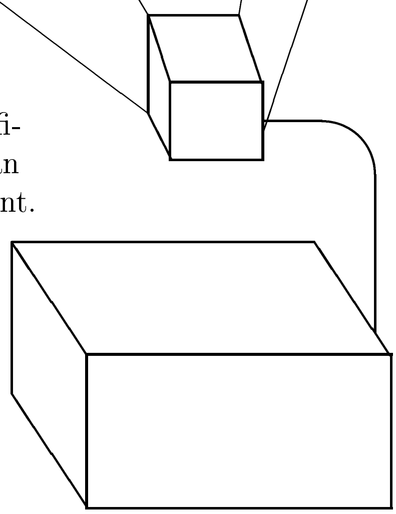
seminar.sty
A LaTeX style for slides and notes
User's Guide

Timothy Van Zandt
tvz@Princeton.EDU

1 April 1993
Version 1.0

seminar.sty is a LaTeX style for typesetting slides or transparencies, and accompanying notes. Here are some of its special features: It is compatible with AMS-LaTeX, and you can use PostScript and AMS fonts. Slides can be landscape and portrait. There is support for color and frames. The magnification can be changed easily. Overlays can be produced from a single slide environment. Accompanying notes, such as the text of a presentation, can be put outside the slide environments. The slides, notes or both together can then be typeset in a variety of formats.



Contents

	Getting Started	**ii**
I	**Just slides**	**1**
1	**Landscape and portrait slides**	1
2	**The height and width of slides**	1
3	**Margins within a slide**	2
4	**Page breaking within a slide environment**	3
5	**Margins on the page**	4
6	**Magnification and lengths**	5
7	**Font sizes**	7
8	**Spacing parameters**	8
9	**Slide frames**	10
II	**Advanced Features**	**12**
10	**Counters**	12
11	**Selectively including or excluding slides**	12
12	**Printing your slides two-up**	13
13	**Notes**	15
14	**How notes are omitted**	16
15	**The article format**	18
16	**Page styles**	19
17	**Other style options**	20
18	**Overlays**	20
III	**Help**	**24**
19	**Tips and tricks**	24

20 Errors 27

21 Troubleshooting 28

22 Converting SLITEX files 30

Appendices: Configuration 32

A The short story about fonts 32

B Choosing fonts 32

C Font bitmaps 33

D Color 35

E Landscape printing and slide rotation 35

F The semcolor style option 36

G A4 and other paper sizes 37

H Configuration file 38

Getting Started

`seminar.sty` is a LaTeX document style for typesetting slides, and more.

You should know how to use LaTeX, as described in Leslie Lamport's *LaTeX: User's Guide and Reference Manual.*

Let's get started:

1. If you are installing `seminar.sty`, read the the accompanying read-me file, and put the input files where your TeX looks for inputs.
2. Typeset the sample file, `semsamp1.tex`, to see that everything is working. You can use LaTeX or $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX.
3. Read Appendix A, on page 32.
4. To start making landscape slides, use

```
\documentstyle{seminar}
\begin{document}
\begin{slide}
  foo
\end{slide}
\end{document}
```

   and print out your document in landscape mode.

5. To start making portrait slides, include the `portrait` style option, and use the `slide*` environment instead of `slide`:

```
\documentstyle[portrait]{seminar}
\begin{document}
\begin{slide*}
  foo
\end{slide*}
\end{document}
```

6. If you have used SLiTeX, see Section 22.
7. For A4 paper, use the `a4` style option.
8. When you are ready to explore `seminar.sty`'s special features, skim the *User's Guide*, including the appendices.
9. Play around with the sample file `semsamp2.tex` to try out some of `seminar.sty`'s special features.
10. When you run into problems, look for help in Part III.

I Just slides

1 Landscape and portrait slides

`seminar.sty` is a LaTeX (or $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX) document style. Thus, begin your document with

```
\documentstyle{seminar}
```

The slide environments are

```
\begin{slide}     ...    \end{slide}
\begin{slide*}    ...    \end{slide*}
```

`slide` is for landscape slides and `slide*` is for portrait slides.

By default, the document is typeset in landscape mode, but if you include the `portrait` style option, the document is typeset in portrait mode. Typesetting the document in landscape mode is different from printing it in landscape mode; you may have to take care of the latter when printing with your dvi driver (see Appendix E).

If you have both landscape and portrait slides in your file, there are two ways to print the slides:

- If you dvi driver supports rotation, then you can print all the slides at once. See Appendix E for details.
- You can first print your landscape slides by putting the command

  ```
  \landscapeonly
  ```

  in the preamble, and then print the portrait slides by inserting instead the command

  ```
  \portraitonly
  ```

  and including the `portrait` style option.

2 The height and width of slides

The dimensions of the slides are set by the lengths

```
\slidewidth
\slideheight
```

The "width" refers to the width of a slide when looking at it in landscape orientation, whether it is a landscape or portrait slide (and the same goes for "height"). The default width is 8.5 inches and the default height is 6.3 inches.

The slide environments have an optional argument that lets you change the dimensions of a single slide, as in

```
\begin{slide*}[7.5in,6in]
```

The first dimension is the slide's "width", and the second dimension is the slides "height". If you remember what we said about what "width" and "height" mean, you will see that the above example begins a portrait slide that is 7.5 inches high and 6 inches wide.

3 Margins within a slide

When a slide is not full, the material is vertically centered within the slide. The command

```
\centerslidesfalse
```

cause the material to be flush to the top, instead. The command

```
\centerslidestrue
```

switches back to vertical centering.

The right margin in slides is ragged by default. You can change this with the command

```
\raggedslides[len]
```

len should be the maximum space between the end of the line and the right margin. The argument is optional; `\raggedslides` is equivalent to

```
\raggedslides[1fil]
```

which gives a ragged right margin, as in LaTeX's `flushleft` environment (the default). On the other hand,

```
\raggedslides[0pt]
```

gives a justified margin, and

```
\raggedslides[2em]
```

gives a semi-ragged margin. Note that as a margin becomes less ragged, hyphenation becomes more likely and more material fits on a slide.

4 Page breaking within a slide environment

A slide environment can contain more than one "page" of slides; TEX will break slides into pages automatically.

If the mere idea disturbs you, put the command

```
\extraslideheight{10in}
```

in your document. You can then divide slides into pages yourself by starting new slide environments or using the

```
\newslide
```

command within a slide environment. And you need read this section no further.

If instead you use the command

```
\extraslideheight{0pt}
```

then TEX will break pages the way you would expect. This is a quick and dirty way to break a whole paper or a long proof into slides.

However, dividing material into slides is usually too delicate a matter to be left up to TEX, and ultimately you will make all the page breaks yourself. On the other hand, automatic page breaking can still be helpful at the early stages, letting TEX find preliminary page breaks for you.

Setting the `\extraslideheight` to `0pt` doesn't give you much flexibility about where to put the page breaks. Of course, you can always put a page break earlier than the one found by TEX, but occasionally you will prefer to let a slide overflow by a small amount rather than rewrite the whole slide.

Therefore, by default, `seminar.sty` uses

```
\extraslideheight{10pt}
```

This adds on an extra `10pt` to the target slide height (as determined by `\slideheight` for landscape slides and `\slidewidth` for portrait slides) *for the purpose of page-breaking only.*

If the resulting slide exceeds the slide height, you will get a message like

```
LaTeX Error: Slide 3 overfull by 9pt.
```

`seminar.sty` then tries to reduce the slide to its maximum height by squeezing out rubber vertical space (e.g., tightening up the interline spacing). If the slide is still too full, you will get another warning like:

```
Overfull \vbox by 4.4pt while output was active.
```

You can go back and look for a way to make the slide shorter, or you can insert a `\newslide` command to change the page-break, or you can just ignore these warnings.

The recommended value to give in the argument of `\extraslideheight` is the largest length by which you might be willing to let a slide overflow.

