `pt`-type options, version 2.1.3 adds the `shortform` option. This can be used instead to create a document which takes up much less space (paper?) and so is suitable for redistribution. The text is sized similar to the `12pt` option of `article.cls`.

Once you have created your FoilTEX file, run LATEX on it to get your `.dvi` file:

`latex` *filename*

3 The basic features

FoilTEX has a number of (hopefully useful) built-in features. The first is that the basic fonts are in large size, approximately 20pt, (so you do not need to do size changing to get large type). The default font is also sans serif as this (in the opinion of many) looks better on foils than serif fonts like roman. We have implemented LATEX's font and font size changing commands relative to this default. More information about fonts and size changing can be found in Section 4.

In spite of the fact that the basic font is sans serif, the numerals and other symbols from the roman font when used in math mode are still in the roman font. Thus mathematics will look exactly the same as in LATEX (only larger) but numerals in text will appear in sans serif. This is one of its nicer features.

In addition, almost all LATEX macros are available including automatic referencing and citation, footnotes, and itemize (which will probably be very popular for foils). The user is not expected to have to do anything to control font types or size changing, except as might be expected in a typical LATEX document.

sampfoil.tex

The next subsections describe a number of additional macros and features that have been defined to make foilmaking easier. See the `sampfoil.tex` file for a look at how some of these are used.

3.1 The class options

The following standard LATEX class options are *not* supported in FoilTEX because they don't make sense in this environment:

```
a5paper, b5paper, executivepaper, 10pt, 11pt, 12pt,
oneside, twoside, openright, openany, titlepage, notitlepage,
onecolumn, twocolumn.
```

These all default to no-ops and, with the exception of `oneside`, `titlepage` and `onecolumn`, all give a warning message.

35mmSlide
headrule
footrule
dvips|dvipsone
magscalefonts
useDCfonts
*paper
landscape

The following new options are supported in FoilTEX.

`35mmSlide` This sets up the page layout for 7.33in. by 11in. paper, which is about the same aspect ratio as a 35mm slide. You can use this if you plan to reproduce on this medium.

`headrule` This places a rule below the header on every page (except the title page).

`footrule` This places a rule above the footer on every page (except the title page). These two items seemed to be in demand and should preclude the use of `fancyheadings.sty` which can collide with some of FoilTEX's page layout (see Section 3.6 for additional information).

`dvips|dvipsone` In order to support rotated foils and `landscape` in a clean way, we need to issue `\specials` which are driver dependent. Since we only had access to one PostScript driver, namely, Rokicki's `dvips`, we could only support these `\specials` for this one driver. Declaring this option enables this special code. The folks a Y&Y, Inc., have passed on the necessary modifications to support their DVISPONE driver. Since I can't test this, I don't support it, but it's included in case it helps. See also Section 9.

`magscalefonts` When using the `T1` encoding, use mag-scaled versions of the 10pt fonts (with some exceptions) rather than the large design sizes of the EC (or DC) fonts. This is new in version 2.1 (see Section 3.1.2).

`useDCfonts` FoilTEX now supports the official EC fonts with the `T1` encoding. For legacy users, this option will revert back to the last release of the DC fonts (version 1.3). This is new in version 2.1 (see Section 3.1.2).

`*paper` We've added support for these other papers sizes for making posters. These are

`ledgerpaper` at 11in by 17in.

`legalpaper` at 14in by 8.5in.

`a3paper` at 420mm by 297mm.

`a2paper` at 594mm by 420mm.

`a1paper` at 840mm by 594mm.

`landscape` This adjusts the page dimensions by essentially swapping height and width. Typically, it does not force the driver to do the necessary rotation steps. As mentioned, in FoilTEX with the `dvips` option or the `dvipsone` option, both actions are handled automatically.

3.1.1 Default options and `foiltex.cfg`

`foiltex.cfg` The default options for FoilTEX are `letterpaper,20pt,final`. However, you can have your own set of modified default options for FoilTEX simply by having in your inputs path a file called `foiltex.cfg` with a line containing `\ExecuteOptions` statements. E.g., to configure for `dvips` and `landscape` and older DC fonts, your `foiltex.cfg` file might look like

```
\ExecuteOptions{dvips,landscape,useDCfonts}
```

This file is not required, but can help set things up either system-wide or just for your personal use.

3.1.2 New Options in Version 2.1

There are two options new to version 2.1. The first is the `magscalefonts`. With the original `OT1` font encoding, the large fonts are basically scaled versions of the 10pt CM fonts (with a few exceptions). Using the `T1` encoding, the fonts are sized by changing the design point size, not scaling a small font. With the use of the EC fonts in this encoding, the presentation of the foils is *very* different, since scaled versions of small fonts look very different from large point design fonts. The author personally found that these scaled EC fonts looked thin and did not present themselves well. Consequently, declaring this option will replace the large design fonts with scaled versions of the smaller fonts (e.g., the normal font `ecss2074` is replaced by `ecss1000 scaled 2074`). The disadvantage of using this option is that you will need effectively two sets of large EC fonts, the scaled small fonts and the large design fonts. I don't know any other way to deal with this problem. This option has no effect when using the `OT1` encoding (which is the default).

The second additional option, `useDCfonts`, is probably temporary (to be removed at a later date). This option will use the (now unsupported?) DC fonts

version 1.3 instead of the default EC fonts. This is included as a courtesy for DC font users who are not yet ready to move to the official EC fonts. This also works in conjuction with the `magscalefonts` option with the same reasons and caveats as above. This option has no effect *unless* you do something to change the font encoding to `T1`.

You can add either or both of these options to your `foiltex.cfg` file as desired (see above).

3.1.3 New Options in Version 2.1.3

As mentioned in Section 1.3, the following options are new in this version: `dvipsone`, `shortform`. The first provides rotation support for Y&Y, Inc.'s PostScript driver. This can be put in your `foiltex.cfg` file. The second, as described above, provides a method for generating an output document which uses less paper. This is probably *not* suitable for the configuration file. Also, we've added support for more (larger) paper sizes.

3.2 The `\maketitle` command

`\maketitle`

FoilTEX's `\maketitle` command produces results similar to LATEX's `titlepage` class option. That is, it reads the contents of `\title{}`, `\author{}`, etc., and produces a titlepage, actually a title foil. The title itself appears horizontally centered and down a small space from the top, in a `\Large` bold sans serif font. The author's name with address and date appear under the title, centered and in the `\normalsize` font. If desired, this can be followed by a (necessarily short) abstract with the word "Abstract" appearing in bold and centered above the text of the abstract. The footer of the title page will contain some special text (see Section 3.4 for more details). The difference with the `titlepage` class option is that the abstract appears on the same page as the title/author information (provided LATEX didn't force a page break).

3.3 The `\foilhead` and `\rotatefoilhead` macros

`\foilhead`
`\rotatefoilhead`

You start new foils with either the `\foilhead` or `\rotatefoilhead` macros. Their use is described by the following examples:

`\foilhead`{*text*}
`\foilhead[`*length*`]`{*text*}

This starts a new page and puts *text* in `\large` bold type at the top center of the new page. After the header, a vertical space is added providing an automatic cushion between the header and the body of the foil. You can adjust this space either up or down by putting in the optional argument a TEX *length*. For example, if you want the body of your foil to sit closer to the header, you could use the command

```
\foilhead[-.5in]{This is the Header}
```

The default spacing is equal to the sum of

```
\parskip + \baselineskip + \foilheadskip
```

The dimension `\foilheadskip` is new to FoilTEX v2.1.3 and defaults to `18pt plus 0pt minus 18pt` in normal mode and `.25in` in compatibility mode. You can reset this default using a `\setlength` command.

The new (to FoilTEX v2.1.3) macro `\rotatefoilhead` can be used just like the above. If the `dvips` class option has been declared, then it attempts to rotate the *entire* foil 90 degrees from the default position. This means a couple of things. First, if the default position is potrait, then this foil will be rotated to landscape. If the default position is landscape (by use of the `landscape` option), then it rotates to portrait. A similar thing happens if the `dvipsone` option is enabled.

Furthermore, if LATEX decides that it needs to split a rotated foil into more than one page, each of these pages will also be rotated. Normal orientation is recovered by the next invocation of `\foilhead`.

The `lscape` and other package files that can be used to rotate the contents of a page only rotate the page body but leave the header and footer in their normal orientation. This makes sense in documents but not on foils. Consequently, we have to support rotation in a different way. `\rotatefoilhead` will rotate everything on the page.

These macros should be used to start any new foil, especially if a new heading is needed. If you try to put too much text on a single foil, FoilTEX will do its own page break. This could cause some odd vertical spacing since there is a fair amount of stretchability in vertical glue, particularly in list environments. This can easily be fixed simply by forcing a page break with an empty `\foilhead{}` command or a `\newpage` command.

3.4 The `\MyLogo` and `\Restriction` macros

`\MyLogo` `\Restriction`

Another pair of macros not in standard LATEXare `\MyLogo` and `\Restriction`. Each takes a single argument and is used to control the contents of part of the footline. By design, the footline consists of the contents of `\MyLogo` followed by the contents of `\Restriction` all left justified, with the page number right justified[2]. On the main foils, the default font size is `\tiny`. The contents of these macros can be an empty box as well. By default, `\Restriction` is empty and `\MyLogo` is the phrase "– Typeset by FoilTEX –".

The declarations for these macros would normally be placed in the preamble to the document, i.e., before the `\begin{document}` command. However, these macros can be declared or redeclared at any place in the document. They (and all the other commands that control the footer and header) are sensitive to LATEX's output routine. Consequently, care must be taken in their placement to be sure they act on the correct pages. In the preamble or immediately after the `\foilhead` command are best. In addition, there are macro switches that can be used to easily turn on or off the logo, without having to do any redeclarations. See Section 3.4.1 for more information.

`\MyLogo` is really intended for something idiosyncratic to the speaker or his organization. For example, if you use the `graphics` or `graphicx` packages of LATEX 2_ε, you can put an `\includegraphics` macro in `\MyLogo` to get some graphic as the logo on every page:

```
\MyLogo{\includegraphics[height=1in]{arclogo}}
```

[2] For the title foil, there is no page number; `\MyLogo` and `\Restriction` are centered and appear in `\footnotesize` font.

(together with a preamble command `\usepackage{graphicx}`) puts a one inch tall version of the IBM Almaden Research Center logo (in EPS format) in the lower left corner of all the author's foils. `\Restriction` was included in case you want to have each foil identified for a particular audience. For example, at IBM, we have the option of displaying the IBM logo and words like "Confidential". The defaults are set in `foils.cls`.

3.4.1 Toggling the logo

`\LogoOff` `\LogoOn`

Users of an early IBM version of FoilTEX requested an easier mechanism (than undefining/redefining `\MyLogo`) for inhibiting a logo from appearing on selected foils or all foils. We implemented this feature with two switches. These macros are `\LogoOn` and `\LogoOff` and they do exactly what their names imply. If `\LogoOff` appears before the footer is processed by the output routine no logo will appear (as if `\MyLogo{}` were declared). This stays in effect until `\LogoOn` is encountered, at which point the contents of `\MyLogo` are restored.

So, for example, if you do not want the logo to appear at all, you can put the `\LogoOff` command *before* the `\begin{document}` command. If you want the logo only on the title page, then you can put this command *after* the first occurrence of `\foilhead`. You can then turn the logo back on by putting the `\LogoOn` command in a convenient place.

3.5 The other three corners of the page

`\rightfooter` `\leftheader` `\rightheader`

Since the macros `\Restriction` and `\MyLogo` control the bottom left corner of the page, there are other macros for putting text in the other three corners. These are, not surprisingly,

`\rightfooter`{*text*}
`\leftheader`{*text*}
`\rightheader`{*text*}

They each take one argument, the text you want to place in the associated corner of the page. These can also be redeclared within the document with the appropriate attention paid to the output routine. See Section 3.4.

By default the headers are empty and the lower right footer is just the page number:

```
\rightheader{}
\leftheader{}
\rightfooter{\quad\textsf{\thepage}}
```

except on the title page where they are all suppressed. You can easily suppress page numbering by declaring `\rightfooter{}`. Unless controlled by a font size changing command, text in these regions appear in a `\tiny` font. These defaults are set in `foils.cls`

3.6 Header and footer rules and `fancyheadings.sty`

Many users requested a simple utility for putting rulers in the header or footer. As mentioned above in Section 3.1, this is now easily obtained with the `headrule` and `footrule` class options.

\headwidth

Some users prefer to use the `fancyheadings.sty` package to control their headers and footers. This allows control of rules, as well as centered text in these areas (which we thought was too much clutter). To get `fancyheadings.sty` to work correctly with FoilTEX's page rotation mechanism, add the following to your preamble:

```
\let\headwidth\textwidth
```

3.7 Predefined `Theorem` and `Proof` environments

\newtheorem
Theorem
Lemma
Corollary
Proposition
Definition
Proof

There are a number of (both starred and unstarred) `\newtheorem` environments built in. These are for `Theorem`, `Lemma`, `Corollary`, `Proposition` and `Definition`. Note the uppercased first letter (to avoid possible collisions with user-defined environments of this type). Each must begin and end with `\begin{}` and `\end{}` commands as usual. Their text begins with a bold sans serif label like **Theorem** and the content of each is typeset in *slanted sans serif*. The unstarred forms are sequentially numbered and support automatic referencing. The starred forms suppress the numbering and referencing.

All these environments also support an optional argument that can be used for the inventor, common name of the theorem, etc.. Thus

```
\begin{Theorem*}[Gauss] Quadratic reciprocity is true!
\end{Theorem*}
```

will produce (in large type)

Theorem. [Gauss] *Quadratic reciprocity is true!*

The unstarred form will be numbered.

To implement this, we added code to LaTeX's `\newtheorem` macro which defines *both* the starred and unstarred forms of these environments at the same time. In this way, users could easily add their own versions of similar environments. For example,

```
\newtheorem{Axiom}{Axiom}
```

would define two environments `Axiom` and `Axiom*` that behaved just like `Theorem` and `Theorem*`. In all other respects, e.g., numbering convention, `\newtheorem` behaves just as in LaTeX.

Finally, there is a `Proof` environment which opens with the word **Proof** and ends with a □. The contents are printed in the normal font.

3.8 Mathematics in bold typeface

\bm

Because FoilTEX is fully integrated into LaTeX, getting bold math is pretty much the same. The only thing we did was to add some extra facilities to get at it. The first we add is the `\bm` macro which takes one argument:

`\bm{`*formula*`}`

This takes it argument (within mathematics mode) and replaces it with it emboldened version. This is different from the `\mathbf` command in that it emboldens everything including symbols. The new version of this (as opposed to FoilTEX v 1.0.1) is much better. It deals correctly with the current math style and so should work even in superscripts!

`boldequation` The second method for getting bold mathematics is a pair of environments:

```
\begin{boldequation}
formula
\end{boldequation}

\begin{boldequation*}
formula
\end{boldequation*}
```

They both set *formula* in bold (even super- and subscripts). The unstarred form has automatic referencing and is numbered; the starred form inhibits the numbering and referencing. Essentially, they just do an automatic `\boldmath` at the beginning and an `\unboldmath` at the end of the formula.

3.9 Hyphenation and raggedright

`\righthyphenmin` `\lefthyphenmin` `\raggedright` FoilTEX turns off hyphenation but allows a fair amount of horizontal interword spacing. Unfortunately, this can create some unpleasant line breaks (at times). This can be fixed with either a manual fill and line break (I use `\hfil\break` but that's very TEXy) or by inserting a discretionary hyphen (better to have them only when needed than to have them happen unpredictably).

Some users feel that `\raggedright` is preferable for foils. It was decided not to make this the default (as this is not the author's opinion), but to leave this to the user's discretion. To get this effect, simply put `\raggedright` in the preamble to your document. This will also reduce the spacing problems.

We turned off hyphenation by setting `\righthyphenmin` and `\lefthyphenmin` each to 100. Resetting this to smaller numbers (less than the length of typical words) will restart hyphenation. (This is an improvement over version 1.0.1, where we didn't even load the hyphenation tables into the format.)

