The FoilTEX class package*

Jim Hafner
IBM Research Division
Almaden Research Center, K53/802
650 Harry Road
San Jose, CA 95120-6099
408-927-1892
hafner@almaden.ibm.com

Printed on September 14, 1995

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 v2.0.1, last revised 1995/06/22.

Contents

This document describes the FoilTEX system (version 2.0.1) 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
`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 DC fonts. (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 AMS-TEX fonts. Finally, the "This is *not* LATEX" warning has been removed as it is completely irrelevant now.

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` 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.

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, legalpaper, 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` 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` In order to support rotated foils and `landscape` in a clean way, we need to issue `\special`s which are driver dependent. Since we only had access to one PostScript driver, namely, Rokicki's `dvips`, we could only support these `\special`s for this one driver. Declaring this option enables this special code. See also Section 9.

We note here that the `landscape` option essentially just adjusts the page dimensions. Typically, it does not force the driver to do the necessary rotation steps. As mentioned, in FoilTeX with the `dvips` 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 support a set of default options with FoilTEX simply by having in your inputs path a file called `foiltex.cfg` with a lines containing `\ExecuteOptions` statements. E.g., to configure for `dvips` and `landscape`, your `foiltex.cfg` file might look like

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

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

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 v2.0.1 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 v2.0.1) 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.

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}}
```

(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`.

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

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`
`\captionwidth`
`\abovetitleskip`
`\titleauthorskip`
`\authorauthorskip`
`\authordateskip`
`\dateabstractskip`
`\zerolistvertdimens`

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, we added a `\captionwidth` so that if the user prefers to have captions narrower than the surround text, that can be easily adjusted. The default for this is the `\textwidth`.

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
```

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.

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]`, or `[30pt]` 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.

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.

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.

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

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-

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

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

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

As experimental, research software, 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. Acceptance and use of this program constitutes the user's understanding that he 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. 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 EXPERIMENTAL PROGRAM THE USER AGREES TO THE FOLLOWING:

a. This program is provided for the user's personal, non-commercial, experimental use and 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 the EXPERIMENTAL SOFTWARE 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 EXPERIMENTAL SOFTWARE 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 are to be used solely for experimental uses and are not to be sold, distributed to a commercial organization, or be commercially exploited in any manner.

f. The user understands and agrees that this program, and any derivative works made from this program, are to be used solely for personal, non-commercial, experimental uses, as per section a., above, and that this program and any such derivative works 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.

g. 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 2.0.1 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$_\varepsilon$. Implementing much of FoilTeX was greatly eased by the way LaTeX 2$_\varepsilon$ is constructed and implemented.

Beyond this, the LaTeX 2$_\varepsilon$ 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.

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.

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.