After you have decided on the page breaks, you can shut up the warnings about overfull slides and `\vbox`'s with

```
\renewcommand{\slidefuzz}{1in}
```

The value of `\slidefuzz` (which should be length, even though `\slidefuzz` is an ordinary command sequence) is the threshold above which `seminar.sty` gives a warning about an overfull slide. The default definition is `2pt`.

5 Margins on the page

After `seminar.sty` makes a slide (or a slide page) as described above, it (optionally) frames the slide. This is described in Section 9. For making and framing the slide, `seminar.sty` does not need to know anything about the paper you are printing on, or the margins you want. Hence, none of LaTeX's standard page parameters are relevant within a slide environment. (However, for the sake of consistency, `seminar.sty` sets `\textwidth` to the width of the slide and `\textheight` to the height of the slide within a slide environment.)

Now `seminar.sty` has to do something with the finished slide. As you read this User's Guide, you will find that there are various options. However, right now we are making slides for printing on transparencies, and so we have to position the slide on the transparency and add headers and footers.

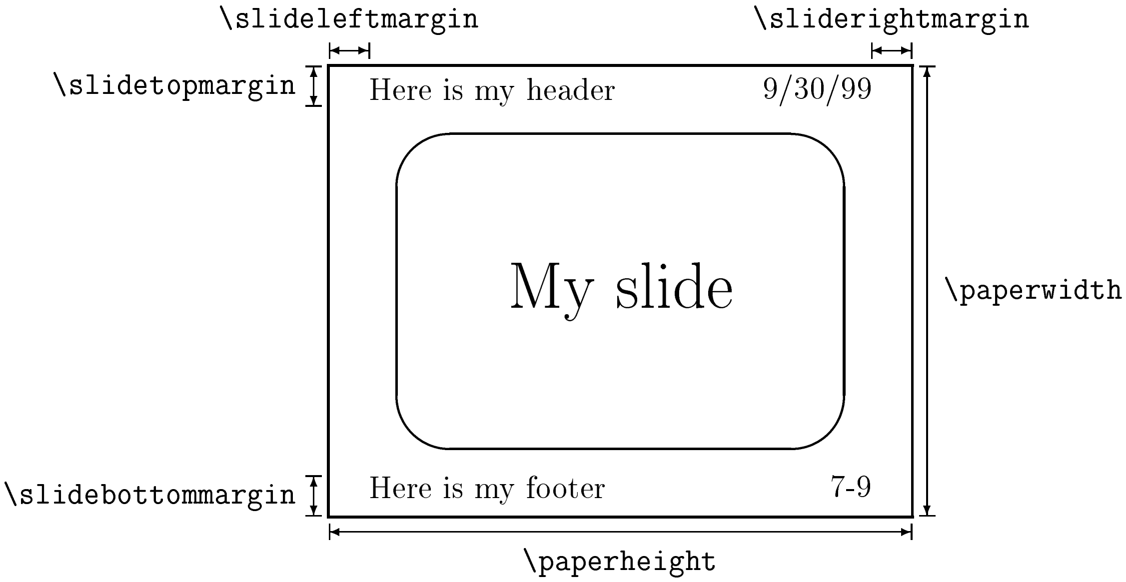


Figure 1: Slide margins.

`seminar.sty` does not use LaTeX's page parameters for this either. Instead, it uses the following parameters:

Parameter:	*Default:*
`\slideleftmargin`	.5in
`\sliderightmargin`	.5in
`\slidetopmargin`	.5in
`\slidebottommargin`	.5in

There are commands, to be set with `\renewcommand`.[1]

Look at Figure 1. Note that the headers and footers lie *inside* the top and bottom margins, respectively. The slide is then centered horizontally and vertically between the margins.

6 Magnification and lengths

`seminar.sty` changes TeX's magnification so that the output is larger than

[1] The only true length parameters (meaning that thay should be set with `\setlength`) in `seminar.sty` are `\slidewidth`, `\slideheight`, `\slideframewidth`, `\slideframesep`, `\semin`, and `\semcm`.

when typesetting an article. This means, for example, that if you paste some input from a paper you are writing with LaTeX into a slide, it will look pretty much the same in the slide, except that it is magnified. For example, `\vspace{.5in}` produces a space that gets bigger along with the fonts and everything else. Of course, it won't look identical, since `seminar.sty` uses its own spacing parameters and margins.

TeX's standard magnifications are in magsteps. n magsteps means a magnification of 1.2^n. `seminar.sty`'s default magnification is 4 magsteps, but you can change this with the command

`\slidesmag{`*n*`}`

n should be an integer between -5 and 9.

As noted above, lengths grow with the magnification. For spacing, like the parameter `\parindent` or using `\\[2pt]` to add a little extra space between lines, this is great, because it is easier to think in unmagnified dimensions. Setting `\parindent` to `.5cm` will look the same (relative to everything else) whatever the magnification.

However, if you want to set the unit in a `picture` environment to 1cm, *as it appears on the slide*, use

```
\setslidelength{\unitlength}{1cm}
```

`\setslidelength` is like LaTeX's `\setlength`, but it scales the size down so that the resulting size after magnification is, in this example, 1cm (in the process, it removes any stretch from rubber lengths).[2] There is also a

`\addtoslidelength{`*cmd*`}{`*len*`}`

command.

`seminar.sty` also provides the lengths

```
\semin
\semcm
```

which are equal to an inch and a centimeter, scaled down to "before magnification" size. For example,

```
\rule{1pt}{4\semcm}
```

[2]If you know what `true` dimensions are, you may be wondering why they haven't been mentioned. These are not recommended because you will not be able to print the slides two-up or use the `article` format.

makes a line 4 centimeters long on the transparency,.[3]

`\textwidth`, `\textheight`, `\columnwidth` and `\linewidth` all have the expected values in slides. (But `\slidewidth` and `\slideheight` do not.) Thus, this would

```
\rule{1pt}{.5\textwidth}
```

make a line that is 1/2 the width of the slide.

For help getting an EPS file to be the right size, see page 24.

Finally, there are some parameters that `seminar.sty` scales for you, and hence you can (and should) set them at their magnified values (the actual size on transparencies):

- `\slidewidth` and `\slideheight`
- `\slideframewidth` and `\slideframesep` (see Section 9).
- The slide margin parameters (see Section 5).
- The LaTeX page parameters, *when they are set in the preamble.*[4]

Note that the warnings about overfull slides (see Section 4) report unmagnified dimensions.

7 Font sizes

The `\slidesmag` command lets you change the magnification of your document. You can effectively fine tune the magnification by using the `11pt` and `12pt` style options. For example, instead of increasing the magnification from 4 to 5, you can switch to the `12pt` style option. There are differences—e.g., magnification affects the space you get from `\hspace{1in}`—but the

[3]Suppose you want to use another unit, such as millimeters, instead of centimeters. Then try this:

```
\newcommand{\semmm}{\semcm}
\renewcommand{\semcmlength}{1mm}
\setslidelength{\semmm}{1mm}
```

[4]The LaTeX page parameters are listed on page 163 of Lamport's *LaTeX User's Guide and Reference Manual.* They are used for notes (see Section 13) and the `article` option (see Sections 12 and 15), but not for printing slides onto transparencies. Lamport warns that resetting page parameters in the middle of the document is tricky; here it is more so, because you do have to worry about scaling them. But you can use `\setslidelength` for this.

Command:	*Initializes:*	*Default*
`\slideparskip`	`\parskip`	`1ex minus .2ex`
`\slideparindent`	`\parindent`	`0pt`
`\slidefootnotesep`	`\footnotesep`	`1.2ex`
`\slideleftmargini`	`\leftmargini`	`1.8em`
`\slideleftmarginii`	`\leftmarginii`	`1.4em`
`\slideleftmarginiii`	`\leftmarginiii`	`1em`
`\slidelabelsep`	`\labelsep`	`.5em`
`\slideitemsep`	`\itemsep`	`.8ex minus .2ex`
`\slidepartopsep`	`\partopsep`	`1ex minus .2ex`
`\slidestretch`	`\baselinestretch`	`1.2`
`\slidearraystretch`	`\arraystretch`	`1.2`

Figure 2: Slide spacing parameters.

fonts will at least be roughly the same size either way. The `11pt` option gives a half-step, somewhat like `\slidesmag{4.5}`, if the latter were permitted.