3.10 Non-floating floats

`figure` `table` Many people like to use figures and tables in their foils and typically want to use them with the same interface as they do in documents (namely, using the `figure` and `table` environments). Unfortunately, in standard LATEX these are "floats" which means they can appear on some place other than their current spot. This, of course, makes no sense on foils. To alleviate this, we define these environments but as non-floats. The user syntax is identical to standard LATEX, but a couple of things happen. First, the placement parameters are completely ignored. Second, they don't float (equivalent to `[H]` placement). Third, there is no "List of Figures" or "List of Tables" data generated. The only other difference is that the starred forms of these environments are unnumbered (as opposed to being one-column in a two-column document).

`\newnonfloat` We have added a simple mechanism for the user to add their own version of

these non-floats. This is done with the `\newnonfloat` mechanism. For example, the `figure` and `figure*` (unnumbered) environments of FoilTEX are defined with:

```
\newnonfloat{figure}{Figure}
```

3.11 User adjustable dimensions

`\foilheadskip`
`\abovefloatskip`
`\captionfraction`
`\abovetitleskip`
`\titleauthorskip`
`\authorauthorskip`
`\authordateskip`
`\dateabstractskip`

There are a number of new dimensions added to FoilTEX which the user can adjust as they desire. One, `\foilheadskip` has already been mentioned.

For controlling the space around floats there are a number of dimensions. First, there are the two predefined in LATEX, namely, `\abovecaptionskip` and `\belowcaptionskip`. Next, we added one (`\abovefloatskip`) for the space above the non-float. See the code for default settings for these skips. Finally, in version 2.1.2 we added a `\captionfraction` command to set the width of the captions as a fraction of the text width. Implementation is similar to LATEX's fractions for floats, so that it should be reset with the `\renewcommand`. The default value is 1.0. Releases of FoilTEX before version 2.1.2 allowed the user to set the caption width with the dimension `\captionwidth`. This was buggy and behaved badly under rotations and landscape and so has been deprecated.

For the titlepage, between every single vertical item there is some extra space. These spaces can take up a lot of room and if you have a lot of authors or a long title or a long abstract and can't get everything you want on the page, adjust these parameters as needed:

```
\abovetitleskip, \titleauthorskip, \authorauthorskip,
\authordateskip, \dateabstractskip
```

`\zerolistvertdimens`

Furthermore, there is a new declaration called `\zerolistvertdimens`. You can use this *inside* a list environment to shrink all the vertical spacing to a minimum.

3.12 Differences with LATEX

`\em`
`\emph`

One simple difference is that the LATEX commands `\em` and `\emph` switch from any unslanted font to *slanted sans serif* and from any slanted font to unslanted sans serif, not to *text italics* and roman, respectively.

Unlike TEX/LATEX, numerals in FoilTEX look different when they are in ordinary text from when they are in math-mode. This means that 12345 in text will print as 12345 and `$12345$` prints as 12345.

3.13 Future versions

There are plans for two major additions to FoilTEX. These are support for user comments (side notes) and for overlays. There is no time-table for these upgrades. (As one can probably tell, this many take a really long time to do.)

In the meantime, you can play with the `commment` package for side notes and the color package (using `white` to hide text) for overlays. For side notes, you can use the comment environment's suppressed mode for the real foils. Then use the `shortform` option and enabled comments to generate the side note foils. The foils will layout very differently, but the content will be there.

FoilTEX should be able to handle rotation and landscape with other dvi drivers. This will be added only with user assistence because the author has no way to test other drivers.

4 Fonts and their sizes

As noted earlier, the default font at `\normalsize` is a sans serif font at size 20pt, unless one of the `[17pt]`, `[25pt]`, `[30pt]` or `shortform` options have been declared in the `\documentclass` command. Table 2 shows the control sequences for other accessible text fonts and the name of the font in a sample of its type. These control sequences give the font at the current size. Font size changing commands for each of the normal point size options are described by Table 3. Note that `\bf` and `\sl` yield sans serif fonts, not the usual variations on roman.

Table 2: Available fonts and their names.

command	font names
`\sf`	Sans Serif
`\it`	*Text Italic*
`\sl`	*Slanted Sans Serif*
`\bf`	**Bold Sans Serif**
`\tt`	`Typewriter`
`\rm`	Roman
`\sc`	Small Caps

Table 3: Type sizes for FoilTEX size-changing commands for the different document style options. In compatibility mode, the largest size is 43pt.

	doc-options				
size	20pt	17pt	25pt	30pt	shortform
`\tiny`	12pt	12pt	12pt	14pt	12pt
`\scriptsize`	12pt	12pt	14pt	17pt	12pt
`\footnotesize`	14pt	12pt	17pt	20pt	12pt
`\small`	17pt	14pt	20pt	25pt	12pt
`\normalsize`	20pt	17pt	25pt	30pt	12pt
`\large`	25pt	20pt	30pt	36pt	14pt
`\Large`	30pt	25pt	36pt	43pt	17pt
`\LARGE`	36pt	30pt	43pt	51pt	20pt
`\huge`	43pt	36pt	51pt	51pt	25pt
`\Huge`	51pt	43pt	51pt	51pt	25pt

There are also the corresponding `\textsf` and related macros. These will work as expected, namely they switch the corresponding font characteristic based on the *current* font.

Table 4: Mathematics type styles and their point sizes at `\normalsize` for the different document style options.

	doc-options				
style	20pt	17pt	25pt	30pt	shortform
displaystyle,textstyle	20pt	17pt	25pt	30pt	12pt
scriptstyle	14pt	12pt	17pt	20pt	12pt
scriptscriptstyle	12pt	12pt	14pt	17pt	12pt

Mathematics is also automatically displayed at normal size unless magnified by a size changing declaration. Table 4 describes the font point sizes for TEX's mathematics styles at each of the normal point size options.

In the previous version of FoilTEX, we loaded the LATEX circle and line fonts at `\magstep4`. This had the advantage that the lines where thicker and some pictures would scale nicely to foils. Unfortunately, this seems to have created as many problems as it solved. Consequently, in the new version, these fonts are not scaled except in compatibility mode.

5 Making color foils

Because color has now been integrated into LATEX itself, we don't add much of anything for color support. Users are encouraged to use the `color` package of the `graphics` package. We no longer recommend the `colordvi` style file that came with the old FoilTEX and still comes as part of Rokicki's `dvips`. In general it will work OK, but there will be some minor problems that crop up.

We have added just enough compatibility mode to the new FoilTEX so that old files which use `colordvi` should still run correctly.

6 Using PostScript fonts

There is not much to say here since the PSNFSS packages do a nice job of things. The only thing to remember is that the default font family is the `\sfdefault` so that using a package that only changes the `\rmdefault` won't help you much. Using a package like `times.sty` will change the `\sfdefault` to Helvetica and *not* to Times-Roman as you might expect. To get your foils (mostly) into Times-Roman, just say

```
\renewcommand{\sfdefault}{ptm}
```

somewhere in the preamble.

7 Installing FoilTEX

To install FoilTEX, simply run LATEX on the `foiltex.ins` file and then follow the instructions at the end. Essentially, just copy the generated files to the appropriate location in your TEX inputs, examples and doc trees.

8 Usage restrictions

8.1 Experimental Software Disclaimer

This program is provided free of charge on an "AS IS" basis without warranty of any kind, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose. IBM does not warrant that the functions contained in this program will meet the user's requirements or that the operation of this program will be uninterrupted or error-free. You are solely responsible for determining the appropriateness of using this program and assume all risks associated with its use, including but not limited to the risks of program errors, damage to or the loss of data, programs or equipment, and unavailability or interruption of operations.

Acceptance and use of this program constitutes the user's understanding that he/she will have no recourse to IBM for any actual or consequential damages, including, but not limited to, lost profits or savings, arising out of the use or inability to use this program, or any damages claimed by you based on a third party claim. Even if the user informs IBM of the possibility of such damages, IBM expects the user of this program to accept the risk of any harm arising out of the use of this program, or the user shall not attempt to use this program for any purpose.

8.2 User Agreement

BY ACCEPTANCE AND USE OF THIS PROGRAM THE USER AGREES TO THE FOLLOWING:

a. The user is granted permission to copy this program to the extent reasonably required for such use. This program may be redistributed on a not-for-profit basis, as long as WARRANTY DISCLAIMER, this USER AGREEMENT, and the copyright notice are included with the program.

b. All title, ownership and rights to this program and any copies remain with IBM, irrespective of the ownership of the media on which the program resides.

c. The user is permitted to create derivative works to this program. However, all copies of the program and its derivative works must contain the IBM copyright notice, the WARRANTY DISCLAIMER and this USER AGREEMENT. Furthermore, the user must document and initial within the program all changes he/she makes.

d. By furnishing this program to the user, IBM does *not* grant either directly or by implication, estoppel, or otherwise any license under any patents, patent applications, trademarks, copyrights or other rights belonging to IBM or to any third party, except as expressly provided herein.

e. The user understands and agrees that this program, and any derivative works made from this program, are not to be sold for profit or commercially exploited in any manner. However, this use restriction shall not operate to deny the right to redistribute the program on a not-for-profit basis, as provided in paragraph a, above.

f. IBM requests that the user supply to IBM a copy of any changes, enhancements, or derivative works which the user may create. The user grants to IBM and its subsidiaries an irrevocable, nonexclusive, worldwide and royalty-free license to use, execute, reproduce, display, perform, prepare derivative works based upon, and distribute, (INTERNALLY AND EXTERNALLY) copies of any and all such materials and derivative works thereof, and to sublicense others to do any, some or all of the foregoing, (including supporting documentation).

9 Acknowledgements, requests and help

For the first release of FoilTEX, we had the following acknowledgement:

> We would like to thank and acknowledge the following people in IBM for their great assistance in helping to put FoilTEX together: Katherine Hitchcock, Myron Flickner, Ekkehard Blanz, Melanie Fulgham, Peter Haas, Rocky Bernstein and the many users who contributed their constructive comments on the early test versions within IBM.
>
> A special thanks goes to Tomas Rokicki for implementing our color setup in his driver and another to Sheri Gish of IBM for asking the right (or was it wrong?) question that got this project started.

We echo that acknowledgement again because there would not be a version 2.0 or later without the first release and their help. But, putting together this new release required a new team of helpers. Mostly, I want to thank the LaTeX3 team for doing such a great job with LaTeX 2_ε. Implementing much of FoilTEX was greatly eased by the way LaTeX 2_ε is constructed and implemented.

Beyond this, the LaTeX 2_ε team was very helpful in a number of discussions about design issues for FoilTEX, particularly on font issues but also on a number of others. Let me single out the following for exceptional service beyond the call: David Carlisle, Sebastian Rahtz, Frank Mittelbach and Alan Jeffrey.

I also want to thank all the many users who have expressed their interest in FoilTEX and who have made suggestions, comments, pointed out the bugs and provided fixes.

A special thanks to the folks at Y&Y, Inc., including Louis Vosloo, for providing the stuff needed for landscaping with DVIPSONE.

FoilTEX is intended to be easy to use, useful and to produce beautiful foils. Consequently, the author welcomes any comments or suggestions.

The author also requests help in supporting landscape and rotation with other drivers. If you find a solution, please pass it back to the author for incorporation in future updates of FoilTEX.

If you have a question that you can't answer by reading *both* this document and [3, 1], or by posting your question to your local TEXperts or to the usual forums, you can contact the author.

10 A sample file

In this section we give a complete and self-annotated sample file. We try in this sample to demonstrate or comment on most of the basic features and show how