Sometimes you want to use smaller font sizes for a single slide, in order to fit in that little bit of extra material, or to use larger font sizes, so that the few things you have to say on a slide don't look too lonely. You can't changing the magnification in the middle of the document, but you can change the size of the fonts with the command

`\ptsize{`*n*`}`

This switches to the font sizes that are in effect when you use the *n*`pt` option. Actually, *n* can be 8, 9, 10, 11, 12, 14 or 17, whether you have `art`*n*`.sty` or not.

8 Spacing parameters

The commands in Figure 2 are used to initialize some of the spacing parameters at the beginning of each slide environment and when `\ptsize` is used in a slide environment. These are spacing parameters that should depend on the size of fonts and that in LaTeX's `article` style would be set in `art10.sty`, etc. These commands should all be changed with `\renewcommand`, even though, except for the `stretch` parameters, their values should be lengths. Note that `ex` and `em` units are used because these are the units that depend of font sizes.

Outside a slide environment, redefine the commands on the left if you want to change these spacing parameters. Within a slide environment, reset the parameters directly, or redefine the commands on the left and then use the `\ptsize` command.

In other LaTeX styles, the extra distance between lines that is inserted when `\baselinestretch` exceeds 1 is eaten up by tall or deep lines. E.g., if the line contains a table or a large math operator, there is probably no extra space at all. The advantage of this system is that the distance between baselines does not fluctuate with every tilde. The disadvantage is that lines can end up too close.

In `seminar.sty`, on the other hand, fraction

```
\slideskip
```

of the extra space cannot be eaten, but up to fraction

```
\slideshrink
```

of this extra space that cannot be eaten can be removed if there is too much material on the slide. This gives you added flexibility about how much material to include on a slide. Both `\slideskip` and `\slideshrink` can be set with `\renewcommand` to a number between 0 and 1. The default value of `\slideskip` is `.75` and the default value of `\slideshrink` is `.25`. Set `\slideskip` to `0` to revert to LaTeX's usual behavior, as described above.

For example, suppose you are using a 10pt font, `\slidestretch` is `1.2`, `\slideskip` is `.75`, and `\slideshrink` is `.25`. The `\baselineskip` for a 10pt font is normally 12pt, leaving a little space between the lines. Then `\baselineskip` is set to

$$\texttt{\textbackslash slidestretch} \times \texttt{12pt} = 1.2{\times}12\text{pt} = 14.4\text{pt}$$

The extra space between lines that is inserted is thus 2.4pt, and

$$\texttt{\textbackslash slideskip} \times 2.4\text{pt} = .75{\times}2.4\text{pt} = 1.8\text{pt}$$

of this cannot be eaten by tall or deep lines. However, the space between lines can be reduced by up to

$$\texttt{\textbackslash slideshrink} \times 1.8\text{pt} = .25{\times}1.8\text{pt} = .45\text{pt}$$

if the slide would otherwise be too long.

9 Slide frames

Slides can be framed. The command

`\slideframe[`*commands*`]{`*style*`}`

specifies the frame style to use. Valid frame styles are `none` and `plain`, unless you use macros that define additional styles. For example, the `fancybox` style option defines the frames `shadow`, `double`, `oval` and `Oval` (corresponding to the `\shadowbox`, `\doublebox`, `\ovalbox` and `\Ovalbox` commands defined in that style option). The `semcolor` option defines the styles `scplain`, `scdouble`, and `scshadow`.

All the frame styles use the lengths

```
\slideframewidth
\slideframesep
```

which are the (magnified) width of the line (default 4pt) and the distance between the slide and the frame (default .4in), respectively.

`\slideframe`'s optional argument is for commands that you want to use to customize the slide frame style. For example:

```
\slideframe[\setlength{shadowsize}{12pt}]{shadow}\\
\slideframe[\psset{fillstyle=gradient}]{scplain}
```

If you want to build your own custom slide frame, use the command

`\newslideframe{`*style*`}[`*commands*`]{`*frame command*`}`

style is the name of the frame, `[`*commands*`]`, which is optional, will be inserted before the frame command and before the *commands* given by `\slideframe`'s optional argument. These commands can be used to set some default parameter values. Then the final argument should frame `#1`.

For example, if `\myframe{foo}` frames `foo`, and if `\myframe` uses the length `\baldness` as a parameter, then you might write

```
\newslideframe{wildframe}[\setlength{\baldness}{.2cm}]%
  {\myframe{#1}}
```

You can still override the default value of `\baldness`, as in

```
\slideframe[\setlength{\baldness}{.1cm}]{wildframe}
```

There is a starred version of `\slideframe` that adds the frame to previously specified frames.[5] This can be used for special tricks. For example, if you are using the `fancybox` and `semcolor` options, then

```
\newslideframe{draft}%
  {\boxput{\rput{30}{\Huge\gray DRAFT}}{#1}}
\slideframe{draft}
\slideframe*{scdouble}
```

puts the word "DRAFT" in the background of each slide, gray and rotated 30 degrees, and then adds a double frame.[6]

[5]If using color or PostScript images, note that each frame gets added to the background.
[6]Use `\boxput*` to put `DRAFT` in the foreground. See `fancybox.doc` for details.

II Advanced Features

10 Counters

The counter for slides is `slide`. The default definition of `\theslide` is `\arabic{slide}`. You can use `\label` and `\ref` to cross-reference slides, and the page number in slide environments is set to `\theslide`.[7] Thus, you can write, for example,

```
See equation (\ref{foo}) on Slide \pageref{foo}.
```

You may want some counters, such as equation counters, to be reset with each new slide environment or `\newslide` command. By default, only the `footnote` counter is reset this way, but you can specify your own (comma-separated) list of counters to be reset with the command

`\slidereset{`*list*`}`

If you want to reset additional counters, rather than replace the list entirely, use

`\addtoslidereset{`*list*`}`

There is another way in which the footnote counter gets special treatment. The command

```
\theslidefootnote
```

is used for the counter text instead of `\thefootnote`. The default definition is

```
\alph{footnote}
```

11 Selectively including or excluding slides

The commands

`\onlyslides{`*list*`}`
`\notslides{`*list*`}`

[7]However, these page or slide cross-references are not always accurate when you let TeX break pages for you within a slide.

can be used to include or exclude only those slides in the given list. The argument should expand to a comma-separated list of numbers or ranges. The numbers do not need to be in order, the list can contain numbers that do not correspond to any slide, and there can be duplicate numbers. Negative numbers should be enclosed in curly braces. Since the argument is first expanded, you can use the `\ref` command in the argument. For example,

```
\onlyslides{{-2},\ref{dp}-10,\ref{chart},0,17-999}
```

is legal. If `\label{dp}` appears in slide 5 and `\label{chart}` appears in slide 12, this is equivalent to:

```
\onlyslides{{-2},5-10,12,0,17-999}
```

12 Printing your slides two-up

Printing your slides two-up is useful both for previewing slides and for making hard copies to distribute or for proofreading.

One way to print your slides two-up is to include the `article` style option. This is described further below.

Another way is to include the command

`\twoup[`*n*`]`

in the preamble. This inputs `2up.tex`, which contains generic macros for two-up printing, and sets the parameters to values that are likely to work. Including the optional argument `[`*n*`]` increases the two-up magnification by n magsteps. n can be a positive or negative integer. If you are not able to get the right layout, then include the `2up` style option, read the documentation of `2up.tex`, and set the parameters yourself, rather than using the `\twoup` command.

The `article` style option is also called the article *format* (as opposed to the slides format). In the article format without the `portrait` option, the slides are centered horizontally and vertically, two to a page (if they fit—landscape slides do fit by default). With the portrait option, the slides are printed side-by-side, two to a page. In the `article` format, you can mix landscape and portrait slides with different orientations, but this does not work well with the `portrait` option.

You can change the article format's magnification with

`\articlemag{`*n*`}`

This command works like the `\slidesmag` command (page 6). The default is

```
\articlemag{0}
```

Increase the article magnification if the slides look too lonely; decrease it if they are not coming out two-up.

The size of the slides depends on the *difference* between the magnifications in the slides and article format. E.g., since the default slides magnification is 4 magsteps, the slides are scaled down in the article format by 4 magsteps. When you change the slides magnification with `\slidesmag`, you also have to change the article magnification with `\articlemag` by the same amount if you want the size of the slides in the article format to remain the same. However, when the article magnification in magsteps is negative , you will be using non-standard font magnifications (which will produce terrible results if you cannot generate the needed fonts—see Appendix C for advice).

Whether or not landscape slides (or rotated portrait slides) come out two-up in the article format without the `portrait` option depends on (i) the difference between the slides and article magnification, (ii) the values of `\slidewidth` and `\slideheight` (iii) the size of the slide frame, and (iv) the article format's page parameters. With the default values of (ii)–(iv), the slides will still come out two-up if you change the difference between the slides and article magnification to 3 (e.g., increase the value of `\articlemag` by 1). Try this if you want the slides to be larger.

If you use the `article` option *and* the `\twoup` command, then your slides should be printed four-up!

You will notice labels on the side or bottom of each slide in the `article` format. The command

`\slidestyle{`*style*`}`

determines where these labels go. There are three predefined slide styles:

`empty` No captions or labels are used.

`left` The labels go on the left of each slide (the default when the `portrait` option is not used).

`bottom` The labels go on the bottom (the default with the `portrait` option).

The label you get is the value of `\slidelabel`. The default definition of `\slidelabel` is

```
\bf Slide \theslide
```

13 Notes

In addition to slides, you can include a few comments following each slide to remind you of what to say, or even the entire text of your presentation for your own benefit or to be distributed to others. These comments or text, referred to in this documentation as "notes", do not go inside any special environment. However, a slide cannot go in the middle of a paragraph of notes.

To include notes this way, you have to use one of the following style options, which determine what is printed:

`slidesonly`	Only the slides are printed.
`notes`	Both notes and slides are printed.
`notesonly`	Only the notes are printed.

These style options are referred to as *selections* in this documentation.

The pages of notes following slide 5 are numbered 5.1, 5.2, etc.

You can use the `portrait` option to print out your notes, even if you don't use this option when printing the slides.

With the `notesonly` selection, the slides are not printed, but they are processed. This means that one can still refer in the notes to slides or to equations in slides.

The fact that various spacing parameters are initialized at the beginning of each slide (see Section 8) means that when you reset these parameters outside a slide environment, only the notes are affected. This lets you use different values for notes and slides. The command `\slidefonts` can be used for any special font commands that should apply only to slides. Also, the command `\everyslide` is executed at the beginning of every slide, and you can use this for any other customizations you want to include.

You can also use the `11pt` or `12pt` style option for your notes, without changing the size of the fonts in the slides, by using the `\ptsize` command to set the font sizes for slides. For example, suppose that the `12pt` document style option is used, but the preamble contains the command

```
\ptsize{11}
```

and the second slide begins with the command

```
\ptsize{9}
```

Then the notes use the `12pt` font size declarations, all the slides but the second one use the `11pt` font size declarations, and the second slide uses the `9pt` font size declarations.

Notes are typeset using LaTeX's standard output routines, and LaTeX's standard page parameters.

Finally, it is even possible to use a different page style for notes and slides. The page style for notes is set by the `\pagestyle` and `\thispagestyle` commands. The command

`\slidepagestyle{`*style*`}`

sets a different page style for the slides. If the argument is empty, then the page style for slides reverts to the one for notes.[8]

With the command

```
\onlynotestoo
```

the `\onlyslides` and `\notslides` commands apply to the notes as well.

14 How notes are omitted

In the previous section, it was stated that notes are not enclosed in a special environment. `seminar.sty` omits notes using special macros[9] that comment out everything *outside* the slide environments (and a few other environments mentioned below). Global declarations that should be processed even when slides are omitted go in `allversions*` environments, described below.

`seminar.sty` also let you use a more conventional approach to notes. If you put the command

```
\noxcomment
```

in the preamble, then the notes go inside `note` environments, and you can put all the global declarations outside any environment. `xcomment.sty` is used to make `note` a comment environment when note should be omitted, and otherwise the `note` environment does nothing. This mode of operation is more robust than omitting everything outside the slide environments, but remembering to insert the

```
\begin{note} ... \end{note}
```

[8] There is no `\thisslidepagestyle` command.

[9] Defined in `xcomment.sty`

is more tedious.

Here are a few technicalities that have to do with `xcomment.sty` and that apply whether or not you use `\noxcomment`.

1. The text that follows the beginning of a slide environment (when not using `\noxcomment`) or the end of a `note` environment *must have balanced curly braces.*

2. `\input` and `\include` commands are followed, even when found in omitted notes, but you must use the LaTeX syntax `\input{`*file*`}` (as opposed to `\input␣`*file*`␣`), and the inputted file must end with `\endinput` (this is a good practice anyway).

3. In omitted text, `%` is still a comment character (hence it is possible to comment out a slide or note environment).

4. A temporary file, `\jobname.tmp`, is created (this is of no consequence—just in case you wanted to know where it comes from).

The rest of this section deals with special considerations when omitting everything outside the slide environments.

If you might want to make a global change to one of the slide parameters after the document preamble, you cannot include it in a slide environment because the change will be local, and you cannot include it in the notes because the change won't be processed when the notes are omitted.

To get around this problem, an environment `allversions*` is provided. It is processed even when notes are omitted, and any parameter changes or command definitions made within the environment are global. Don't generate any output within this environment.

On a rare occasion you might want to include some text that should be typeset even when the other notes are omitted (e.g., a list of the slides, or a cover page). The `allversions` environment is provided for this purpose.

If you do any serious hacking, you might want to add to the list of environments that should be included with the `slidesonly` selection. Do this with the command

`\addtoslidelist{`*list*`}`

where *list* is a comma separated list of environments, without spaces.

15 The article format

The `article` style option was described in Section 12 as a way to print your slides two-up. This option is also a good way to print your notes.

The `article` option gets its name because your document is typeset somewhat like it would be with LaTeX's `article` style. (It uses LaTeX's standard page parameters, but some of the default values are different.) It is also called the article *format*; the slides format is what you get without the `article` option.

You probably don't want to use the `portrait` style option with the `article` format except when you are only printing portrait slides. But you are welcome to experiment.

With the `notesonly` selection, you can make the notes as big as you want using the `\articlemag` command (e.g., for easy reading while giving a presentation). Just as in the `slides` format, the page parameters are scaled at the beginning of the document so that they can be set with their true dimensions in the preamble.

The commands

`\setartlength{`*cmd*`}{`*len*`}`
`\addtoartlength{`*cmd*`}{`*len*`}`

are analogous to `\setslidelength` and `\addtoslidelength`, but they scale the lengths so that they end up at the specified size after magnification in the `article` format. You never would use such a command in slide environments or to set slide parameters, but you might use these in the notes if you were planning on typesetting your notes with the `article` format rather than the `slides` format.

You can change the placement of slides in the `article` format using the

`\slideplacement{`*name*`}`

command. Here are the valid placement names:

float The slides are floated. This is the default with the `notes` selection and the `portrait` option.

float* Like `float`, but if the notes are printed in a two-column format the slides extend across both columns (e.g., like `table` versus `table*`). This is the default with the `notes` selection without the `portrait` option.

onepercol Each slide is centered horizontally and vertically within a single column. This is the default when only slides are printed and the `portrait` option is used.

twopercol Each slide is centered horizontally, two or one per column, depending on how many fit. This is the default when only slides are printed and the `portrait` option is not used.

here Each slide is centered horizontally, separated from adjoining text or other slides by the rubber length `\slidesep`. The default value of `\slidesep` is `\intextskip`. This is useful with the `notes` selection when you want the slides to precede accompanying comments and when you do not mind large spaces at the bottom of pages.

here* This is like `here*`, but with `here*` the length `\slidesep` is not lost when it falls at the beginning of a page or after a slide.