the fonts look in FoilTEX. More details can be found in the sections above.

```
⟨*sample⟩
%%%% Everything above this sentence is relevant only to this    %%%%%
%%%% sample and should not be included in your foils document.  %%%%%
% Start with a \documentclass declaration.  The [20pt] is optional
% as that is the default.
\documentclass[20pt,dvips]{foils}
%
% An alternate uses smaller fonts, passes the draft mode to other
% included packages (e.g., graphics packages) and does everything
% in landscape mode.
% \documentclass[17pt,draft,landscape]{foils}
%
% This next version declares the setup for Rokicki's dvips so that we
% can support rotated foils and the use of color names.
%\documentclass[20pt,dvips]{foils}
%
% Add other packages you might want to use here.  E.g., you can
% use the \usepackage{graphics} or \usepackage{graphicx} to include
% figures (e.g., postscript figures) or \usepackage{color} for color.
% See the \FoilTeX documentation for use of color.
% \usepackage{graphicx}
% \usepackage[usenames]{color}
% Use this package file for the special \LaTeX symbols.
\usepackage{latexsym}
%
% Declare the title, author and date as you would in regular \LaTeX.
%
\title{A Sample Foils Document}
%
\author{Jim Hafner\\
        IBM Research Division\\
        Almaden Research Center\\
       \texttt{hafner@almaden.ibm.com}
       }
% This is optional
\date{\today}
%
% We set up the header and footer information now.  \Restriction is
% always placed next to \MyLogo on the bottom left corner of the
% page.  The other macros have their obvious placement.  If you want
% nothing, leave the contents of these declarations empty.  Note that
% ``-- Typeset by FoilTeX --'' logo will print automatically, unless
% controlled by redefining \MyLogo or by using the switches \LogoOn
% and \LogoOff.
%
%\MyLogo{-- Typeset by \FoilTeX\ --}   % this is the default
%\rightfooter{quad\textsf{\thepage}}   % this is the default
\leftheader{Jim Hafner}
\rightheader{\filedate}                % \filedate is defined above
\Restriction{Is this for a restricted audience?}
%
\newcommand\bs{\char '134}   %  a backslash character for the \tt font
%
```

```
% Now we can begin the document.  The first thing is the title page
% on which we might put a VERY short abstract.
\begin{document}
\maketitle
%
\begin{abstract}
This is where an abstract might go if you want one.  There is usually
not a lot of room for much here.
\end{abstract}
%
% Now we can begin to start the individual foils.  Note that we let
% \LaTeX split the pages and don't force each foil to be a separate
% page.   We use \foilhead to start a new foil or foil topic.
\foilhead{Colors}

\FoilTeX\ (version 2) is fully integrated with \LaTeXe\ so that the
supported \texttt{color} package (part of the \texttt{graphics}
package) is the preferred way to use colors.

For examples, see the \texttt{graphics} package documentation.

You can still use the old \texttt{colordvi} package that comes with
Rokicki's \texttt{dvips}, but some things won't work exactly as
expected (except in compatibility mode).

\foilhead[-.5in]{Itemize}
\LogoOff  % We decided to turn the logo off on this page.
          % The rest of the footer and header will stay the same.
% This foil has a lot on it so we used the [-.5in] option to move the
% body of the foil closer to the foilhead.  We could move it down with
% a positive length.
%
% By default there is large item separation at the first level (equal
% to \parsep).  At the lower levels, it decreases to 0.  To change
% you can either try to redefine \@listii and \@listiii in a style
% file or you can change it internally to an item list.  See comment
% below at second level.

\begin{itemize}
\item
This is the first level of an itemize.
  \begin{itemize}     % \itemsep 20pt % this stretches the space
                      % between items but only at this level. We leave
                      % the default here.
  \item Here we jump to second level
     \begin{itemize}
     \item Even third level (and there is a fourth level as well).
     \item The second item at third level.
     \end{itemize}
  \item The second item at second level.
  \item A third item at second level.
  \end{itemize}
\item The second item at first level.
\item A third item at first level.
```

```
\end{itemize}

Note that we have turned off the logo on this page.  It returns on the
next page.

\foilhead{}
\LogoOn

This should be a \emph{new foil} with no header, followed by a quote:
\begin{quote}
\ldots it's a good idea to make your input file as easy to
read as possible.
\end{quote}
and some enumerating:
\begin{enumerate}
\item this is enumerated
\item this is also enumerated
\end{enumerate}

\begin{center}
\framebox{\parbox{5.5in}{Above, we used \texttt{\bs emph\{new foil\}}
instead of the old \texttt{\{\bs em new foil\bs/\}}!}}
\end{center}

On the following page we decided to stop the headers from appearing
and move the date to the footline.

 % Here are more samples of font actions, particularly size changing.
 % We retained the ability to get a true Roman font via \textrm.

\foilhead{Fonts}
\rightheader{}   % These cancel the headlines from now on
\leftheader{}    %
\rightfooter{\filedate\quad\textsf{\thepage}}

\begin{itemize}
\item This shows \textit{italics}, \textsl{slanted}, \textbf{boldface},
\texttt{typewriter}, \textrm{roman}, and \textsc{small caps}.
\item Unslanted \emph{emphasize} and
      \textsl{slanted \emph{emphasize}}.
\item \texttt{\bs textrm} means \textrm{roman and
      \texttt{\bs textsf} means \textsf{sans serif}}.
\item size changing from {\tiny tiny},
      {\scriptsize scriptsize},
      {\footnotesize footnotesize},
      {\small small}, {\normalsize
      normalsize},\,{\large large},%
      {\Large Large},\,{\LARGE LARGE},
      {\huge huge}, and {\Huge Huge}.
\item 12pt is the smallest preloaded.
\item 43pt is the largest preloaded in compatibility mode.
\item 51pt is the largest preloaded in normal mode.
\end{itemize}
```

```
\foilhead{Special Characters and Accents}

\begin{itemize}
\item Here is a list of accents:
  \begin{itemize}
   \item \‘{o}, \’{o}, \^{o}, \"{o}, \~{o}, \={o}, \.{o}, \u{o},
         \v{o}, \H{o}, \t{oo}, \c{o}, \d{o}, \b{o}.
  \end{itemize}
\item First in paragraph mode (with \verb|\copyright|):
  \begin{itemize}
  \item \dag, \ddag, \S, \P, \pounds, 0123456789, \copyright
  \end{itemize}
\item Then in math mode: (numerals are different!)
  \begin{itemize}
  \item $\dag$, $\ddag$, $\S$, $\P$, $\pounds$,
        $0123456789$
  \end{itemize}
\item Here are more non-english language symbols:
  \begin{itemize}
  \item \oe, \OE, \ae, \AE, \aa,   \AA, \o, \O, \l, \L, \ss, ?‘, !‘
  \end{itemize}
\item \textrm{\TeX}’s special symbols: \#, \$, \%, \&, \_, \{, \}.
\end{itemize}

\foilhead{Some Mathematics}

$$
{\cal F}\cdots\frac{x+y}{1+\frac{y}{z+1}} =\sqrt{x+y}\times \sqrt[n]{2}
$$

Here are some funny math symbols (we needed the \texttt{latexsym}
package for a couple of these):
$$
\forall\exists\flat\natural\sharp\partial\angle\Re\mho\jmath\aleph
$$
$$
\infty\Diamond\clubsuit\diamondsuit\spadesuit\heartsuit\ell
$$
$$
\bigcap\bigcup\bigvee\bigwedge\bigodot\bigotimes\bigoplus\biguplus
\sum\prod\coprod\int\oint
$$
$$
\sum_{i=1}^nx_i =\int_0^1 f[x]dx.
$$
and $\gcd(m,n)$ and $x\equiv y\pmod{a+b}$.

\foilhead{More Math: arrays}

$$
\begin{array}{clcr}
a+b+c & uv    & x-y & 27 \\
a+b   & u+v   & z   & 134  \\
a     & 3u+vw & xyz & 2,978
```

```
\end{array}
$$

$$
\left( \begin{array}{c}
    \left| \begin{array}{cc}
             x_{11} & x_{12} \\
             x_{21} & x_{22}
           \end{array}
     \right| \\
      y \\
      z
      \end{array}
\right)
$$

$$
x=\left\{ \begin{array}{ll}
              y & \mbox{if $y>0$} \\
              z+y & \mbox{otherwise}
          \end{array}
   \right.
$$

\foilhead{More Math: \texttt{equation} and \texttt{array} Numbering}

Here is a numbered equation
\begin{equation}
  E=mc^2
\end{equation}
and a numbered array
\begin{eqnarray}
 x & = & 17y \\
 y & > & a+b+c+d+e+f+\nonumber \\
   &   & k+l+m+n+o+p
\end{eqnarray}

More math accents: $\hat{a}$, $\check{a}$, $\breve{a}$, $\acute{a}$,
$\grave{a}$, $\tilde{a}$, $\bar{a}$, $\vec{a}$, $\dot{a}$, $\ddot{a}$.

Over and underline:
$$
  \overline{\overline{x}^{2} + 1}
\qquad
\overbrace{a+\underbrace{b+c}_{25}+d}^{16\alpha}
$$

\foilhead{Bold Mathematics}

The \verb|boldequation| environment, with
numbering yields
\begin{boldequation} \label{eq:bold}
   2\sqrt{x} \Pi^y \sim\pi\times x
\end{boldequation}
```

```
and without numbering yields
\begin{boldequation*}
   2\sqrt{x} \Pi^y \sim\pi\times x
\end{boldequation*}
We can reference bold equations like (\ref{eq:bold}).

There is also \verb|\mathbf| and \verb|\bm| in the middle of a formula,
with \hfil\break \verb|\mathbf{a +} a + \bm{a+x\pi-\rho} -\rho|
$$
\mathbf{a +}a+\bm{a+x\pi-\rho} -\rho
$$
Note the difference between the two bold ``a'' in result.

\foilhead{Theorem and Proof Environments}

\begin{Theorem}[TUG'92]
There are some \TeX\ tools that are easier to use than others.  This
theorem is numbered and has an optional acknowledgement.
\end{Theorem}

\begin{Corollary*}
This obvious corollary is not numbered because it uses the *-form.
\end{Corollary*}

\begin{Proof}
The details of the proof are left to the reader.  Note that the
environment names are case sensitive.
\end{Proof}

We put a header back on the next foil to see that it is correctly
rotated.

\rotatefoilhead{Rotated Foils}
\leftheader{I'm rotated!}

This entire page will (should?)~be rotated if we declared the
\verb|\documentclass| option \texttt{dvips}.   In the other cases,
this is not supported and the user is warned.

The next foil will return to normal.  Only foils that begin with
\verb|\rotatefoilhead| will be rotated.  If \LaTeX\ needs to split a
rotated foil into two foils, \emph{both} will be rotated.

\foilhead{Tables and Figures}
\leftheader{}

Here is a short table:

\begin{table}
\begin{center}
\begin{tabular}{|l|c|r|} \hline
First stuff & Second stuff & Third Stuff \\ \hline\hline
foo & bar & bug \\ \hline
foofoo & barbar & bugbug \\ \hline
```

```
\end{tabular}
\caption{\label{tab:one}This is the first table.}
\end{center}
\end{table}

The above table is Table~\ref{tab:one}.  It is a ``nonfloat'', since
it doesn't float at all, but appears right where it was placed in the
document.

 % In the picture environment, we have put the fonts back to \LaTeX's
 % normal size.  This may have been a mistake but ...
\foilhead{\texttt{picture} Environments}

\begin{figure}
\begin{center}
\setlength{\unitlength}{.08in}
\begin{picture}(50,50)(0,0)
 %\thicklines         % don't know why it looks worse with this
\put(10,35){\framebox(10,10){gnat}}
\put(30,20){\circle{20}}
\put(30,20){\vector(0,1){2}}
\put(45,35){\circle*{5}}
\put(30,20){\oval(20,30)}
\end{picture}
\end{center}
\caption{Isn't this a pretty picture?}
\end{figure}

\verb|\thicklines| would be nice here but the picture doesn't look as
good that way (why?).  We should use the \texttt{graphics} package to
load our graphics pictures here.  Note that this Figure doesn't float!

\foilhead{Marginal Notes, Footnotes and Citations}

A marginal note\footnote{This is a footnote.}
is made with the \verb|\marginpar| command, having the
text as its argument.  The note\marginpar{This is a mar\-gin\-al note}
is placed in the margin, its first line even with the line of the text
containing the command\footnote{This is a second footnote.}.

This is shows \fbox{frameboxes} at work. We can even cite references
like \cite{rocky} and \cite{bullwinkle}.

 % We can use any bibliography mechanism, including BibTeX.
\begin{thebibliography}{99}

\bibitem{rocky} Rocky and Bullwinkle, Open problems, in \textsl{Mr.
Know-it-all's Rock Encyclopedia}.

\bibitem{bullwinkle} Bullwinkle, Getting things out of hats,
\textsl{Annals} \textbf{1} (1990BC), 1--2.

\end{thebibliography}
```

```
\foilhead{Other Features}
\raggedright

\begin{itemize}
\item \verb|\raggedright| can be used in the preamble to get this
      effect throughout as we did on this page.
\item Access to the AMS Fonts symbols and Euler fonts can easily be
      obtained with the \texttt{amslatex} (or is it \texttt{amsmath})
      packages.  You might not get fonts at the largest sizes however.
\item \textsc{PostScript} fonts can used just by adding any of the
      package files from the PSNFSS package.
\end{itemize}

\end{document}
⟨/sample⟩
```

11 Description of the code in the FoilTEX package

In this section we describe the macros used in the package. This is broken up into a number of parts. In the first part we describe the code that sets up and implements the option parsing. This includes the loading of the font definition information and the size option file. Details of the font data itself can be found in Section 14. In the second subsection, we define the basic macros used in FoilTEX. Subsections 11.3-11.4 have the code for list spacing and list structures. Page layout is described in Subsections 11.5-11.7. The next subsections cover bibliographies (11.8), non-floating floats (11.9), foil heads (11.10), new theorem environments (11.11), and bold mathematics (11.12).

11.1 The package options

`\if@openbib` `\if@landscape` `\if@dvips` `\if@dvipsone` `\if@header@rule` `\if@footer@rule`

To setup some of the mechanisms for parsing the package options, we need a few conditionals. The first and second conditionals are related to standard LATEX document class options. The `\if@dvips` conditional, if true, sets up the necessary `\special`s for page rotation and the like. These are only supported for Rokicki's `dvips` and provided for Y&Y, Inc's, `dvipsone`. The two rule conditionals determine if a rule is set in the header and footer, respectively. By default, all of these conditionals are false. The next two deal with how `T1` encoded fonts are used.

```
⟨*package⟩
\newif\if@openbib \@openbibfalse
\newif\if@landscape \@landscapefalse
\newif\if@dvips \@dvipsfalse
\newif\if@dvipsone \@dvipsonefalse
\newif\if@header@rule \@header@rulefalse
\newif\if@footer@rule \@footer@rulefalse
\newif\if@magscaleECfonts \@magscaleECfontsfalse
\newif\if@useDCfonts \@useDCfontsfalse
```

`a4paper` `letterpaper` `35mmSlide` `landscape` `ledgerpaper` `legalpaper` `a3paper` `a2paper` `a1paper`

These document class options are used to set the paper layout. Three of these are standard LATEX, but the fourth (`35mmSlide`) is new to FoilTEX. This sets the paper dimesions in the aspect ratio for 35mm film, namely, 2×3 and just big enough to fit on US letter paper (I'd have picked A4 paper, but I'm US centric!).

```
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{ledgerpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {7.33in}}
\DeclareOption{legalpaper}
   {\setlength\paperheight {14in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{a3paper}
   {\setlength\paperheight {420mm}%
    \setlength\paperwidth  {297mm}}
\DeclareOption{a2paper}
   {\setlength\paperheight {594mm}%
    \setlength\paperwidth  {420mm}}
\DeclareOption{a1paper}
   {\setlength\paperheight {840mm}%
    \setlength\paperwidth  {594mm}}
\DeclareOption{35mmSlide}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {7.33in}}
\DeclareOption{landscape}
   {\setlength\@tempdima   {\paperheight}%
    \setlength\paperheight {\paperwidth}%
    \setlength\paperwidth  {\@tempdima}%
    \@landscapetrue}
```

`\@ptsize` As in the standard LaTeX classes, this macro holds the point size of the normal font. The available options here are `17pt`, `20pt`, `25pt`, `30pt` with `20pt` being the default. There is also a special option `shortform` which compresses the full large pages into a shorter form. In this form, text is smaller (generally 12pt), new foils do not cause a page break, only a vertical skip. This is useful for making either hard or soft copy for redistribution (to save paper).

```
\newcommand\@ptsize{}
\DeclareOption{shortform}{\renewcommand\@ptsize{shrt}}
\DeclareOption{17pt}{\renewcommand\@ptsize{17}}
\DeclareOption{20pt}{\renewcommand\@ptsize{20}}
\DeclareOption{25pt}{\renewcommand\@ptsize{25}}
\DeclareOption{30pt}{\renewcommand\@ptsize{30}}
```

The next few options are standard LaTeX: `leqno`, `fleqn`, `draft`, `final`, `openbib`.

```
\DeclareOption{leqno}{\input{leqno.clo}}
\DeclareOption{fleqn}{\input{fleqn.clo}}
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\DeclareOption{final}{\setlength\overfullrule{0pt}}
\DeclareOption{openbib}{\@openbibtrue}
```

`headrule` `footrule` The "rule" options enable rules in the header and footer.

```
\DeclareOption{headrule}{\@header@ruletrue}
\DeclareOption{footrule}{\@footer@ruletrue}
```

`dvips|dvipsone` The `dvips` or `dvipsone` option are *required* for the rotation of individual foils. We also use this together with the `landscape` option to issue the `\special{landscape}` required to rotate the whole document. This just saves the user from doing it.

```
\DeclareOption{dvips}{\@dvipstrue}
\DeclareOption{dvipsone}{\@dvipsonetrue}
```

`magscalefonts` `useDCfonts` These two options control the conditionals which trigger changes in the use of the `T1` encoding. See Section 3.1.2 for more details.

```
\DeclareOption{magscalefonts}{\@magscaleECfontstrue}
\DeclareOption{useDCfonts}{\@useDCfontstrue}
```

Because a number of standard LaTeX options don't make sense in FoilTeX, we have disabled them. These options are `a5paper`, `b5paper`,`executivepaper`, `10pt`, `11pt`, `12pt`, `oneside`, `twoside`, `openright`, `openany`, `titlepage`, `notitlepage`, `onecolumn`, `twocolumn`. With the exceptions of `oneside`, `titlepage`, `onecolumn`, they all return a warning to the user. In the case of point size options, suggestions are given to the user to try the official FoilTeX size options. (Note, in versions prior to 2.1.3, `legalpaper` was not available.)

```
\DeclareOption{a5paper}{%
    \ClassWarning{FoilTeX}{No 'a5paper' option for foils.}}
\DeclareOption{b5paper}{%
    \ClassWarning{FoilTeX}{No 'b5paper' option for foils.}}
\DeclareOption{executivepaper}{%
    \ClassWarning{FoilTeX}{No 'executivepaper' option for foils.}}
\DeclareOption{10pt}{%
    \ClassWarning{FoilTeX}{No '10pt' foils option, try shortform,
    17pt, 20pt, 25pt or 30pt (defaulting to 20pt).}}
\DeclareOption{11pt}{%
    \ClassWarning{FoilTeX}{No '11pt' foils option, try shortform,
    17pt, 20pt, 25pt or 30pt (defaulting to 20pt).}}
\DeclareOption{12pt}{%
    \ClassWarning{FoilTeX}{No '12pt' foils option, try shortform,
    17pt, 20pt, 25pt or 30pt (defaulting to 20pt).}}
\DeclareOption{oneside}{}
\DeclareOption{twoside}{%
    \ClassWarning{FoilTeX}{No 'twoside' option for foils.}}
\DeclareOption{openright}{%
    \ClassWarning{FoilTeX}{No 'openright' option for foils.}}
\DeclareOption{openany}{%
    \ClassWarning{FoilTeX}{No 'openany' option for foils.}}
\DeclareOption{titlepage}{}
\DeclareOption{notitlepage}{%
    \ClassWarning{FoilTeX}{No 'notitlepage' option for foils.}}
\DeclareOption{onecolumn}{}
\DeclareOption{twocolumn}{%
    \ClassWarning{FoilTeX}{No 'twocolumn' layout for foils.}}
```

Here we execute the default options and do the processing of the document class option list and what is specified in the `foiltex.cfg` file if one exists. We add to this the issuing to the dvi file of the `\special{landscape}` if the user has the `landscape` *and* `dvips` or `dvipsone` options declared.

```
\ExecuteOptions{letterpaper,20pt,final}
\InputIfFileExists{foiltex.cfg}{}{}
\ProcessOptions
\if@landscape
  \if@dvips{\special{landscape}}\else
  \if@dvipsone{\special{landscape}}\fi\fi
\fi
```