If you want each slide environment to begin a new page (with any of the selections), put the command

```
\slideclearpagetrue
```

in the preamble.

The `notesonly` selection has a starred version, `notesonly*`, which produces a slide marker for each slide environment, like this one:

Slide 4

Commands that are irrelevant in the `article` format are simply ignored. The idea is that it should be possible to switch back and forth between the two formats without making any other changes in the document. However, this is not entirely possible. Changing `\parindent` or `\textwidth` in the preamble affects both formats. You can get around this using the `\ifarticle` conditional. For example,

```
\ifarticle
  blah blah
\else
  blee blee
\fi
```

The `\else` part is optional.

16 Page styles

The page styles `empty`, `plain`, `headings` and `myheadings` work like in LaTeX's article style. There is also a page style `align` which puts `+` signs in the corners, like in SLiTeX.[10] You can change the size of the fonts used in headers and footers in the slides format by setting the commands

```
\slideheadfont
\slidefootfont
```

to the desired size, using `\renewcommand` (but any explicit font declarations in a page style override these commands). The default definition of these commands is `\scriptsize`.

17 Other style options

The style options described so far, `portrait`, `article`, `slidesonly`, `notes`, and `notesonly`, were designed specifically for `seminar.sty`. There are other such style options described elsewhere in this documentation: `semrot` (page 36), `semlayer` (page 20). `semcolor` (page 36), `semhelv` (page 33), `semlcmss` (page 33), and `a4` (page 37).

Here are two more style options that are part of the `seminar.sty` package, but that can be used with other document styles:

fancybox `fancybox.sty` is specific to slides, but it contains commands and documentation that are useful for making slides.

slidesec This sets up some sectioning/heading commands for slides, and lets you print a list of slides or a table of contents. See the file `slidesec.sty` for details.

It is impossible to predict, much less describe, the consequences of using all the other available LaTeX style option with each selection and format. Since this document style is an extension of LaTeX's `article` style, style options and commands that do not work with SLiTeX may well work here. Just about any style option that works with LaTeX's `article` style will work with the `article` format. When unsure of the effect of an option, just try it and see for yourself what happens.

18 Overlays

For overlays, you must use both the `semlayer` and `semcolor` style options (see page 36).

[10]See question 4 in Section 19 for suggestions on defining new page styles.

Overlays have two purposes. First, you can use overlays to gradually add layers of information on a slide during a presentation. Second, you can use overlays to make color layers; each layer is printed in black-on-white on paper, and then you use some kind of color copying service to put each layer on a transparency in a different color. This is not nearly as nice as having a color printer or using a color printing service, but it's the next best thing.

`seminar.sty` lets you make both types of overlays (and the main slide) from a single slide environment. The command[11]

`\overlay{`*n*`}`

puts whatever is in its scope (TEX group) on overlay n, where $n = 0, \ldots, 9$. Actually, overlay 0 refers to the main slide, but you might use `\overlay{0}` because overlay commands can be nested. As implemented by the `semcolor` style option, these overlay commands can be used just about anywhere, including in math mode, tables, and around an included graphics file (if it is a conforming EPS file, at least). Also, it is all right to use non-consecutive overlay numbers.

For color layers, you have to start by defining some colors using the command

`\colorlayers{`*colors*`}`

colors should be a comma separated list of color names, without spaces, as in

```
\colorlayers{red,green,blue}
```

Then you can use the command `\red` just like the command `\overlay{`*n*`}`; everything in its scope goes on a red color layer.

The `\colorlayers` command obeys the usual rules on scope. You can use this command any time, including in a slide environment. The command is cumulative, meaning that previously defined color layers continue to exist. The command was purposely defined so that it does not complain when a color name is already defined; this makes it easier to switch from some other color system to a layer system. However, you should be careful not to inadvertently redefine some command that you need. Fortunately, there are no TEX primitives whose names are the names of colors.

When you print out the slides, the main slide is printed, followed by each of the color layers (if any) for the main slide. Then each of the overlays is

[11] You can also write

`\begin{overlay}{`*n*`}` ... `\end{overlay}` printed, together with each of its color layers. Only overlays or color layers that are actually used (i.e., that are not empty) are printed.

You can turn overlays and color layers on and off with the commands

```
\overlaystrue
\overlaysfalse
\layerstrue
\layersfalse
```

These commands can be used at any time, and they obey the usual scoping rules. The default is for overlays to be active in the `slides` format and suppressed in the `article` format.

The counter `overlay` keeps track of the overlays. The default definition of `\theoverlay` is:

```
\theslide-\alph{overlay}
```

Overlays can be cross-referenced.

The command `\currlayer` is set to the name of the current color layer. `\thelayer` makes a label for layers; its default definition is:

```
\theoverlay-\currlayer
```

Color layers cannot be cross-referenced.

For example, if slide 7 has overlays 1 and 2 and colors `red` and `green`, then the main slide is numbered 7, followed by layers 7-red and 7-green, followed by overlay 7-a, followed by layers 7-a-red and 7-a-green, followed by overlay 7-b, followed by layers 7-b-red and 7-b-green.

The caption used in the slide styles is `\overlaylabel` for overlays and `\layerlabel` for layers. The defaults are, respectively,

```
\bf Overlay \theoverlay
\bf Layer \thelayer
```

By default the overlays and layers use the same page styles in the `slides` format as their "owner." You can specify special page styles with the commands:

```
\overlaypagestyle{style}
\layerpagestyle{style}
```

Finally, by default the overlays and layers use the same frame style as their "owner." You can specify special frame styles with the commands:

\overlayframe[*commands*]{*style*}
\layerframe[*commands*]{*style*}

III Help

Road map:

- If you just want to figure out how to do something, check Section 19, "Tips and tricks."
- If you are trying to decipher an error message, check Section 20, "Errors."
- If you are trying to solve some system-dependent problem that has arisen, check Section 21, "Troubleshooting."
- To convert SLITEX files, see Section 22, "Converting SLITEX files".

19 Tips and tricks

1 How can I fit more material in the slides?

Or alternatively, make the little material you have fill up the slide.

Let us count the ways:

1. Change the magnification with `\slidesmag{`*n*`}` (affects all the slides).
2. Change the height and/or width of one or all the slides.
3. Use `\ptsize{`*n*`}`.
4. Change `\slidestretch`.
5. Use the `\raggedslides` command, which changes the propensity to hyphenate.

2 How do I include an Encapsulated PostScript figure?

The only thing tricky about including EPS files is getting the size right. For example, suppose the you want to include a postscript file in a slide using the `epsf.sty` macros, and you want it to be 6 inches wide. Then this will do the trick:

```
\setslidelength{\epsfxsize}{6in}
\epsffile{mypic.eps}
```

If you include the command

```
\espfslidesize
```

(e.g., in the preamble), then `epsf.sty` will take care of scaling the size for you, and so you can just set `\epsfxsize` and `\epsfysize` to their magnified sizes (using `\setlength` rather than `\setslidelength`). Or you can not set these parameters at all, and then the eps file appears at its natural size on the slide.

Either way, this will have the expected effect:

```
\setlength{\epsfysize}{.8\textheight}
\epsffile{mypic.eps}
```

`psfig.tex` does not handle magnification properly. By using magnified dimensions, without `\setslidelength`, as in

```
\psfig{file=mypic.eps,width=6in}
```

the picture should appear correctly within a slide, but then it will not scale properly if you try to use the `article` option or the `\twoup` command. The solution, if using Rokicki's `dvips`, is to use `epsf.sty` (written by the man himself).

3 How can I print just selected overlays, layers or pages of notes in the `slides` format?

The `\onlyslides` and `\notslides` commands affect only slides. If a slide is omitted, so are all its overlays and color layers. To be more selective, you have to use your dvi driver to select the pages to be printed.

In the `slides` format, the page numbers recognized by dvi drivers correspond to the numbers of the slides. This means that a slide and all the overlays, layers and notes that correspond to that slide have the same page number. Some drivers allow one to select occurrence n of a page number. E.g., with `dvips`,

```
dvips -p2.1 -l2.4 myslides
```

will print the first through fourth page of overlays or notes that follow slide 2.

If your dvi driver does not support such selection, and you would prefer that the driver recognize physical page numbers, then put the command `\truepagenumbers` in the preamble.

There is nothing special about defining new page styles in `seminar.sty`. However, to make it easier to do this in the preamble, the commands

```
\newpagestyle{style}{header}{footer}
\renewpagestyle{style}{header}{footer}
```

are provided.[12]

style is the name of the page style, and *header* is the header, and *footer* is the footer. These can only be used for simple page styles that are the same for odd and even pages, and that do not do anything special with section marks.

Headers and footers are set in an `\hbox` the width of the page (`\textwidth`). You can use stretchable space such as `\hspace*{\fill}` or `\hfil` to center some information or put it flush against the margins. See the definitions of page styles in `latex.tex` and `article.sty` for examples.

Here is an example: Professor Starr wants lots of information in the headers and footers for the slides, and so she defines the page style `mypagestyle`:[13]

```
\newpagestyle{mypagestyle}%
  {\sl Big U \hfil \thedate \hfil \thepage}%
  {\hfil File \jobname.tex; printed \today\hfil}
```

Prof. Starr wants to use the standard `headings` page style for notes and the `mypagestyle` page style for slides, and she wants overlays to just have the overlay number. Therefore, she defines another page style for the overlays:

```
\newpagestyle{myoverlays}{\hfil \thepage}{}
```

and she puts

```
\pagestyle{headings}
\slidepagestyle{mypagestyle}
\overlaypagestyle{myoverlays}
```

[12] You can also use the macros in `fancyheadings.sty`, which is available from various archives. However, you have to set the pagestyle *after* `\begin{document}`. Otherwise, the dimensions get screwed up. If using the `slidesonly` selection, then you also need to enclose the `\pagestyle` command in an `allversions*` environment. E.g.,

```
\begin{allversions*}
  \pagestyle{fancy}
\end{allversions*}
```

[13] `\thedate` is set with the `\date` command.

in the preamble.

5 How do I change a parameter only for the `slides` format?

The conditionals

```
\ifarticle ... \else ... \fi
\ifslidesonly ... \else ... \fi
\ifnotes ... \else ... \fi
\ifnotesonly ... \else ... \fi
\ifportrait ... \else ... \fi
```

allow one to select for what versions material is to be processed. `\ifnotes` is true if and only if the `notes` selection is in effect, and so on. The `\else` clause is optional. See *The TEXbook*, Chapter 20, for more information about using conditionals.

For example, the first line below sets the slide rotation to `right` in the `article` format only. The second line changes the page style for the `notes` and `notesonly` selections and `article` format:

```
\ifarticle\sliderotation{right}\fi
\ifarticle\ifslidesonly\else\pagestyle{myheadings}\fi\fi
```

6 Why does extra space get inserted at the top of a slide when I begin the slide with a color or overlay command (when using the `semcolor` option)?

Color and overlay commands with the `semcolor` option use `\special`'s. TEX adds the space `\parskip` between the `\special`'s and the first material in the slide.

Here is the workaround: If that material is an ordinary paragraph, put the command `\leavevmode` just after the color or overlay command. In other cases, if you are sure this is the problem, put `\vskip-\parskip` just before the color or overlay command.

20 Errors

There are several errors that identify themselves as being from `seminar.sty`, but the error messages are so self-explanatory that that there is no need to describe them here. Instead, this section explains a few especially cryptic TEX error messages that can arise when using `seminar.sty`. These errors may also arise for reasons that are not particular to the `seminar.sty` macros. See your favorite TeX and LaTeX manuals for more help in debugging your documents.

```
! File ended while scanning use of \next.
```

You are missing an `\end{document}` on the main file or an `\endinput` on a file that is input.

```
! File ended within \read.
```

You may have an unmatched curly brace following `\begin{slide}` on the same line. See Section 14.

```
! Paragraph ended before \begin@slide was complete.
```

You have not specified the optional argument for the `slide` or `slide*` environment correctly (see Section 2), or the first character in a slide environment is a `[` (put a pair `{}` of braces before the `[`).

21 Troubleshooting

1 A few slides do not come out two-up with the `article` option.

First, read Section 12 carefully. If the slides still do not come out twoup, it might be that there is extraneous output between the slides. Try putting the commands

```
\slideplacement{here*}
\setlength{\slidesep}{8pt plus 1fill}
```

in the preamble (`8pt` should be one-half the minimum distance between slides). If this solves the problem, look for the extraneous output, or just leave those commands in the preamble.

2 What kind of incompatibilities are there between `seminar.sty` and other macros?

Of course, whenever you load macros that are not part of the standard LaTeX distribution and that were not designed to work with `seminar.sty`, problems may arise because of name conflicts. What are listed here are changes made to LaTeX commands that may conflict with other macros that also redefine these command. The problems listed here are very unlikely to occur, however, unless you have a habit of seriously hacking the standard LaTeX macro files.

1. This style modifies the definitions of the LaTeX primitive `\document`. It generally will not be upset by, nor will it void, modifications to `\document` made before `seminar.sty` is input. However, subsequent modifications of `\document` may cause problems.

2. `xcomment.sty`, which is input by `seminar.sty` with the `slidesonly` selection, modifies the definition of `\end` within included environments. This is generally compatible with modifications to `\end` made outside environments, but may conflict with modifications to `\end` made inside environments.

3 Why are the landscape slides displayed sideways and the portrait slides displayed upside down on my previewer?

This document style makes frequent use of landscape mode. Some DVI-to-PS converters, such as older versions of Rokicki's `dvips`, use their own PostScript macros to print a landscape document, rather than simply instructing PostScript to use landscape mode. If the PostScript output of such a converter is viewed using a PostScript previewer that does not allow you to choose the orientation of the display, the output will be positioned correctly on the page, but the page will always be displayed in portrait mode. When viewing slides without the `portrait` option, the landscape slides will be sideways, and the portrait slides will be upside-down!

There is nothing this style can do to coerce the page to be displayed in landscape mode. There are various ways to minimize neck strain, however:

1. The direction in which the portrait slides are rotated can be reversed, so that they end up right-side-up. Just put

   ```
   \sliderotation{right}
   ```

   before the beginning of the document (assuming that you are using rotation macros).

2. You can also get the portrait slides to be displayed right-side-up by using the `portrait` style option (Section 17).

3. In the `article` format, the document is typeset in portrait mode (unless the `portrait` option is used), and so the landscape slides are right-side-up. You can use this format when composing and proof-reading the slides.

4 Why is my dvi driver soooo slow?

Probably it cannot find the right font bitmaps, and so it is either automatically generating new ones, or it is scaling the ones it can find. Either way, as long as you can somehow generate the missing font bitmaps, this problem is transitory. See Appendix C for details.

5 Why does TEX complain about missing circle fonts?

Older versions of the NFSS use the names `circle10` and `circlew10` for the LATEXcircle fonts, instead of the otherwise standard names `lcircle10` and `lcirclew10`. You can copy your `.pk` and `.tfm` files to the new names, or get a new version of the NFSS.

22 Converting SLITEX files

`seminar.sty` can do everything SLITEX can do, and much more[14] Here is a brief and incomplete description of how to do with `seminar.sty` what you can do with SLITEX, and how to convert SLITEX files.

- Use `seminar` as your document style, instead of `slides`.
- Run LATEX or AMS-LATEX, instead of SLITEX.
- With `seminar.sty`, the preamble and slides can all go in the same file.
- The default in `seminar.sty` is to get landscape slides. If you want to convert a SLITEX file containing portrait slides, add the `portrait` style option, and replace your `slide` environments by `slide*` environments.
- With `seminar.sty`, notes do not need to go in a separate environment. To convert a SLITEX file containing `note` environments, define a note environment that does nothing:

  ```
  \newenvironment{note}{}{}
  ```

- For color layers:
  - Use the `\colorlayers` command instead of the `\colors` command.
  - Delete the argument to the `slide` and `slide*` environments that lists the color layers.