Finally we are ready to load the other FoilTEX external files required. The first loads all the font definition information, including font shape data usually stored in `.fd` files. We put them together for simplicity and to reduce the number of files loaded. The second is the file used for the size option.

```
\input{fltfonts.def}
\input{foil\@ptsize.clo}
```

11.2 Some basic macros

`\FoilTeX` Of course, we have to define the FoilTEX logo. As is done for the LATEX logos, We use the new `\DeclareRobustCommand`. Note that, in contrast with the other logos, we force the FoilTEX logo to always appear in the same font family, shape and series with only the size varying.

```
\DeclareRobustCommand\FoilTeX{{\normalfont%
   {\sffamily Foil}\kern-.03em{\rmfamily\TeX}}}
```

`underscore` The underscore in LATEX is too thin, in fact it has fixed thickness independent of the font size. This makes some sense if the font sizes aren't allowed to vary that much. We fix it here so that it varies with font size always.

```
\renewcommand\_{\leavevmode\kern.06em\vbox{\hrule width.4em height.12ex}}
```

`\footnoterule` `\@makefntext` `\@makefnmark` The default LATEX footnote rule is to short (2in) on foils so we make it longer. Note that the thickness of the rule is still the same (.4pt). The macro for making the footnote text is not part of the default format but is defined in class files like `article.cls`. We steal that definition and put it here. Curiously, the macro for making the footnote mark *is* defined in the format (`latex.ltx`), but we fix it here so that it gets sans serif numerals.

```
\renewcommand\footnoterule{\kern-3\p@\hrule width.4\textwidth\kern2.6\p@}
\newcommand\@makefntext[1]{\parindent 1em\noindent
 \hbox to 1.8em{\hss\@makefnmark}#1}
\def\@makefnmark{\hbox{$^{\mathsf{\@thefnmark}}\m@th$}}
```

`\marginpar` `\@oldmarginpar` `\@foilmarginpar` `\@foilmarginparR` `\@foilmarginparRL` The default definition of `\marginpar` doesn't work well in FoilTEX because of the lack of hyphenation, the large size of the fonts and the very narrow space for these items. So, we fix it a little by forcing it to be `\raggedright` (for right margin paragraphs) and `\raggedleft` (for left margin paragraphs). We cheat here so that we don't need to worry about what is happening in LATEX's innards in this context.

```
\let\@oldmarginpar\marginpar
\def\@marginragged{\if@reversemargin\raggedleft\else\raggedright\fi}
\def\@foilmarginpar{\@ifnextchar[{\@foilmarginparRL}{\@foilmarginparR}}
\def\@foilmarginparRL[#1]#2{%
```

```
  \@oldmarginpar[{\@marginragged #1\par}]{{\@marginragged #2\par}}}
\def\@foilmarginparR#1{\@oldmarginpar{\@marginragged #1\par}}
\let\marginpar\@foilmarginpar
```

`\Black` `\globalColor` Stricly speaking, we probably shouldn't define these and they shouldn't be part of a FoilTEX class file, since they are not completely compatible with the color package. However, we still define them and use them for two purposes. First is backward compatibility with old FoilTEX documents, and second, so that things still work reasonably well with the old `colordvi` style file. They are both defined as no-ops.

```
\def\Black#1{#1}
\def\globalColor#1{#1}
```

`\today` Since this one isn't defined by default, we define it.

```
\newcommand\today{\ifcase\month\or
 January\or February\or March\or April\or May\or June\or
 July\or August\or September\or October\or November\or December\fi
 \space\number\day, \number\year}
```

`\lineskip` `\normallineskip` `\baselinestretch` `\parskip` `\parindent` `\jot` Next, we set a few of the basic space parameters. Note that we have increased the `\parindent` in the new version but retained the old value in compatibility mode for old versions of FoilTEX files. The value for baseline stretch is not strictly needed since that is the default. We put it here for completeness. We adjust the spacing set by `\jot` in the new mode, since we found we used this a lot in practice. We left it as the default in compatibility mode.

```
\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
\renewcommand\baselinestretch{1}
\setlength\parskip{18\p@ \@plus 4\p@ \@minus 4\p@}
\if@compatibility
 \setlength\parindent{15\p@}
 \setlength\jot{3\p@}
\else
 \setlength\parindent{30\p@}
 \setlength\jot{10\p@}
\fi
```

`\righthyphenmin` `\lefthyphenmin` We set these minimum hyphenation values to some large value so that we can disable hyphenation. This is unlike FoilTEX v1.0.1, where we turned off hyphenation by disabling the hyphenation tables. This is more or less essential here since the tables are already loaded by the LaTEX format. It does have the additional advantage in that the user can re-enable it simply by resetting these values.

```
\righthyphenmin=100
\lefthyphenmin=100
```

`\@eqnnum` `\theequation` We set the equation number in the default sans serif font (don't exactly know why this is necessary — maybe it is not!) and place it in arabic numerals.

```
\def\@eqnnum{\hbox{\reset@font\sffamily (\theequation)}}
\renewcommand\theequation{\arabic{equation}}
```

titlepage
\@pnumwidth
@tocrmarg
\@dotsep

We leave the next set of macros here, even though they are either irrelevant or taken care of by other things (like `\maketitle`), just in case someone wants to use them. They are not supported except in compatibility mode. The parameter settings are used only in the table of contents (which we don't have on foils), so these are irrelevant.

```
\if@compatibility
  \newenvironment{titlepage}{%
    \@restonecolfalse \newpage \thispagestyle{empty}
      \setcounter{page}{0}}{\newpage}
 \newcommand\@pnumwidth{1.55em}
 \newcommand\@tocrmarg {2.55em}
 \newcommand\@dotsep{4.5}
\fi
```

\sloppyfoils

We can be really sloppy with tolerances and fuzz on foils, so we set the parameters for this and invoke (probably could just avoid the definition all together).

```
\def\sloppyfoils{\tolerance 9000 \hfuzz 2\p@ \vfuzz 2\p@ \hbadness 2000}
\sloppyfoils
```

11.3 List environments: spacing

In this subsection, we set up all the code needed for list environments spacing. Most of this is extracted in some form or another from the basic LaTeX classes like `article` with some modifications for use in foils.

\leftmargin*

We begin with the list margin indentation parameters.

```
\setlength\leftmargini{25\p@}
\setlength\leftmarginii{22\p@}
\setlength\leftmarginiii{18.7\p@}
\setlength\leftmarginiv{17\p@}
\setlength\leftmarginv{10\p@}
\setlength\leftmarginvi{10\p@}
\setlength\leftmargin\leftmargini
```

\labelwidth
\labelsep

We need to set the label width and label separation parameters. Unfortunately, they were done in the wrong order in the old FoilTeX, so here we do it right but under compatibility mode, do it the old way so things don't change.

```
\if@compatibility
  \setlength\labelwidth\leftmargini\advance\labelwidth-\labelsep
  \setlength\labelsep{.5em}
\else
  \setlength\labelsep{10\p@}
  \setlength\labelwidth\leftmargini\advance\labelwidth-\labelsep
\fi
```

\partopsep

This is set fairly small, no reason to be too spacious.

```
\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
```

$\geq$20pt	`\@listIa`
17pt	`\@listIb`
14pt	`\@listIc`
12pt, shortform	`\@listId`

Table 5: Usage of `\@listI*` styles versus font size in LaTeX2e mode.

`\@listIa` `\@listIb` `\@listIc` `\@listId` These macros define the spacing for top level list environments. We define four default sets of parameters and overlap their usage in the various point size class option files. Table 5 shows the usage of each with respect to font size. (This table is somewhat different in compatibility mode, particularly for small relative font sizes in the larger class options. See Section 12 for the details.)

```
\newcommand\@listIa{\leftmargin\leftmargini
\topsep 14\p@ \@plus 2\p@ \@minus 4\p@
\parsep 14\p@ \@plus 4\p@ \@minus 4\p@
\itemsep 14\p@ \@plus 4\p@ \@minus 2\p@}
\newcommand\@listIb{\leftmargin\leftmargini
\topsep 8\p@ \@plus 2\p@ \@minus 2\p@
\parsep 2\p@ \@plus 1\p@ \@minus 1\p@
\itemsep \parsep}
\newcommand\@listIc{\leftmargin\leftmargini
\topsep 6\p@ \@plus 1\p@ \@minus 1\p@
\parsep 2\p@ \@plus 1\p@ \@minus 1\p@
\itemsep \parsep}
\newcommand\@listId{\leftmargin\leftmargini
\topsep 4\p@ \@plus 1\p@ \@minus 1\p@
\parsep 2\p@ \@plus 1\p@ \@minus 1\p@
\itemsep \parsep}
```

`\@listi*` Now we set the parameters for lower level lists. These are fixed throughout all the point sizes. Essentially, there is very little additional space here except the indentation. We allowed a fair amount of space in the top level, however. We close this section with a call to `\normalsize` to make sure that all the internal list macros are defined appropriately. This call is also done in each of the `foil*.clo` files but before the above parameters sets are defined. This caused a problem by delaying the definition of `\@listi`. This is new in version 2.1.1. (Alternatively, we could move all this stuff *before* we load these `.clo` files but that seemed to be too much of a change.)

```
\newcommand\@listii{\leftmargin\leftmarginii
 \labelwidth\leftmarginii\advance\labelwidth-\labelsep
 \topsep \z@
 \parsep \z@
 \itemsep \parsep}
\newcommand\@listiii{\leftmargin\leftmarginiii
 \labelwidth\leftmarginiii\advance\labelwidth-\labelsep
 \partopsep 1\p@ \@plus 0\p@ \@minus 1\p@
 \topsep \z@
 \parsep \z@
 \itemsep \topsep}
```

```
\newcommand\@listiv{\leftmargin\leftmarginiv
 \labelwidth\leftmarginiv\advance\labelwidth-\labelsep}
\newcommand\@listv{\leftmargin\leftmarginv
 \labelwidth\leftmarginv\advance\labelwidth-\labelsep}
\newcommand\@listvi{\leftmargin\leftmarginvi
 \labelwidth\leftmarginvi\advance\labelwidth-\labelsep}
\normalsize
```

`\zerolistvertdimens` We define this macro in the new version so we have a simple mechanism for shrinking as much as possible all the vertical space in list environments.

```
\if@compatibility
 \let\zerolistvertdimens\relax
\else
 \def\zerolistvertdimens{\parskip0pt\topsep0pt\partopsep0pt%
    \parsep0pt\itemsep0pt}
\fi
```

11.4 List environments: structure

Here we take up the structure of list environments including the standard environments like `verse` and `description`

`\@item` We start by redefining `\@item` in compatibility mode to get the effect we had before for protecting the item label from spurious color corruption. This of course will only work with the `colordvi` style. As mentioned above, this is not the recommended way to do color in the new LaTeX. This code is mostly taken from the new LaTeX and modified to fit the old needs. The only real change is in the addition of the `\globalColor` macro surrounding the `\makelabel`.

```
\if@compatibility
 \def\@item[#1]{%
   \if@noparitem
     \@donoparitem
   \else
     \if@inlabel \indent \par \fi
     \ifhmode \unskip\unskip \par \fi
     \if@newlist
       \if@nobreak
         \@nbitem
       \else
         \addpenalty\@beginparpenalty
         \addvspace\@topsep \addvspace{-\parskip}
       \fi
     \else
       \addpenalty\@itempenalty \addvspace\itemsep
     \fi
     \global\@inlabeltrue
   \fi
 \everypar{\global\@minipagefalse\global\@newlistfalse
   \if@inlabel
     \global\@inlabelfalse \hskip -\parindent \box\@labels \penalty\z@
   \fi
   \everypar{}}
```

```
\global\@nobreakfalse
\if@noitemarg \@noitemargfalse
  \if@nmbrlist \refstepcounter{\@listctr}\fi
\fi
\sbox\@tempboxa{\globalColor{\makelabel{#1}}} \global\setbox\@labels
\hbox{\unhbox\@labels \hskip \itemindent
  \hskip -\labelwidth \hskip -\labelsep
  \ifdim
    \wd\@tempboxa >\labelwidth \box\@tempboxa
  \else
    \hbox to\labelwidth {\unhbox\@tempboxa}
  \fi
  \hskip \labelsep}
\ignorespaces
}
\fi
```

`\theenum*` `\labelenum*` `\labelitem*`

We set up the default format for the enumeration list numbers at the four default levels. Similarly, we default the tags for the itemized lists.

```
\renewcommand\theenumi{\arabic{enumi}}
\renewcommand\theenumii{\alph{enumii}}
\renewcommand\theenumiii{\roman{enumiii}}
\renewcommand\theenumiv{\Alph{enumiv}}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\newcommand\labelitemi{$\m@th\bullet$}
\newcommand\labelitemii{{\normalfont\bfseries --}}
\newcommand\labelitemiii{$\m@th\ast$}
\newcommand\labelitemiv{$\m@th\cdot$}
```

`\descriptionlabel` `description` `verse` `quotation` `quote`

Finally, we predefine the standard environments which are built from the list environment.

```
\newcommand\descriptionlabel[1]{\hspace\labelsep \normalfont\bfseries #1}
\newenvironment{description}{\list{}{\labelwidth\z@
  \itemindent-\leftmargin \let\makelabel\descriptionlabel}}{\endlist}
\newenvironment{verse}{\let\\=\@centercr
 \list{}{\itemsep\z@ \itemindent -1.5em\listparindent \itemindent
 \rightmargin\leftmargin\advance\leftmargin 1.5em}\item[]}{\endlist}
\newenvironment{quotation}{\list{}{\listparindent 1.5em
 \itemindent\listparindent
 \rightmargin\leftmargin \parsep 0\p@ \@plus 1\p@}\item[]}{\endlist}
\newenvironment{quote}{\list{}{\rightmargin\leftmargin}\item[]}{\endlist}
```

11.5 Page layout parameters

In this section we setup the parameters that adjust the page layout, including margins, text height and width, etc.

\oddsidemargin
\evensidemargin
\topmargin
\headsep
\headheight
\footskip
\head@footskip
\footnotesep
\footins
\@mpfootins

We start with the margin settings and the header settings. Some things change a little in the new version so we keep the old versions in compatibility mode. This is mostly for the footer stuff. Of course, the `\footheight`, `@maxsep` and `\@dblmaxsep` are no-ops in all cases but is only defined in compatibility mode. In the default mode (i.e., non-compatibility) the `\head@footskip` length contains the sum of the vertical space taken by the header and footer. The description of this is given below. Initially it is set to one inch and the `\footskip` is calculated from this.

```
\setlength\oddsidemargin{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\topmargin{0\p@}
\setlength\headsep{14\p@}
\setlength\headheight{15\p@}
\if@compatibility
 \setlength\footheight{25\p@}
 \setlength\footskip{45\p@}
 \setlength\@maxsep{20\p@}
 \setlength\@dblmaxsep{20\p@}
\else
 \newdimen\head@footskip
 \setlength\head@footskip{1in}
 \setlength\footskip{\head@footskip}
 \addtolength\footskip{-\headsep}
 \addtolength\footskip{-\headheight}
\fi
\setlength\footnotesep{10\p@}
\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
\skip\@mpfootins = \skip\footins
```

\marginparwidth
\marginparsep
\marginparpush

Here we set up the dimensions for margin paragraphs. We base these on one inch margins, so if the user choses to shrink these margins and plans to use margin paragraphs, they should be careful about these dimensions.

```
\setlength\marginparwidth{54\p@}
\setlength\marginparsep{10\p@}
\setlength\marginparpush{5\p@}
```

\textheight
\textwidth
\@foilheight
\@foilwidth
\@rotdimens
\@defaultdimens
\setp@gelayoutdimens

We have added in the new version a `\@foilheight` and `\@foilwidth` which are internal dimensions of the printable area of the page (excluding margin paragraphs and the like). They are setup initially as the paper dimensions less two inches (each) since we set the default top margin and side margins to zero. This leaves exactly a one inch margin all the way around in all supported paper sizes. At the `\begin{document}` we invoke a macro (`\setp@gelayoutdimens`) that adjusts these dimensions to account for user settings for page layout. Furthermore, it defines the internal commands `\@rotdimens` and `\@defaultdimens` which are used to perform the rotation of individual pages (including header and footer). Set up this way, the macros can take into account the rotation of the whole document done by a `landscape` class option (since this is based only on the paper dimensions).