[14]However, color layers and overlays require PSTricks and a PostScript printer.

  - Include the `semlayer` style option.

- `seminar.sty` does not use separate environments for overlays:
  - Remove the `overlay` environments.
  - Replace `\invisible` commands by `\overlay{1}`, `\overlay{2}`, etc.
  - Include the `semlayer` style option.

- To actually produce the overlays and color layers, you have to have PSTricks and a PostScript printer, and you must include the `semcolor` style option.

- Use `\onlyslides{`*list*`}` and `\onlynotestoo` instead of `\onlynotes`.

Appendices: Configuration

Before reading this appendix, you should follow the installation instructions in the file `sem-read.me` that is distributed with `seminar.sty`.

A The short story about fonts

`seminar.sty` is a LaTeX style, and you can use whatever fonts that are compatible with LaTeX (or AMS-LaTeX). However, you are likely to want to use special fonts with `seminar.sty`; see Appendix B for help. Furthermore, you are likely to want to use fonts in sizes that are larger than the standard sizes; see Appendix C for help.

But if you don't want to read these appendices, you can just use whatever LaTeX fonts you know how to use (e.g., the standard Computer Modern fonts). To be sure you need only standard font magnifications:[15]

- Don't use the `11pt` or `12pt` style options (or the `\ptsize` command).
- Don't change the document magnification (or only use 0–4 magsteps).
- Don't use any font size larger than `\large`.
- Don't use the `\twoup` command.

B Choosing fonts

So that you can use different fonts for the notes and the slides (if you want), the command `\slidefonts` is invoked at the beginning of every slide environment. Define it to set up any special fonts for the slides.

Here are the font configurations that come ready-to-use with `seminar.sty`. You might also use these as a starting point for your own configurations.

1. You can just use the regular old Computer Modern fonts that you probably grew up on. For this, you don't have to do anything at all. Ain't that easy? But see the next section about font bitmaps.

[15]This is not a consideration if you are using only PostScript or other scalable fonts.

2. The `semhelv` style option sets up the PostScript Helvetica text fonts for the slides. The Computer Modern fonts are still used for math and for the notes with the `article` format. You need a PostScript printer, a dvi-to-ps driver that supports PostScript fonts (e.g., Rokicki's `dvips`), and the New Font Selection Scheme. You should also check the font file names used in `semhelv.sty`, and change them if necessary to match the names on your system.

   This combination of fonts is highly recommended because sans serif fonts look good for slides, Helvetica is a resident font in just about all PostScript printers, and the fonts are scalable and thus there is no problem of needing new bitmaps.

3. The `semlcmss` style option sets up the SLITEX sans serif fonts for the slides, and uses the Computer Modern fonts for notes with the `article` format. The SLITEX fonts are ugly, but they might be the only usable sans serif font you have on your system and you are really dying for that kind of font. Also, they usually come ready for printing magnified documents, and so you won't need too many new font bitmaps. You must have the New Font Selection Scheme. `semlcmss` uses

   ```
   \slidesmag{4}
   \ptsize{10}
   ```

   Deviating from this will increase the need for new font bitmaps.

The New Font Selection Scheme (NFSS) mentioned above is a macro package for LaTeX that greatly simplifies using non-standard fonts. The NFSS was written by Frank Mittelbach and Rainer Schöpf, and is available from various archives, including:

`ftp.uni-stuttgart.de`

It is far easier to take the 10 minutes or so that are required (in theory, at least) to install the NFSS, than it is to try to muck around with LaTeX's font primitives.

C Font bitmaps

If you are only using PostScript or other scalable fonts, or if you obey the restrictions listed on page 32, then you can ignores this appendix.

To use `seminar.sty`, you may need font bitmap sizes that are not currently found on your system. This section describes how to avoid this and what to do about it. First, a few paragraphs about magnification and font bitmaps.

size	default (10pt)	11pt option	12pt option
`\tiny`	0	0	0
`\scriptsize`	0	0	0
`\footnotesize`	0	0	0
`\small`	0	0	1/2
`\normalsize`	0	1/2	1
`\large`	1	1	2
`\Large`	2	2	3
`\LARGE`	3	3	4
`\huge`	4	4	5
`\Huge`	5	5	5

Table 1: Font magnification in magsteps (n means a magnification of 1.2^n) for a font that is available in 5pt, 6pt, 7pt, 8pt, 9pt and 10pt sizes.

TeX's Metafont fonts are designed for a type size, such as 5pt. Most font families are available at least in the sizes 5pt, 6pt, 7pt, 8pt, 9pt and 10pt. Some are also available in the sizes 12pt and 17pt (and others), but often any size above 10pt is obtained by scaling the 10pt fonts. Table 1 lists the possible magnifications for fonts, depending on which option you are using (`10pt`, `11pt` or `12pt`), and depending on the LaTeX type size declaration that is in effect (e.g., `\small`). The magnifications are given in magsteps, which is TeX's standard unit for font magnifications. n magsteps means a magnification of 1.2^n.

Both the `article` and the `slides` formats may also magnify the document. By default, the magnification of the `slides` format is 4 magsteps, and the magnification of the `article` format is 0. To find the total font magnification, add the magnification listed in the table to the magnification of the document. E.g., in the `slides` format with the `12pt` option (or the command `\ptsize{12}` at the beginning of a slide), the `\large` command may invoke fonts that have a magnification of up to 7 magsteps.

Most systems have font bitmaps for 0, 1/2, 1, 2, 3, 4 and 5 magsteps. If you obey the restrictions listed on page 32, then you will only need fonts in these standard magnifications. However, if you want to use other font magnifications, then check with your system administrator to determine which of the following applies to you (or just plunge ahead and see what happens):

1. If you have Metafont and `dvips` or some other driver that automatically

generates font bitmaps as needed, and if this feature is enabled, then you will simply notice that it takes a long time to print documents at first, because the driver has to wait for the new font bitmaps to be made. Eventually, you will have generated all the extra bitmaps you need, and this delay will go away.

2. If you have Metafont, but your dvi driver does not automatically generate needed bitmaps, then you just need to keep track of what font bitmaps you are missing as you use `seminar.sty` and occasionally run Metafont to make them. You will also notice a delay when a font bitmap is missing, because your driver will probably scale the closest bitmap it finds, and this can take time on some systems.

3. If you do not even have Metafont, then you have to try to get by without the extra bitmaps. Scaled fonts look lousy, but an occasional scaled font in a heading is not so bad.

Of course, PostScript and other scalable fonts do not present any problem, and so it is a good idea to use these as much as possible. The Computer Modern fonts are available from Blue Sky Research in PostScript Type I format. If you do not have PostScript versions of the CM fonts, and you instead use other PostScript fonts for text, then you are likely to still need bitmapped fonts for mathematics. However, since mathematics is usually set at `\normalsize` or smaller, this is not a big problem.

If you are installing `seminar.sty` on a multi-user system, then hopefully you will make the needed fonts available.

D Color

You can use whatever color commands you ordinarily use with LaTeX. You might try the `semcolor` style option, which lets you use the PSTricks color commands for printing on a color PostScript printer. The `semcolor` option combined with the `semlayer` option (see Section 18) lets you print color layers.

E Landscape printing and slide rotation

If your dvi driver supports a `\special` for landscape printing, then you can define `\printlandscape` in the preamble of your document to invoke this command. E.g., for `dvips` put the line

```
\renewcommand{\printlandscape}{\special{landscape}}
```

in the preamble. Otherwise, `seminar.sty` will display a message reminding you to print your document in landscape mode, when appropriate.