In compatibility mode, we use the parameter settings of FoilTEX v1.0.1 for text height but in the new version we let the text height expand to its largest possible subject to space for header and margins. Also in this mode, we disable the `\@rotdimens` and `\@defaultdimens` since we didn't support this in the old version.

```
\if@compatibility
 \setlength\textheight{7.6in}
 \setlength\textwidth{6.5in}
 \let\@rotdimens\relax
 \let\@defaultdimens\relax
\else
 \newdimen\@foilheight
 \newdimen\@foilwidth
 \setlength\textheight{\paperheight}
 \addtolength\textheight{-2in}
 \addtolength\textheight{-\head@footskip}
 \setlength\textwidth{\paperwidth}
 \addtolength\textwidth{-2in}
 \def\setp@gelayoutdimens{%
  \setlength\head@footskip{\footskip}
  \addtolength\head@footskip{\headsep}
  \addtolength\head@footskip{\headheight}
  \setlength\@foilheight{\textheight}
  \addtolength\@foilheight{\head@footskip}
  \setlength\@foilwidth{\textwidth}
  \def\@rotdimens{\textheight\@foilwidth \textwidth\@foilheight
        \addtolength\textheight{-\head@footskip}
        \vsize\textheight \hsize\textwidth \linewidth\textwidth
        \columnwidth\textwidth \@colroom\textheight \@colht\textheight}
  \def\@defaultdimens{\textheight\@foilheight \textwidth\@foilwidth
        \addtolength\textheight{-\head@footskip}
        \vsize\textheight \hsize\textwidth \linewidth\textwidth
        \columnwidth\textwidth \@colroom\textheight \@colht\textheight}
   \@defaultdimens}   % end of \def\setp@gelayoutdimens
\fi
\AtBeginDocument{\if@compatibility\else\setp@gelayoutdimens\fi}
```

`\@*penalty`

A number of parameters for float separation and the like are set to the defaults in LaTeX (both old and new versions) so we leave out settings for these in this class file. We do set up the penalty values (not really knowing what they do) to values consistent with other class files and old style files.

```
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
\@beginparpenalty -\@lowpenalty
\@endparpenalty -\@lowpenalty
\@itempenalty -\@lowpenalty
```

`\arraycolsep` `\tabcolsep` `\arrayrulewidth` `\doublerulesep` `\tabbingsep`

Here are some settings for space in tables and tabbing environments. These aren't terribly critical but something has to be set.

```
\setlength\arraycolsep{10\p@}
\setlength\tabcolsep{12\p@}
\setlength\arrayrulewidth{1\p@}
\setlength\doublerulesep{3\p@}
\setlength\tabbingsep\labelsep
```

`\fboxsep` `\fboxrule`

And finally some settings for the frame box rule and separation. We make the separation a bit bigger to compensate for the larger fonts.

```
\if@compatibility
 \setlength\fboxsep{6\p@}
\else
 \setlength\fboxsep{10\p@}
\fi
\setlength\fboxrule{1\p@}
```

11.6 The title page

In this section we describe the construction of the title page macros.

\abovetitleskip
\titleauthorskip
\authorauthorskip
\authordateskip
\dateabstractskip

I'm a big fan of letting the user easily adjust spacing in lots of different contexts, so we set all the vertical spacing parameters as adjustable lengths instead of hard-coding their values. This may take too many TEX resources but hopefully not (I'm sure I'll get messages about this if it does). Each of these dimensions adjust vertical space on the title page. Each name indicates where the space is, so the user can easily shrink or stretch space as necessary. The default values are as in the old FoilTEX v1.0.1, so this should cause no additional confusion.

```
\newlength\abovetitleskip
\newlength\titleauthorskip
\newlength\authorauthorskip
\newlength\authordateskip
\newlength\dateabstractskip
\setlength\abovetitleskip{2em}
\setlength\titleauthorskip{1.5em}
\setlength\authorauthorskip{.5em}
\setlength\authordateskip{1em}
\setlength\dateabstractskip{1em}
```

\maketitle
\@maketitle

We don't do too much that is not in `article.cls` for defining the `\maketitle` and `\@maketitle` commands. About the only differences are the addition of these adjustable spacing parameters. (By changing the way footnote marks are done overall, we no longer need to change the definitions within the `\maketitle` as in the old FoilTEX.) This fixed a bug when the user tried to redefine the footnote mark in terms of the footnote symbols.

```
\def\maketitle{\par
  \begingroup
    \setcounter{page}{0}
    \def\thefootnote{\fnsymbol{footnote}} \newpage
    \@maketitle
    \thispagestyle{foilheadings}
    \@thanks
  \endgroup
  \setcounter{footnote}{0}
  \let\maketitle\relax \let\@maketitle\relax
  \gdef\@thanks{}\gdef\@author{}\gdef\@title{}\let\thanks\relax%
}
```

Note that in `\@maketitle` we use the `\zerolistvertdimens` to get all the list spacing shrunk to zero so we can control it explicitly with the above parameters.

```
\def\@maketitle{\newpage
  \zerolistvertdimens
```

```
  \if@compatibility\else
    \advance\abovetitleskip -\baselineskip % \null adds this space
  \fi
  \null\vskip\abovetitleskip
  \begin{center}
     {\Large\bfseries \@title \par}
     \vskip\titleauthorskip
     {\lineskip \authorauthorskip
       \begin{tabular}[t]{c}\@author\end{tabular}
       \par}
     \vskip\authordateskip {\@date}
  \end{center}
  \par\vfil
}
```

\abstractname
abstract

To complete the title page, we add the code for the abstract environment, if desired by the user. We define the \abstractname so the user can do this in their favorite language. We then construct the environment using the new parameters and still try to simulate the old layout.

```
\newcommand\abstractname{Abstract}
\newenvironment{abstract}{%
  \if@compatibility
    \dateabstractskip\parskip
    \advance\dateabstractskip\topsep
    \advance\dateabstractskip\baselineskip
  \fi
  \vskip\dateabstractskip
  \centerline{\reset@font\bfseries\abstractname}
  \if@compatibility\vspace{-.5em}\vspace{0\p@}\fi
  \list{}{\listparindent 1.5em
          \itemindent\listparindent \rightmargin\leftmargin
          \zerolistvertdimens
         }\item[]%
  }{\endlist\vfill}
```

11.7 Page style

We have now arrived at the part of the code that deals with the page style, headers, footers, etc.

\leftheader
\rightheader
\rightfooter
\@leftheader
\@rightheader
\@rightfooter

We start with three corners of the page (all but the bottom left corner). These are simply controlled using the next set of declaration macros. They each define internal macros which are placed in the appropriate place in the header and footer. The default definitions for the header is empty and for the right footer is the page number.

```
\newcommand\leftheader[1]{\gdef\@leftheader{#1}}
\newcommand\rightheader[1]{\gdef\@rightheader{#1}}
\newcommand\rightfooter[1]{\gdef\@rightfooter{#1}}
\leftheader{}
\rightheader{}
\rightfooter{\quad\textsf{\thepage}}
```

\ifLogo
\LogoOff
\LogoOn
\Restriction
\MyLogo

The left footer is a combination of two macros, \MyLogo and \Restriction. By default, the logo appears and is set to be the string "Typeset by FoilTEX". The \Restriction is empty, but can be used for things like "Confidential" or perhaps for something that is aimed to the intended audience. The \LogoOff and \LogoOn declarations are conditional toggles that turn on or off the setting of the logo. The conditional they set is tested by \ifLogo, which defaults to true. This gives a simple mechanism for turning it off completely, or perhaps on individual pages as necessary.

```
\newif\ifLogo \Logotrue
\newcommand\LogoOff{\Logofalse}
\newcommand\LogoOn{\Logotrue}
\newcommand\Restriction[1]{\gdef\@Restriction{#1}}
\Restriction{}
\newcommand\MyLogo[1]{\gdef\@MyLogo{\ifLogo{#1}\else\fi}}
\MyLogo{-- Typeset by \FoilTeX\ --}
```

\ps@foilheadings

Now that we have the code so the user can easily set up the contents of the page style, we need to use these to construct the page style. We have added some extras to this new version. In particular, we now support a rule below the header and a rule above the footer (much as is available with fancyheadings.sty). We still wrap everything in the \Black macro for backwards compatibility and just in case the user decides to use colordvi.sty, though we don't recommend it or really support it. Note that these headings are different on the title page. We also clean this up abit from the old version by using \let to do the even pages.

We made changes here to fix a problem with the footrule being placed over a graphic in the footer if the graphic was too tall.

```
\newsavebox\@tempfootbox
\newdimen\@tempfootht
\newcommand\ps@foilheadings{\let\@mkboth\@gobbletwo
  \def\@oddhead{%
    \ifnum \c@page>0
      {\Black{%
      \if@header@rule\hbox to\z@{\rule[-5\p@]{\textwidth}{1\p@}\hss}\fi
      \reset@font\tiny
      \@leftheader\hfil\@rightheader}}%
    \else
      \hfill
    \fi}%
  \def\@oddfoot{%
    \ifnum \c@page>0
      {\Black{%
      \sbox\@tempfootbox{\tiny\@MyLogo\ \@Restriction\hfil\@rightfooter}%
      \@tempfootht\ht\@tempfootbox
      \advance\@tempfootht 5.66666\p@
      \if@footer@rule%
      \hbox to\z@{\rule[\@tempfootht]{\textwidth}{1\p@}\hss}%
      \fi%
      \reset@font\tiny
     \@MyLogo\ \@Restriction\hfil\@rightfooter}}%
    \else
      {\Black{\hfil\reset@font\footnotesize%
```

```
812       \@MyLogo\ \@Restriction\hfil}}%
813     \fi}%
814   \let\@evenhead\@oddhead%
815   \let\@evenfoot\@oddfoot%
816 }
817 \ps@foilheadings
```

\pagenumbering
\onecolumn
\mark

Finally, we set the page numbering style, force this to one column (who'd want two column foils anyway) and fill the `\mark` with empty stuff just to cover our bases (this is irrelevant on foils).

```
818 \pagenumbering{arabic}
819 \onecolumn
820 \mark{{}{}}
```

11.8 Bibliography or References

\refname
\bibindent
\newblock
thebibliography

Though they may not be used very much, we do support a bibliography environment. This is a modified version of that from `article.cls`. The essential changes are the deletion of anything that adjusts the header (e.g., `\@mkboth`) since this is irrelevant for foils and the replacement of a sectioning macro with just the placement of the `\refname`, which is defined first.

```
821 \newcommand\refname{References}
822 \newdimen\bibindent
823 \setlength\bibindent{1.5em}
824 \newcommand\newblock{}
825 \newenvironment{thebibliography}[1]{
826  \vskip 3.5ex \@plus -1ex \@minus -.2ex
827  \noindent{\large\bfseries\refname}
828  \vskip 2.3ex \@plus .2ex
829  \list{\@biblabel{\arabic{enumiv}}}%
830  {\settowidth\labelwidth{\@biblabel{#1}}%
831  \leftmargin\labelwidth
832  \advance\leftmargin\labelsep
833  \if@openbib
834    \advance\leftmargin\bibindent
835    \itemindent -\bibindent
836    \listparindent \itemindent
837    \parsep \z@
838  \fi
839  \usecounter{enumiv}%
840  \let\p@enumiv\@empty
841  \renewcommand\theenumiv{\arabic{enumiv}}}
842  \if@openbib
843    \renewcommand\newblock{\par}
844  \else
845    \renewcommand\newblock{\hskip .11em \@plus .33em \@minus -.07em}
846  \fi
847  \sloppy\clubpenalty4000\widowpenalty4000%
848  \sfcode`\.=\@m\relax}%
849  {\def\@noitemerr{\@latex@warning{Empty `thebibliography' environment}}%
850  \endlist}
```

11.9 Non-floating floats

In this section we construct new definitions for float-type environments that don't actually float! They will look to the user like any other float environment but will act as if given the [H] (put it HERE!) placement.

\abovecaptionskip
\belowcaptionskip
\abovefloatskip
\captionfraction

First we set up some parameters that control the spacing around and within floats. They have obvious meanings based on their names. The default values for these spaces are as in standard LaTeX where most of them are hardcoded. Here we let the user set them.

Many users want the caption to have different width than the text width, so here it can be set by the user. The old FoilTeX method was to set the dimension \captionwidth. This actually introduced a bug when combined with \rotatefoilhead. In this new version of FoilTeX, we do two things. First, we fix the bug, but we also introduce an alternative (and preferred) method for adjusting the caption width with \captionfraction. As with other LaTeX fractions (for floats), this is reset with the \renewcommand. The default value is 1.0.

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\newlength\@captionwidth
\newlength\captionwidth
\newcommand\captionfraction{1.0}
\newlength\abovefloatskip
\setlength\abovecaptionskip{15\p@ \@plus 5\p@ \@minus 5\p@}
\setlength\belowcaptionskip{0\p@}
\setlength\captionwidth\z@
\setlength\abovefloatskip{20\p@ \@plus 5\p@ \@minus 10\p@}
```

\@makecaption
\@caption
\caption

For non-floating environments like "figure" and "table", we are going to provide a *-form which is unnumbered. To enable that in an easy way, we make a conditional which can be used to determine whether we are to include counter or not in the caption. This requires some modifications to a couple of macros. Other changes are noted below. We modify the standard definition for \@makecaption so that it uses the \captionwidth parameter (old method) or the \captionfraction (new method). Then we modify the \@caption macro, removing all the extraneous stuff that added things to the table of contents and the like. Any text in the optional argument to a \caption command is just swallowed. Finally we modify the definition of \caption to skip incrementing a counter in *starmode* (since there isn't one!).

```
\newif\if@starmode\@starmodefalse
\newcommand\@makecaption[2]{%
```

To set the internal \@captionwidth we first check for the old method. If \captionwidth was set by the user, then we take the minimum of this value and \hsize. If not, then the new method sets the caption width by taking the fraction of the \hsize in all cases.

```
 \ifdim\captionwidth>\z@
   \ifdim\captionwidth>\hsize
     \setlength\@captionwidth\hsize
   \else
     \setlength\@captionwidth\captionwidth
```

```
    \fi
  \else
    \setlength\@captionwidth{\captionfraction\hsize}
  \fi
```

Now we layout the caption itself in something no bigger than the above dimension, but ignoring the counter in *-mode.

```
  \vskip \abovecaptionskip
  \if@starmode\sbox\@tempboxa{#2}\else\sbox\@tempboxa{#1: #2}\fi%
  \ifdim \wd\@tempboxa >\@captionwidth
    \centering\parbox[t]{\@captionwidth}{\unhbox\@tempboxa\par}
  \else
    \hbox to\hsize{\hfil\box\@tempboxa\hfil}
  \fi
  \vskip\belowcaptionskip}
 \long\def\@caption#1[#2]#3{\par \begingroup \@parboxrestore \normalsize
  \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
  \endgroup}
 \def\caption{\if@starmode\else\refstepcounter\@captype\fi%
  \@dblarg{\@caption\@captype}}
```

`\@xfloat` To construct non-floating floats, we need to do a couple of things. First, we take out anything from the `\@xfloat` that refers to placement, since these don't move. We also take out all the float penalty code and anything that saves this stuff in the float list since that is now irrelevant. This cleans up this code considerably. Basically, all it does now is build a box with the appropriate grouping and wrapping. Note that this starts a couple of groups which have to be closed later (at the end of the environment).

```
 \def\@xfloat#1[#2]{%
    \def \@captype {#1}%
    \ifhmode \@bsphack \fi \vskip\abovefloatskip
    \vbox\bgroup \color@begingroup \normalcolor
    \hsize\columnwidth \@parboxrestore
    \if@nobreak
      \def\outer@nobreak{\global\@nobreaktrue}\global\@nobreakfalse
    \fi}
```

`\end@nonfloat` This macro is used to close the non-float environments. We could have modified the `\end@float` and used that but it seemed like this was simpler.

```
 \def\end@nonfloat{\par\vskip\z@skip
    \color@endgroup
    \outer@nobreak
    \egroup}
```

`\newnonfloat` This new macro is a general mechanism for defining or creating non-floating environments. It is probably similar to what can be found in `float.sty`, but we didn't check that for our purposes. Each invocation of this macro will create *two* environments, a non-starred version and a starred version. The starred version is not numbered (and so non-reference-able). The first thing this macro does is define the *name* of the environment, i.e., the non-float type used in the caption. It next defines a counter for the non-starred version of the environment. It sets the style of the counter to arabic (this can probably be overriden by the user, but

we haven't tested this). Next it sets the style for the caption structure just as in standard LaTeX. Finally, it defines the actual environments in terms of `\@float` and `\end@nonfloat`. The former is defined within LaTeX in terms of `\@xfloat` (modified above).

```
\newcommand\newnonfloat[2]{%
  \expandafter\newcommand\csname#1name\endcsname{#2}%
  \expandafter\newcounter{#1}%
  \expandafter\renewcommand\csname the#1\endcsname
        {\@arabic\csname c@#1\endcsname}%
  \expandafter\newcommand\csname fnum@#1\endcsname
    {\csname#1name\endcsname~\csname the#1\endcsname}%
  \expandafter\newenvironment{#1}{\@float{#1}}{\end@nonfloat}%
  \expandafter\newenvironment{#1*}{\@float{#1}\@starmodetrue}%
                                  {\end@nonfloat}%
}
```

`table` `figure` `\figurename` `\tablename`

Finally, we use the above constructor to give definitions for the standard `table`, `table*`, `figure`, and `figure*` environments. Note that the starred forms mean unnumbered versions and *not* two column versions. The `\newnonfloat` automatically defines a "name" macro for the non-float. So, for example, the following declarations define `\tablename` and `\figurename` to "Table" and "Figure", respectively.

```
\newnonfloat{table}{Table}
\newnonfloat{figure}{Figure}
```

11.10 The foil title macros and foil rotation

We next describe the macros for starting new foils. There are some enhancements to the old version here. In particular, we now have a simple way to support rotated foils.

`\foilheadskip`

Of course, we have a length parameter that we use to set the vertical spacing around the foil title (called `\foilhead`, perhaps in error). This is set to a new value in the new FoilTeX, with much more shrinkability. We hope this reduces the amount of manual fiddling usually needed to squeeze that extra bit of stuff on a single foil. The default space between the foil title and the text below it is set as `\parskip+\baselineskip+\foilheadskip+[option]` where the `[option]` comes from the optional argument in the `\foilhead` macro.

```
\newlength\foilheadskip
\if@compatibility
  \setlength\foilheadskip{.25in}
\else
  \setlength\foilheadskip{18\p@ \@plus 0\p@ \@minus 18\p@}
\fi
```

`\foil@rot@start`

This macro is used exclusively for individual foil rotation and is only called if the `dvips` or `dvipsone` option is in place. In effect, this issues a PostScript call to translate vertically by the current `\@foilheight` and then do the necessary rotation. We don't need to do any graphics save or restore in this process since it is only invoked at the very beginning of a page where this has just happened.

```
\def\foil@rot@start{\special{ps:
\if@dvips
  0 \strip@pt\@foilheight\space 72.27 div Resolution mul translate
  90 neg rotate
\else% dvipsone
  0 \strip@pt\@foilheight\space 72.27 mul 65536 mul rmoveto
  90 rotate
\fi}%
}
```

`\ifcur@rot@state` `\ifnew@rot@state` We need some mechanism to test the current state of rotation so we know whether we should issue the rotation call again if there are two invocations of rotated foils in a row. We capture this in these conditionals.

```
\newif\ifcur@rot@state
\newif\ifnew@rot@state
```

`\foilhead` `\rotatefoilhead` We can now define the macros used for starting new foils, both in standard orientation and rotated. (Note that standard is not by default portrait if the `landscape` class option is set.) These are both defined in terms of `\@foilhead` which is used to parse the optional argument for vertical space adjustment. `\foilhead` sets the state of the "new" foil to non-rotated, then happily goes off to `\@foilhead`. `\rotatefoilhead` is not supported in compatibility mode so it isn't even defined there. Rotation is not supported unless the `dvips` or `dvipsone` class option is declared but we only issue a warning in this case and set the "new" state to non-rotated. If all the pieces are in place, it sets the "new" state to rotated. Finally, it goes off to `\@foilhead`.

```
\newcommand\foilhead{\new@rot@statefalse%       set state for the page
   \@ifnextchar[{\@foilhead}{\@foilhead[0\p@]}}
\if@compatibility
\else
  \newcommand\rotatefoilhead{%
    \if@dvips
      \new@rot@statetrue
    \else\if@dvipsone
      \new@rot@statetrue
    \else
      \@warning{Without the 'dvips' or 'dvipsone' option, rotation is
          not supported}
      \new@rot@statefalse
    \fi\fi
    \@ifnextchar[{\@foilhead}{\@foilhead[0\p@]}}
\fi
```

`\@foilhead` The new version of the `\@foilhead` macro now has more work to do. First, it clears the previous page so we start fresh on a new page. Next, it checks if we are supposed to rotate this page. If so, it sets the current page state to rotated and establishes the dimensions for the rotated page (`\@rotdimens`, see Section 11.5). If not, it sets the current page state to non-rotated and establishes the non-rotated page dimensions (`\@defaultdimens`, see Section 11.5). Then it sets the font in large boldface, centers the title and adjusts the vertical spacing. We also wrap this header in a color group and reset to normal color so it doesn't get corrupted by other things.

```
\def\@foilhead[#1]#2{\vfill\eject
  \ifnew@rot@state
    \cur@rot@statetrue\@rotdimens
  \else
    \cur@rot@statefalse\@defaultdimens
  \fi
  {\color@begingroup\normalcolor
   \reset@font\large\bfseries\centering#2\par\null\color@endgroup}%
  \advance\foilheadskip by #1 \vspace{\foilheadskip}
  \advance\foilheadskip by -#1}
```

`\@shipoutsetup` We do some trickery here to get the right things to happen at shipout to enable rotation. Namely, before we do any shipout, if rotation is in effect, we issue the `\foil@rot@start` (handled by `dvips`) and then do the shipout. This enables us to interact with perhaps other utilities that muck with these internals and make sure that the rotation stuff happens at exactly the right point. The trickery is standard stuff, save the old definition, and merge the saved defintion with the new stuff added.

```
%\let\old@shipoutsetup\@shipoutsetup
%\def\@shipoutsetup{%
%  \ifcur@rot@state\foil@rot@start\fi
%  \old@shipoutsetup}
```

`\@begindvi` Unfortunately, They changed the output routine so this old stuff above didn't work anymore, which is why it is commented out. Below is a hack that resolves the problem but some of Them (the LaTeX3 team) didn't like it because it misuses the command `\@begindvi`. We use it anyway. First check to see if They change things again in the future with `\CheckCommand*`. If so, this will warn the user (and so me). Next we hack in something to get the rotation effects we need.

```
\CheckCommand*\@begindvi{%
  \unvbox \@begindvibox
 \global\let \@begindvi \@empty}
\def \@begindvi{%
  \ifcur@rot@state\foil@rot@start\fi
  \unvbox \@begindvibox
 %\global\let \@begindvi \@empty
  \gdef\@begindvi{\ifcur@rot@state\foil@rot@start\fi\@empty}
}
```

`\@outputpage` The more recommended fix follows. It really isn't very robust but it is what They recommended. Essentially, I check the old definition of `\@outputpage` (the one from LaTeX2_ε, 1995/06/01 patch level 3) and see if this has changed at all. If it has, then the user gets a warning to this effect. I then simply redefine it the way I want it based on this definition. All this does is provide some warning in case the official definition changed when I wasn't looking.

```
%\CheckCommand*\@outputpage{%
%\begingroup
% \set@typeset@protect
% \@shipoutsetup
% \let \protect \noexpand
% \shipout \vbox{%
```

```
%   \set@typeset@protect
%   \aftergroup\set@typeset@protect
%   \@begindvi
%   \vskip \topmargin
%   \moveright\@themargin \vbox {%
%     \setbox\@tempboxa \vbox to\headheight{%
%       \vfil
%       \color@hbox
%         \normalcolor
%         \hb@xt@\textwidth {%
%           \let \label \@gobble
%           \let \index \@gobble
%           \let \glossary \@gobble %% 21 Jun 91
%           \@thehead
%           }%
%       \color@endbox
%       }%                          %% 22 Feb 87
%     \dp\@tempboxa \z@
%     \box\@tempboxa
%     \vskip \headsep
%     \box\@outputbox
%     \baselineskip \footskip
%     \color@hbox
%       \normalcolor
%       \hb@xt@\textwidth{%
%         \let \label \@gobble
%         \let \index \@gobble      %% 22 Feb 87
%         \let \glossary \@gobble   %% 21 Jun 91
%         \@thefoot
%         }%
%     \color@endbox
%     }%
%   }%
% \endgroup
% \global \@colht \textheight
% \stepcounter{page}%
% \let\firstmark\botmark
%}
%\def\@outputpage{%
%\begingroup
% \set@typeset@protect
% \@shipoutsetup
% \let \protect \noexpand
% \shipout \vbox{%
%   \set@typeset@protect
%  \aftergroup\set@typeset@protect
%   \@begindvi
% \ifcur@rot@state\foil@rot@start\fi  %% added by for foils.cls
%   \vskip \topmargin
%   \moveright\@themargin \vbox {%
%     \setbox\@tempboxa \vbox to\headheight{%
%       \vfil
%       \color@hbox
%         \normalcolor
```

```
% \hb@xt@\textwidth {%
%   \let \label \@gobble
%   \let \index \@gobble
%   \let \glossary \@gobble %% 21 Jun 91
%   \@thehead
%   }%
% \color@endbox
% }%                          %% 22 Feb 87
% \dp\@tempboxa \z@
% \box\@tempboxa
% \vskip \headsep
% \box\@outputbox
% \baselineskip \footskip
% \color@hbox
%   \normalcolor
%   \hb@xt@\textwidth{%
%     \let \label \@gobble
%     \let \index \@gobble      %% 22 Feb 87
%     \let \glossary \@gobble   %% 21 Jun 91
%     \@thefoot
%     }%
% \color@endbox
% }%
%  }%
% \endgroup
% \global \@colht \textheight
% \stepcounter{page}%
% \let\firstmark\botmark
%}
```

11.11 Theorem-like environments

The next few macros set up special environments for theorem-like structures. The unstarred forms are numbered; the starred forms are unnumbered. We use slanted fonts for all of these. User's can easily change this if they wish. Both the starred and unstarred are created with one simple `\newtheorem` declaration (similar to the non-floats of Section 11.9). We try to use as much of the default LaTeX stuff as possible here.

`\newtheorem` `\@Othm` `\@Nthm` `\@Sthm` `\@starthm` `\@xstarthm` `\@ystarthm`

As in the standard definition, we have to deal with optional arguments to the `\newtheorem` declaration. We set this up an extra level deep by intercepting the default arguments with `\@Othm` and `\@Nthm` and then passing the arguments down one level simultaneously to the new `\@Sthm` and the old `\@othm` and `\@nthm`. The `\@Sthm` defines the starred name and the environment in terms of the `\@starthm`, which again recursively calls subconstructs to deal with the second optional argument and which do the actual environment construction. This is very parallel to standard LaTeX. Notice that we put everything in slanted text.

```
\def\newtheorem#1{\@ifnextchar[{\@Othm{#1}}{\@Nthm{#1}}}
\def\@Othm#1[#2]#3{\@Sthm{#1}{#3}\@othm{#1}[#2]{#3}}
\def\@Nthm#1#2{\@Sthm{#1}{#2}\@nthm{#1}{#2}}
\def\@Sthm#1#2{{{\global\@namedef{#1*}{\@starthm{#2}}
   \global\@namedef{end#1*}{\@endtheorem}}}}
```

```
\def\@starthm#1{\@ifnextchar[{\@ystarthm{#1}}{\@xstarthm{#1}}}
\def\@xstarthm#1{\@beginstartheorem{#1} \ignorespaces}
\def\@ystarthm#1[#2]{\@opargbeginstartheorem{#1}{#2}\ignorespaces}
\def\@begintheorem#1#2{\trivlist
      \item[\hskip\labelsep{\bfseries #1\ #2. }]\slshape}
\def\@opargbegintheorem#1#2#3{\trivlist
      \item[\hskip\labelsep{\bfseries #1\ #2.\ [#3] }]\slshape}
\def\@beginstartheorem#1{\trivlist
      \item[\hskip\labelsep{\bfseries #1. }]\slshape}
\def\@opargbeginstartheorem#1#2{\trivlist
      \item[\hskip\labelsep{\bfseries #1.\ [#2] }]\slshape}
```

`Theorem` `Lemma` `Corollary` `Proposition` `Definition`

For convenience, we predefine a few of the more standard environments. We use up token space here so perhaps this will have to be disabled so the user can do this for themselves. Note that the names are UpperCased to avoid some conflicts with user definitions that might already be in place.

```
\newtheorem{Theorem}{Theorem}
\newtheorem{Lemma}{Lemma}
\newtheorem{Corollary}{Corollary}
\newtheorem{Proposition}{Proposition}
\newtheorem{Definition}{Definition}
```

`Proof` `\ProofBox`

Finally, we set up a default `Proof` environment, with a nifty little square box at the end (this typically would come from the `latexsym` package but we steal just what we need for this to work here.

```
\newenvironment{Proof}{\begin{trivlist}\item[] {\bfseries Proof.}}{%
  \ifhmode\nolinebreak[4]~$\ProofBox$\else$\ProofBox$\fi \end{trivlist}}
\DeclareMathSymbol\ProofBox{0}{flasy}{"32}
```

11.12 Boldface mathematics

`\bm` `\bmstyle` `boldequation*` `boldequation`

Happily, in the new LaTeX, this is all done much better, particularly, with font selection. What we do here is just define a simple macro for setting small pieces of mathematics in bold (italics) and entire equations. This is mostly for backward compatibility. Note that the `\bm` macro is now much more robust. It uses the `\mathpalette` to set the style of its contents in the style of its surrounding environment rather than in just `\textstyle` as was done in the earlier FoilTeX. There are two bold equation environments, a starred (unnumbered) form and the unstarred (numbered) form. These environments have their macros constructed explicitly, rather than with the `\newenvironment` declaration.