You can print both landscape and portrait slides in one shot if you can rotate the portrait slides when printing in landscape mode or the landscape slides when printing in portrait mode. If you are using a PostScript printer, you probably are using a dvi driver that supports rotation; otherwise, you probably are not. Here are suggestions for setting up rotation:

- The `semcolor` option provides an interface to the rotation macros in PSTricks (see Appendix F); it works with many popular dvi-to-ps drivers.
- If you are using Rokicki's `dvips` and want rotation but do not want to load the entire PSTricks package, then use the `semrot` option instead of the `semcolor` option.
- If the `semcolor` and `semrot` options do not work for you, but you have your own rotation macros, then you have to define the commands

  ```
  \leftsliderotation
  \rightsliderotation
  ```

  so that they rotate something left and right, respectively. For example,

  ```
  \renewcommand{\leftsliderotation}[1]{\rotateleft{#1}}
  ```

In any case, you can determine the direction of rotation using

`\sliderotation{`*direction*`}`

where valid directions are `none`, `left` and `right`. The default is `left`.

By default, the headers and footers aren't rotated, but you can switch between rotating and not rotating the headers with the commands:

```
\rotateheaderstrue
\rotateheadersfalse
```

F The semcolor style option

The `semcolor` option sets up an interface between `seminar.sty` and the PSTricks package. PSTricks is a collection of PostScript macros for TeX. It works with Rokicki's `dvips`, and several other dvi-to-ps drivers.

You can probably get the PSTricks package from wherever you obtained `seminar.sty`, or check the archives listed in the PSTricks read-me file, `read-me.pst`, which is distributed with `seminar.sty`.

What the `semcolor` style option gives you, compared to just using the PSTricks package, is:

Color A small patch to make the PSTricks color commands more robust in slides.

Rotation The rotations `left` and `right` are defined. `left` is the default.

Framing The frame styles `scplain` (using `\psframebox`), `scdouble` (using `\psdblframebox`) and `scshadow` (using `\psshadowbox`).

Overlays Overlays and layers, when used in conjunction with the `semlayer` option.

G A4 and other paper sizes

Use the `a4` option when using A4 paper. Note that this option does not correspond to an independent file.

If you want to configure seminar.sty for A4 paper by default (without having to include style options), then you can do one of the following:

1. Add the following line to `seminar.con` (see Appendix H):

   ```
   \input{sem-a4.sty}
   ```

2. Add the following lines to `seminar.con` (these lines are just the contents of `sem-a4.sty`):

   ```
   \def\paperwidth{210mm}
   \def\paperheight{297mm}
   \input sem-page.sty
   \slidewidth 222mm
   \slideheight 152mm
   ```

3. Create a file such as `mysem.sty`, to be used as a document *style* (rather than style *option*), with the following lines:

   ```
   \def\paperwidth{210mm}
   \def\paperheight{297mm}
   \input seminar.sty
   \slidewidth 222mm
   \slideheight 152mm
   ```

For other paper sizes, you can create a style option by modifying `sem-a4.sty`. Then any of the options described above is available to you, (with the appropriate parameter values). However, this will give satisfactory results only for paper sizes that are close to A4 or 8.5in by 11in.

Note that for any paper size, even 8.5in by 11in, the page parameters that are set this way are just suggested defaults, and most people will want to customize them. Because of the variety of ways in which `seminar.sty` documents can be printed, this is a little more complex than with most document styles. Examine `sem-page.sty` to see what page parameters need to be set, and when.

H Configuration file

You can put customizations to `seminar.sty` in a file named `seminar.con`. This file is optional. It is loaded by `seminar.sty` if it exists, *before* loading the style option files. E.g., `seminar.con` might contain the following lines:

```
\input semhelv.sty
\input semcolor.sty
\renewcommand{\printlandscape}{\special{landscape}}
\endinput
```

Index

2up (option), **13**

a4 (option), **37**
\addtoartlength, **18**
\addtoslidelength, **6**
\addtoslidelist, **17**
\addtoslidereset, **12**
allversions (environment), **17**
allversions* (environment), 16, **17**, 26
article (option), 7, **13**, 13, 14, 18, 25, 28, 33
\articlemag, **13**, 14, 18

\centerslidesfalse, **2**
\centerslidestrue, **2**
\colorlayers, **21**
counters:
- overlay, **22**
- slide, **12**

\currlayer, **22**

environments:
- allversions*, 16, **17**, 26
- allversions, **17**
- note, **16**, 16, 17
- slide*, ii, **1**, 1
- slide, ii, **1**, 1

\espfslidesize, **25**
\everyslide, **15**
\extraslideheight, **3**

fancybox (option), 10, 11

\ifarticle, 19, **27**
\ifnotes, **27**
\ifnotesonly, **27**
\ifportrait, **27**
\ifslidesonly, **27**

\landscapeonly, **1**
\layerframe, **23**
\layerlabel, **22**
\layerpagestyle, **22**
\layersfalse, **22**
\layerstrue, **22**
\leftsliderotation, **36**

\newpagestyle, **26**
\newslide, **3**, 12
\newslideframe, **10**
note (environment), **16**, 16, 17
notes (option), **15**, 18, 19
notesonly (option), **15**, 18, 19
notesonly* (option), **19**
\notslides, **12**
\noxcomment, **16**, 17

\onlynotestoo, **16**, 31
\onlyslides, **12**, 31
options:
- 2up, **13**
- a4, **37**
- article, 7, **13**, 13, 14, 18, 25, 28, 33
- fancybox, 10, 11
- notesonly*, **19**
- notesonly, **15**, 18, 19
- notes, **15**, 18, 19
- portrait, ii, **1**, 1, 13, 14, 18, 19
- semcolor, 10, 11, 20, 27, 31, 35, **36**, 36, 37
- semhelv, **33**
- semlayer, **20**, 20, 31, 35, 37
- semlcmss, **33**
- semrot, **36**, 36
- slidesonly, **15**, 26
- slides, 18

\overlay, **21**, 21
overlay (counter), **22**
\overlayframe, **23**
\overlaylabel, 22
\overlaypagestyle, **22**

\overlaysfalse, **22**
\overlaystrue, **22**

\paperheight, 5
\paperwidth, 5
portrait (option), ii, **1**, 1, 13, 14, 18, 19
\portraitonly, **1**
\printlandscape, **35**
\ptsize, **8**, 32

\raggedslides, **2**, 2, 24
\renewpagestyle, **26**
\rightsliderotation, **36**
\rotateheadersfalse, **36**
\rotateheaderstrue, **36**

\semcm, 5, **6**
semcolor (option), 10, 11, 20, 27, 31, 35, **36**, 36, 37
semhelv (option), **33**
\semin, 5, **6**
semlayer (option), **20**, 20, 31, 35, 37
semlcmss (option), **33**
semrot (option), **36**, 36
\setartlength, **18**
slide (counter), **12**
slide (environment), ii, **1**, 1
slide* (environment), ii, **1**, 1
\slidearraystretch, **8**
\slidebottommargin, **5**, 5
\slideclearpagetrue, **19**
\slidefonts, **15**
\slidefootfont, **20**
\slidefootnotesep, **8**
\slideframe, **10**, 10, 11
\slideframesep, 5, **10**
\slideframewidth, 5, **10**
\slidefuzz, 4
\slideheadfont, **20**
\slideheight, **2**, 5, 14
\slideitemsep, **8**
\slidelabel, **14**, 14
\slidelabelsep, **8**
\slideleftmargin, **5**, 5
\slideleftmargini, **8**
\slideleftmarginii, **8**
\slideleftmarginiii, **8**
\slidepagestyle, **16**
\slideparindent, **8**
\slideparskip, **8**
\slidepartopsep, **8**
\slideplacement, **18**
\slidereset, **12**
\sliderightmargin, **5**, 5
\sliderotation, **36**
slides (option), 18
\slidesep, **19**, 19
\slideshrink, **9**, 9
\slideskip, **9**, 9
\slidesmag, **6**, 14, 24
slidesonly (option), **15**, 26
\slidestretch, **8**, 9, 24
\slidestyle, **14**
\slidetopmargin, **5**, 5
\slidewidth, **2**, 5, 14

\thelayer, **22**
\theoverlay, **22**
\theslide, **12**, 12
\theslidefootnote, **12**
\truepagenumbers, **25**
\twoup, **13**, 13, 14, 25, 32