```
\def\bm#1{\mathpalette\bmstyle{#1}}
\def\bmstyle#1#2{\mbox{\boldmath$#1#2$}}
\@namedef{boldequation*}{\boldmath$$}
\@namedef{endboldequation*}{$$\global\@ignoretrue\unboldmath}
\def\boldequation{\boldmath$$\refstepcounter{equation}}
\def\endboldequation{\eqno\@eqnnum%
    $$\global\@ignoretrue\unboldmath}
⟨/package⟩
```

12 The class option files

In these subsections we simply do a dump of the code for the size class option files. There is nothing deep here going on. Mostly we set the font sizes for the different relative font scaling, and adjust some of the basic settings in each level for display skips and for top level lists. There are two things to note here. First, we have added some larger fonts (51pt) over what was directly available in the old version. Second, the list setting parameters are set differently in the new version but act exactly as they did before in compatibility mode.

New here in version 2.1.3 are definitions for the `\big` and related operators which are now scaled according to the default font size (with the exception of `shortform` where this is not needed). In base LaTeX, these are handled the same across the different point sizes (because those sizes varied so little). The numbers here are basically computed from the values in the base LaTeX. An alternative approach to this is the `exscale` package.

12.1 The `17pt` option

```
⟨*17pt⟩
\def\normalsize{\@setfontsize\normalsize\@xviipt{22}%
\abovedisplayskip 20\p@ \@plus 3\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 2\p@
\belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
\let\@listi\@listIb}
\normalsize

\def\small{\@setfontsize\small\@xivpt{18}%
\abovedisplayskip 16\p@ \@plus 2\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 1\p@
\belowdisplayshortskip 3\p@ \@plus 1\p@ \@minus 2\p@
\let\@listi\@listIc}

\def\footnotesize{\@setfontsize\footnotesize\@xiipt{15}%
\abovedisplayskip 13\p@ \@plus 2\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 1\p@
\belowdisplayshortskip 2\p@ \@plus 1\p@ \@minus 1\p@
\let\@listi\@listId}

\def\large{\@setfontsize\large\@xxpt\@xxvpt
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\let\@listi\@listIa}

\let\scriptsize=\footnotesize
\let\tiny=\footnotesize
\def\Large{\@setfontsize\Large\@xxvpt{32}\let\@listi\@listIa}
\def\LARGE{\@setfontsize\LARGE\@xxxpt{38}\let\@listi\@listIa}
\def\huge{\@setfontsize\huge\@xxxvipt{45}\let\@listi\@listIa}
```

```
\def\Huge{\@setfontsize\Huge\@xliiipt{54}\let\@listi\@listIa}

\def\big#1{{\hbox{$\left#1\vbox to14.5\p@{}\right.\n@space$}}}
\def\Big#1{{\hbox{$\left#1\vbox to19.5\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to24.5\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to30\p@{}\right.\n@space$}}}
⟨/17pt⟩
```

12.2 The 20pt option

```
⟨*20pt⟩
\def\normalsize{\@setfontsize\normalsize\@xxpt\@xxvpt
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\let\@listi\@listIa}
\normalsize

\def\small{\@setfontsize\small\@xviipt{22}%
\abovedisplayskip 20\p@ \@plus 3\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 2\p@
\belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
\let\@listi\@listIb}

\def\footnotesize{\@setfontsize\footnotesize\@xivpt{18}%
\abovedisplayskip 16\p@ \@plus 2\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 1\p@
\belowdisplayshortskip 3\p@ \@plus 1\p@ \@minus 2\p@
\let\@listi\@listIc}

\def\scriptsize{\@setfontsize\scriptsize\@xiipt{15}%
 \if@compatibility\else\let\@listi\@listId\fi}
\let\tiny=\scriptsize
\def\large{\@setfontsize\large\@xxvpt{32}}
\def\Large{\@setfontsize\Large\@xxxpt{38}}
\def\LARGE{\@setfontsize\LARGE\@xxxvipt{45}}
\def\huge{\@setfontsize\huge\@xliiipt{54}}
\if@compatibility
  \let\Huge=\huge
\else
  \def\Huge{\@setfontsize\huge\@lipt{62}}
\fi

\def\big#1{{\hbox{$\left#1\vbox to17\p@{}\right.\n@space$}}}
\def\Big#1{{\hbox{$\left#1\vbox to23\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to27\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to35\p@{}\right.\n@space$}}}
⟨/20pt⟩
```

12.3 The 25pt option

```
⟨*25pt⟩
\def\normalsize{\@setfontsize\normalsize\@xxvpt{32}%
```

```
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\let\@listi\@listIa}
\normalsize

\def\small{\@setfontsize\small\@xxpt\@xxvpt
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\if@compatibility
 \let\@listi\@listIb
\else
 \let\@listi\@listIa\fi
}

\def\footnotesize{\@setfontsize\footnotesize\@xviipt{22}%
\abovedisplayskip 20\p@ \@plus 3\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 2\p@
\belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
\let\@listi\@listIb}

\def\scriptsize{\@setfontsize\scriptsize\@xivpt{18}%
 \if@compatibility\else\let\@listi\@listIc\fi}
\def\tiny{\@setfontsize\tiny\@xiipt{15}%
 \if@compatibility\else\let\@listi\@listId\fi}
\def\large{\@setfontsize\large\@xxpt{38}}
\def\Large{\@setfontsize\Large\@xxvipt{45}}
\def\LARGE{\@setfontsize\LARGE\@xliiipt{54}}
\if@compatibility
  \let\huge=\LARGE
  \let\Huge=\LARGE
\else
  \def\huge{\@setfontsize\huge\@lipt{62}}
  \let\Huge=\huge
\fi

\def\big#1{{\hbox{$\left#1\vbox to21\p@{}\right.\n@space$}}}
\def\Big#1{{\hbox{$\left#1\vbox to29\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to36\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to44\p@{}\right.\n@space$}}}
⟨/25pt⟩
```

12.4 The 30pt option

```
⟨*30pt⟩
\def\normalsize{\@setfontsize\normalsize\@xxpt{38}%
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\let\@listi\@listIa}
```

```
\normalsize

\def\small{\@setfontsize\small\@xxvpt{32}%
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\if@compatibility
 \let\@listi\@listIb
\else
 \let\@listi\@listIa\fi
}

\def\footnotesize{\@setfontsize\footnotesize\@xxpt\@xxvpt
\abovedisplayskip 30\p@ \@plus 3\p@ \@minus 9\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 3\p@
\belowdisplayshortskip 7\p@ \@plus 3\p@ \@minus 4\p@
\if@compatibility
 \let\@listi\@listIb
\else
 \let\@listi\@listIa\fi
}

\def\scriptsize{\@setfontsize\scriptsize\@xviipt{22}%
 \if@compatibility\else\let\@listi\@listIb\fi}
\def\tiny{\@setfontsize\tiny\@xivpt{18}%
 \if@compatibility\else\let\@listi\@listIc\fi}
\def\large{\@setfontsize\large\@xxxvipt{45}}
\def\Large{\@setfontsize\Large\@xliiipt{54}}
\if@compatibility
  \let\LARGE=\Large
  \let\huge=\Large
  \let\Huge=\Large
\else
  \def\LARGE{\@setfontsize\LARGE\@lipt{62}}
  \let\huge=\LARGE
  \let\Huge=\LARGE
\fi

\def\big#1{{\hbox{$\left#1\vbox to25.5\p@{}\right.\n@space$}}}
\def\Big#1{{\hbox{$\left#1\vbox to34.5\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to43.5\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to52.5\p@{}\right.\n@space$}}}
⟨/30pt⟩
```

12.5 The `shortform` option

In this version, we set `\normalsize` and hang all smaller sizes off of this. For larger sizes, we take a simple short cut. We are making no attempt to optimize most of this, just to scale the fonts to a much smaller relative size.

Finally, because we want to change `\foilhead` so that it doesn't force a page break, we redefine `\@foilhead` in a much more simple manner. This is done in `\AtBeginDocument` since this file is read before the definitions in `foils.cls`. (We

also make a change to \parskip so that paragraphs are less separated.)

```
⟨*short⟩
\def\normalsize{\@setfontsize\normalsize\@xiipt{15}
\abovedisplayskip 12\p@ \@plus 3\p@ \@minus 4\p@
\belowdisplayskip \abovedisplayskip
\abovedisplayshortskip \z@ \@plus 2\p@
\belowdisplayshortskip 4\p@ \@plus 2\p@ \@minus 2\p@
\let\@listi\@listId}
\normalsize

\let\small\normalsize
\let\footnotesize\normalsize
\let\scriptsize=\footnotesize
\let\tiny=\footnotesize

\def\large{\@setfontsize\large\@xivpt{18}\let\@listi\@listId}
\def\Large{\@setfontsize\Large\@xviipt{22}\let\@listi\@listId}
\def\LARGE{\@setfontsize\LARGE\@xxpt{25}\let\@listi\@listId}
\def\huge{\@setfontsize\huge\@xxvpt{30}\let\@listi\@listId}
\let\Huge\huge

\AtBeginDocument{
\def\@foilhead[#1]#2{{\vspace{2em}\pagebreak[1]%
     \color@begingroup\normalcolor
     \reset@font\large\bfseries\centering#2\nopagebreak[4]%
     \par\null\color@endgroup}}%
\setlength\parskip{4\p@ \@plus 4\p@ \@minus 4\p@}
}
⟨/short⟩
```

13 Compatibility mode

We include here the code for generating the compatibilty mode style file. This is modelled after the LaTeX `article.sty`.

```
⟨*oldstyle⟩
\@obsoletefile{foils.cls}{foils.sty}
\LoadClass{foils}
⟨/oldstyle⟩
```

14 Font definitions

Here starts everything that sets up font information. For almost everything we use a new font family. These names are essentially the standard names with an `f` prefix. We could probably get away with just overriding the standard ones but we don't really know what we're doing here and this seems to work. We begin with a short section that sets up the default font characteristics. We absolutely need to do the family definition (and at this early stage) to make sure that the default LaTeX stuff is not loaded. (The `\normalsize` declaration in the `foil\@ptsize.clo` files think these are the LaTeX defaults of `{OT1}{cmr}{m}{n}` and these don't come in a large enough size!). The other lines are not strictly needed but we add them for

completeness.

```
⟨*fonts⟩
\def\f@encoding{OT1}
\def\f@family{fcmss}
\def\f@series{m}
\def\f@shape{n}
```

14.1 Font macro utilities

Here we add a couple of short hand macros for declaring font shapes. These are blazenly borrowed from a recent LaTeX 2_ε release (clever of those people to come up with such nice shorthand). Note that we have to wrap this in some special category code changes (in a group) to get the parsing correct). We have two versions of the general font shape definition, one for `OT1` encoding and one for the `T1` encoding. We have another version for math families, one for substition and a special *odd* one because CMR is used in both math and text mode (sometimes). The last conditional test just makes it easier to implement the EC to DC font switch.

```
\begingroup
\nfss@catcodes
\providecommand{\flt@family}[5]{%
  \DeclareFontShape{#1}{#2}{#3}{#4}
  {<12><14.4><17.28><20.74><24.88><29.86>
   <35.83><43.00><51.60> #5 }{}}
\providecommand{\fltodd@family}[6]{%
  \DeclareFontShape{#1}{#2}{#3}{#4}
  {<12.1><14.5><17.38> #5
   <12><14.4><17.28><20.74><24.88><29.86><35.83><43.00><51.60> #6}{}}
\providecommand{\flt@subfamily}[5]{%
  \DeclareFontShape{#1}{#2}{#3}{#4}{<->ssub * #5}{}}
\providecommand{\fltEC@family}[5]{%
  \DeclareFontShape{#1}{#2}{#3}{#4}
  {<12><14.4><17.28><20.74><24.88><29.86>
   <35.83><43.00><51.60> genb * #5}{}}
\providecommand{\fltmath@family}[6]{%
  \DeclareFontShape{#1}{#2}{#3}{#4}
  {<12.1><14.5><17.38> #5
   <20.74><24.88><29.86><35.83><43.00><51.60> #6}{}}
\if@useDCfonts
\def\EC{dc}
\def\ECrm{dcr}
\def\ECrb{dcb}
\def\ECui{dcu}
\else
\def\EC{ec}
\def\ECrm{ecrm}
\def\ECrb{ecrb}
\def\ECui{ecui}
\fi
```

14.2 The font family and shape declarations: OT1

Mostly we just set up only large versions of the fonts. We had to do some non-obvious things here in the smaller versions. See Subsection 14.6 for more information (this will explain the 12.1, 14.5 and 17.38 sizes below instead of the expected 12.0, 14.4 and 17.28). We first load the font family and shape information for the standard Computer Modern text fonts. We start with Roman (family `fcmr`).

```
\DeclareFontFamily{OT1}{fcmr}{}
```

First we make a shape declaration for medium/normal. Note that we are a bit redundant here on the smaller point sizes. That is so that the math sizes and the scaled text sizes can work together without generating too many error messages. See also Subsection 14.6.

```
\fltodd@family{OT1}{fcmr}{m}{n}    {cmr7}{cmr10}
```

Then we make one for medium/italics. (Slanted is handled by a substitution given later to the default family `fcmss`.)

```
\flt@family{OT1}{fcmr}{m}{it}      {cmti10}
```

We really should have a small caps version of cmss, but no such font is default, so we leave it as is.

```
\flt@family{OT1}{fcmr}{m}{sc}      {cmcsc10}
```

Now, we add one for bold-extended.

```
\flt@family{OT1}{fcmr}{bx}{n}      {cmbx10}
```

Next, we add some for bold-extended in italics and slanted. We don't expect these to get called directly, but as substitution fonts for sans serif versions.

```
\flt@family{OT1}{fcmr}{bx}{sl}     {cmbxsl10}
```

As in the standard LaTeX, we use the upright shape font, essentially only for the `\pounds` symbol.

```
\flt@family{OT1}{fcmr}{m}{ui}      {cmu10}
% Finally, we fill in a few missing gaps with substition fonts.
\flt@subfamily{OT1}{fcmr}{m}{sl}   {fcmss/m/sl}
\flt@subfamily{OT1}{fcmr}{b}{n}    {fcmr/bx/n}
\flt@subfamily{OT1}{fcmr}{bx}{it}  {fcmr/bx/sl}
```

Now we set up the essential stuff for the default family, San Serif (`fcmss`). Please consult pages 201–202 of [1] for more information about the use of `\fontdimen` stuff here. Essentially, we are allowing more inter-word space stretching, to compensate for the lack of hyphenation.

```
\DeclareFontFamily{OT1}{fcmss}{\fontdimen3\font=1.7\fontdimen3\font}
```

First we make a shape declaration for medium/normal.

```
\flt@family{OT1}{fcmss}{m}{n}      {cmss10}
```

Here is the slanted version.

```
\flt@family{OT1}{fcmss}{m}{sl}     {cmssi10}
```

I don't know why this is here, but it is analogous to what is in the standard sets so we leave it in, just rescale the sizes. This is semi-bold condensed (or is that demi-bold condensed?).

```
\flt@family{OT1}{fcmss}{sbc}{n}    {cmssdc10}
```

Finally, we get to the bold versions. There really is a bold extended sans serif so we point off to that.

```
\flt@family{OT1}{fcmss}{bx}{n}     {cmssbx10}
```

Here are all the substitution fonts. Sans serif italics is not a standard font so we substitute roman italics. That's OK, because we will use slanted for emphasize anyway and I think user's expect roman italics.

```
\flt@subfamily{OT1}{fcmss}{m}{it} {fcmr/m/it}
```

For small-caps, we substitute the roman small caps, which is what I think most users will expect and because we don't really have a small caps, sans serif.

```
\flt@subfamily{OT1}{fcmss}{m}{sc} {fcmr/m/sc}
```

We again use the roman upright for the `\pounds` symbol.

```
\flt@subfamily{OT1}{fcmss}{m}{ui} {fcmr/m/ui}
```

There is no plain bold, consequently, we substitute the bold extended. Finally, if someone asks for slanted or italics in bold extended, since we don't have either of these we substitute the roman counter parts. It seemed that if the user was in either bold extended (or italics) and wanted to italicize (or embolden), then they should see some change. From above, the `fcmr/bx/it` is replaced by `fcmr/bx/sl` so this becomes our complete substitute for bold italics sans serif. We do these latter substitutions with a warning, so the user knows what happens and it doesn't come as a complete surprise.

```
\flt@subfamily{OT1}{fcmss}{b}{n}  {fcmss/bx/n}
\flt@subfamily{OT1}{fcmss}{bx}{sl}{fcmr/bx/sl}
\flt@subfamily{OT1}{fcmss}{bx}{it}{fcmr/bx/it}
```

Next we add the typewriter font (`fcmtt`). This is short and sweet. We set up for medium normal and slanted and everything else is a substitution.

```
\DeclareFontFamily{OT1}{fcmtt}{\hyphenchar\font\m@ne}
\flt@family{OT1}{fcmtt}{m}{n}     {cmtt10}
\flt@family{OT1}{fcmtt}{m}{sl}    {cmsltt10}
\flt@subfamily{OT1}{fcmtt}{m}{it} {fcmtt/m/sl}
\flt@subfamily{OT1}{fcmtt}{bx}{n} {fcmtt/m/n}
\flt@subfamily{OT1}{fcmtt}{bx}{it}{fcmtt/m/it}
\flt@subfamily{OT1}{fcmtt}{bx}{sl}{fcmtt/m/sl}
```

14.3 The font family and shape declarations: T1

Now we do the same for the EC font encoding. This is a bit tricky since we want to support choices between large design sizes and scaled fonts and choices between EC and old DC fonts. This has not been extensively tested but it seems to work and is logically consistent with the stuff from the LaTeX distribution, up to some changes to be consistent with how FoilTEX dealt with the old versions of DC fonts. Where different, I've left LaTeX2$_\varepsilon$'s model as a comment. Feel free to switch if you want. We make no additional comments throughout the rest of these part.

```
\DeclareFontFamily{T1}{fcmr}{}
\if@magscaleECfonts
 \flt@family{T1}{fcmr}{m}{n}      {\ECrm 1000}
 \flt@family{T1}{fcmr}{m}{it}     {\EC ti1000}
 \flt@family{T1}{fcmr}{m}{sc}     {\EC cc1000}
 \flt@family{T1}{fcmr}{bx}{n}     {\EC bx1000}
 \flt@family{T1}{fcmr}{bx}{sl}    {\EC bl1000}
 \flt@family{T1}{fcmr}{m}{ui}     {\ECui 1000}
\else
 \fltEC@family{T1}{fcmr}{m}{n}    {\ECrm}
```

```
 \fltEC@family{T1}{fcmr}{m}{it}   {\EC ti}
 \fltEC@family{T1}{fcmr}{m}{sc}   {\EC cc}
 \fltEC@family{T1}{fcmr}{bx}{n}   {\EC bx}
 \fltEC@family{T1}{fcmr}{bx}{sl}  {\EC bl}
 \fltEC@family{T1}{fcmr}{m}{ui}   {\ECui}
\fi
\flt@subfamily{T1}{fcmr}{m}{sl}   {fcmss/m/sl}
%\if@magscaleECfonts
% \flt@family{T1}{fcmr}{m}{sl}    {\EC sl1000}
%\else
% \fltEC@family{T1}{fcmr}{m}{sl}  {\EC sl}
%\fi
\flt@subfamily{T1}{fcmr}{b}{n}    {fcmr/bx/n}
\flt@subfamily{T1}{fcmr}{bx}{it}  {fcmr/bx/sl}
\DeclareFontFamily{T1}{fcmss}{\fontdimen3\font=1.7\fontdimen3\font}
\if@magscaleECfonts
 \flt@family{T1}{fcmss}{m}{n}     {\EC ss1000}
 \flt@family{T1}{fcmss}{m}{sl}    {\EC si1000}
 \flt@family{T1}{fcmss}{bx}{n}    {\EC sx1000}
\else
 \fltEC@family{T1}{fcmss}{m}{n}   {\EC ss}
 \fltEC@family{T1}{fcmss}{m}{sl}  {\EC si}
 \fltEC@family{T1}{fcmss}{bx}{n}  {\EC sx}
\fi
\flt@family{T1}{fcmss}{sbc}{n}    {\EC ssdc10}
\flt@subfamily{T1}{fcmss}{m}{it}  {fcmr/m/it}
%\if@magscaleECfonts
% \flt@family{T1}{fcmss}{m}{it}   {\EC si1000}
%\else
% \fltEC@family{T1}{fcmss}{m}{it} {\EC si}
%\fi
\flt@subfamily{T1}{fcmss}{m}{sc}  {fcmr/m/sc}
\flt@subfamily{T1}{fcmss}{m}{ui}  {fcmr/m/ui}
\flt@subfamily{T1}{fcmss}{b}{n}   {fcmss/bx/n}
\flt@subfamily{T1}{fcmss}{bx}{it} {fcmr/bx/it}
%\if@magscaleECfonts
% \flt@family{T1}{fcmss}{bx}{it}  {\EC so1000}
%\else
% \fltEC@family{T1}{fcmss}{bx}{it}{\EC so}
%\fi
\flt@subfamily{T1}{fcmss}{bx}{sl} {fcmr/bx/sl}
%\if@magscaleECfonts
% \flt@family{T1}{fcmss}{bx}{sl}  {\EC so1000}
%\else
% \fltEC@family{T1}{fcmss}{bx}{sl}{\EC so}
%\fi
\DeclareFontFamily{T1}{fcmtt}{\hyphenchar\font\m@ne}
\if@magscaleECfonts
 \flt@family{T1}{fcmtt}{m}{n}     {\EC tt1000}
 \flt@family{T1}{fcmtt}{m}{sl}    {\EC st1000}
\else
 \fltEC@family{T1}{fcmtt}{m}{n}   {\EC tt}
 \fltEC@family{T1}{fcmtt}{m}{sl}  {\EC st}
\fi
```

```
\flt@subfamily{T1}{fcmtt}{m}{it}  {fcmtt/m/sl}
%\if@magscaleECfonts
% \flt@family{T1}{fcmtt}{m}{it}   {\EC it1000}
%\else
% \fltEC@family{T1}{fcmtt}{m}{it} {\EC it}
%\fi
\flt@subfamily{T1}{fcmtt}{bx}{n}  {fcmtt/m/n}
\flt@subfamily{T1}{fcmtt}{bx}{it} {fcmtt/m/it}
\flt@subfamily{T1}{fcmtt}{bx}{sl} {fcmtt/m/sl}
% %%%%%% DO WE NEED TO DO THE TS1 COMPANION FONTS???? %%%%%%%%%%%%
```

14.4 The font family and shape declarations: OM*

Of course, we still haven't dealt directly with any of the math fonts. We start with math italics. Note again the strange point sizes in the smaller fonts.

```
\DeclareFontFamily{OML}{fcmm}{\skewchar\font'177}
\fltmath@family{OML}{fcmm}{m}{it} {cmmi7}{cmmi10}
\fltmath@family{OML}{fcmm}{b}{it} {cmmib7}{cmmib10}
\flt@subfamily{OML}{fcmm}{bx}{it} {fcmm/b/it}
```

Next we add the standard symbol fonts in normal and bold.

```
\DeclareFontFamily{OMS}{fcmsy}{\skewchar\font'60}
\fltmath@family{OMS}{fcmsy}{m}{n} {cmsy7}{cmsy10}
\fltmath@family{OMS}{fcmsy}{b}{n} {cmbsy7}{cmbsy10}
\flt@subfamily{OMS}{fcmsy}{bx}{n} {fcmsy/b/n}
```

The December 1994 release added stuff for cmr math symbols, so we have to do the same here to avoid warning messages. We add a few more than what is in the standard LaTeX because we don't like these warning messages.

```
\DeclareFontFamily{OML}{fcmss}{\skewchar\font'177}
\flt@subfamily{OML}{fcmss}{m}{n}  {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{m}{it} {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{m}{sl} {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{m}{sc} {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{bx}{n} {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{b}{n}  {fcmm/m/it}
\flt@subfamily{OML}{fcmss}{bx}{sl}{fcmm/m/it}
\flt@subfamily{OML}{fcmss}{bx}{it}{fcmm/m/it}
\DeclareFontFamily{OMS}{fcmss}{\skewchar\font'60}
\flt@subfamily{OMS}{fcmss}{m}{n}  {fcmsy/m/n}
\flt@subfamily{OMS}{fcmss}{m}{it} {fcmsy/m/n}
\flt@subfamily{OMS}{fcmss}{m}{sl} {fcmsy/m/n}
\flt@subfamily{OMS}{fcmss}{m}{sc} {fcmsy/m/n}
\flt@subfamily{OMS}{fcmss}{bx}{n} {fcmsy/b/n}
\flt@subfamily{OMS}{fcmss}{b}{n}  {fcmsy/b/n}
\flt@subfamily{OMS}{fcmss}{bx}{sl}{fcmsy/b/n}
\flt@subfamily{OMS}{fcmss}{bx}{it}{fcmsy/b/n}
\DeclareFontFamily{OML}{fcmr}{\skewchar\font'177}
\flt@subfamily{OML}{fcmr}{m}{n}   {fcmm/m/it}
\DeclareFontFamily{OML}{fcmtt}{\skewchar\font'177}
\flt@subfamily{OML}{fcmtt}{m}{n}  {fcmm/m/it}
\DeclareFontFamily{OMS}{fcmr}{\skewchar\font'60}
\flt@subfamily{OMS}{fcmr}{m}{n}   {fcmsy/m/n}
\DeclareFontFamily{OMS}{fcmtt}{\skewchar\font'60}
```

```
\flt@subfamily{OMS}{fcmtt}{m}{n}  {fcmsy/m/n}
```

Don't forget the extended symbol font for delimiters and the like.

```
\DeclareFontFamily{OMX}{fcmex}{}{}
 \fltmath@family{OMX}{fcmex}{m}{n} {cmex7}{cmex10}
```

We were going to add one for a new FoilTEX-LATEX symbol font (`flasy`) but decided that was a bit much. Instead, we just set up the `lasy` font family and modify its parameters here. In this way, if the user specifies `\usepackage{latexsym}` the default Ulasy.fd is *not* loaded and the sizes don't get overrun. Also, we avoid having to expressly define all the required symbols. We do define a symbol font (`flasy`) however so we can use it for the closing box at the end of proofs.

```
\DeclareFontFamily{U}{lasy}{}
 \fltmath@family{U}{lasy}{m}{n}    {lasy7}{lasy10}
 \flt@family{U}{lasy}{b}{n}        {lasyb10}
\endgroup  % end of nfss@catcodes group
\DeclareSymbolFont{flasy}{U}{lasy}{m}{n}
```

14.5 The font selection mechanisms

We've got all the font family and shape declaration tables set up. Now we configure the font selection mechanism for FoilTEX. First we define the defaults.

```
\def\rmdefault{fcmr}
\def\sfdefault{fcmss}
\def\ttdefault{fcmtt}
\def\itdefault{it}
\def\sldefault{sl}
\def\bfdefault{bx}
```

This next definition finally sets up the sans serif as the default family.

```
\renewcommand\familydefault{\sfdefault}
```

Next we set up the old font changing commands and the ones used in math (except small caps and slanted which are prohibited in math). Notice here that (in contrast to SLITEX) the `\rm` is still roman! Also, we replace the `\em` declaration so that it switches between the upright shape and the *slanted* shape.

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand\em{\@nomath\em \ifdim \fontdimen\@ne\font >\z@
     \upshape \else \slshape \fi}
```

14.6 Math and symbol fonts

We need to replace all the standard symbol, operator, letter and large symbol fonts with our new font families as declared above in Subsection 14.4. We do this for both normal and bold equations.

```
\SetSymbolFont{operators}{normal}{OT1}{fcmr}{m}{n}
\SetSymbolFont{letters}{normal}{OML}{fcmm}{m}{it}
\SetSymbolFont{symbols}{normal}{OMS}{fcmsy}{m}{n}
```

```
\SetSymbolFont{largesymbols}{normal}{OMX}{fcmex}{m}{n}
\SetSymbolFont{operators}{bold}{OT1}{fcmr}{bx}{n}
\SetSymbolFont{letters}{bold}{OML}{fcmm}{b}{it}
\SetSymbolFont{symbols}{bold}{OMS}{fcmsy}{b}{n}
\SetSymbolFont{largesymbols}{bold}{OMX}{fcmex}{m}{n}
```

We also need to set up the math alphabets for the font change commands and declarations in math mode. Note that we expressly set the math fonts to roman unless expressly asked for with `\mathsf` or `\mathtt`. All the others (`\mathbf` and `\mathit`, along with `\mathrm`) yield the roman (serif) counterpart. This is in constrast to what we do in text.

```
\DeclareMathAlphabet{\mathrm}{OT1}{fcmr}{m}{n}
\DeclareMathAlphabet{\mathbf}{OT1}{fcmr}{bx}{n}
\DeclareMathAlphabet{\mathsf}{OT1}{fcmss}{m}{n}
\DeclareMathAlphabet{\mathit}{OT1}{fcmr}{m}{it}
\DeclareMathAlphabet{\mathtt}{OT1}{fcmtt}{m}{n}
```

Here are a few commands for caligraphic and math italics fonts:

```
\DeclareRobustCommand*\cal{\@fontswitch{\relax}{\mathcal}}
\DeclareRobustCommand*\mit{\@fontswitch{\relax}{\mathnormal}}
```

Now we get to the nub of the subtle issue of font point sizes. The natural thing to want to do (and was easy to do in OFSS) was the following. First determine the font (with natural point size) you want, determine mag-scaling needed to get this to the approximate point size desired and hard-code this as exactly equal. So, we could say that a 12pt version of a 7pt font was just scaled by magstep 3. This was in spite of the fact that this was more likely to be a 12.1pt font. In NFSS, the desired point size and the design size are used to compute the scaling desired so a declaration like 12pt scaling of 7pt font would require a non-standard magnification (and so drivers would either have to generate a whole new library of fonts, or scale existing fonts or LaTeX would have to scream out error messages about font substitutions.

This could be avoided if we simply used scaled versions of just 10pt fonts, but we found in the earlier FoilTeX that it looked much better in math mode to try to maintain as much of the relative scaling of fonts as was in Knuth's original design, namely 10pt (design-size) font should have a 7pt (design-size) exponent and 5pt super-exponent. In FoilTeX, we compromise a little on this and try to force (as can be seen from the font shape declarations above) scaled versions of 7pt fonts for exponents and smaller scaled versions of 7pt for super-exponents. But of course, this means that the math size for this needs to reflect this as much as possible (hence the 12.1pt version of 7pt fonts). On the other hand, some text fonts that also play a role in math mode (like roman) need to have their corresponding 12pt text versions (in `\tiny`, for example). That's why some of these fonts have size information for 12.1pt (cmr7) and 12pt (cmr10).

Consequently, we declare some tokens for the small math sizes as 12.1, 14.5 and 17.38 and then set the math sizes for the various point sizes. We also stick to the principle that no font should be smaller than 12pt so this limits what we can do in exponents and the like. We add one for really big fonts but only if we are not running in compatibility mode. Somewhere at sometime I wanted these really big stuff and so here it is now.

```
\newcommand\@xii@ipt{12.1}
\newcommand\@xiv@vpt{14.5}
```

```
\newcommand\@xvii@iiipt{17.38}
\newcommand\@xxxpt{29.86}
\newcommand\@xxxvipt{35.83}
\newcommand\@xliiipt{43}
\newcommand\@lipt{51.60}
\DeclareMathSizes{\@xiipt}{\@xii@ipt}{\@xii@ipt}{\@xii@ipt}
\DeclareMathSizes{\@xivpt}{\@xiv@vpt}{\@xii@ipt}{\@xii@ipt}
\DeclareMathSizes{\@xviipt}{\@xvii@iiipt}{\@xii@ipt}{\@xii@ipt}
\DeclareMathSizes{\@xxpt}{\@xxpt}{\@xiv@vpt}{\@xii@ipt}
\DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xvii@iiipt}{\@xiv@vpt}
\DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxpt}{\@xvii@iiipt}
\DeclareMathSizes{\@xxxvipt}{\@xxxvipt}{\@xxvpt}{\@xxpt}
\DeclareMathSizes{\@xliiipt}{\@xliiipt}{\@xxxpt}{\@xxvpt}
\if@compatibility\else
  \DeclareMathSizes{\@lipt}{\@lipt}{\@xxxvipt}{\@xxxpt}
\fi
```

Finally, we have to deal with the picture environment fonts (circle and line). These seem to be broken no matter what I do. I liked the fatter lines that I got with the magscaled versions in the old FoilTEX, but it seemed to be very hard to convert user's pictures from LATEX to FoilTEX. So, in the new version we don't scale and in compatibility mode, we do scale. In either case, it's easy to generate pictures that just don't work!

```
\if@compatibility
  \font\tencirc=lcircle10 scaled \magstep4
  \font\tencircw=lcirclew10 scaled \magstep4
  \font\tenln=line10 scaled \magstep4
  \font\tenlnw=linew10 scaled \magstep4
\else
  \font\tencirc=lcircle10
  \font\tencircw=lcirclew10
  \font\tenln=line10
  \font\tenlnw=linew10
\fi
⟨/fonts⟩
```

References

[1] Michel Goossens, Frank Mittelbach and Alexander Samarin, *The LATEX Companion*, Addison-Wesley, Reading, Massachusetts, 1994.

[2] Donald Knuth, *The TEXbook*, Addison-Wesley, Reading, Massachusetts, 1983, revised in 1993.

[3] Leslie Lamport, *LATEX: A Document Preparation System*, 2nd ed., Addison-Wesley, Reading, Massachusetts, 1994.