

Using bardproj.sty Style for  
Bard Senior Projects  
and  
Bard M.A.T. Mathematics Research Projects

Ethan Bloch

March 24, 2008

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Document Format</b>	<b>3</b>
<b>3</b>	<b>Labels</b>	<b>3</b>
<b>4</b>	<b>Chapters and Sections</b>	<b>4</b>
<b>5</b>	<b>Theorems and the like</b>	<b>4</b>
<b>6</b>	<b>Numbered Displayed Equations</b>	<b>6</b>
<b>7</b>	<b>Some Standard T<sub>E</sub>X Mistakes</b>	<b>6</b>
<b>8</b>	<b>Miscellaneous Mathematics Symbols</b>	<b>7</b>
<b>9</b>	<b>Comments</b>	<b>7</b>
<b>10</b>	<b>Typesetting Simulated Computer Code</b>	<b>8</b>
<b>11</b>	<b>Title Page</b>	<b>11</b>
<b>12</b>	<b>Table of Contents and List of Figures</b>	<b>11</b>
<b>13</b>	<b>Abstract, Dedication and Acknowledgments</b>	<b>12</b>
<b>14</b>	<b>Double spacing</b>	<b>12</b>
<b>15</b>	<b>Figures</b>	<b>13</b>
<b>16</b>	<b>Bibliography</b>	<b>15</b>
	<b>Sample Senior Project Outline</b>	<b>18</b>

## 1 Introduction

This manual explains how to use the style file “bardproj.sty.” This style file is designed for Bard senior projects and M.A.T. mathematics research projects, and takes care of a number of formatting issues for senior projects, such as the title page, dedication, acknowledgements and correct margins; it also includes formatting for theorems, definitions and the like, as well as a few miscellaneous items. It is assumed that you already know the basics of LaTeX. Only the specifics of formatting a project are discussed here.

This style file combines good amounts of svsing2e.sty (Springer-Verlag) and Ragsmac.sty (from Bob McGrail), and has theorem formatting by Ethan Bloch, with help from Amy Hendrickson. Aaron Schumacher made the style file work for M.A.T. mathematics research projects. The file “bardproj.sty,” as well as the associated template “bardproj\_template.tex,” can be obtained from Ethan Bloch (bloch@bard.edu), and it can be downloaded from <http://math.bard.edu/bloch/bardtex.htm>.

## 2 Document Format

The basic form of the document is

```
\documentclass[11pt, onside, reqno]{book}
\usepackage{amssymb, amsthm, amsmath, amsfonts}
\usepackage{bardproj}
\usepackage{amsrefs}

\begin{document}

<Text of document>

\end{document}
```

## 3 Labels

One of the great features of LaTeX is that theorems, equations and the like are numbered automatically. Thus, if you decide to insert a lemma in the middle of a section, everything after that is automatically renumbered. The way automatic numbering works is that everything that is numbered is given a label, usually written

```
\label{<Label>}
```

The label can be any character string starting made of numbers and letters, starting with a letter. Of course, each labeled item must have a unique label. Since the label does not appear in the printed version of the text, you can choose anything you want for the label; something mnemonic is always a good idea.

To refer to the labeled item, for example a theorem, write

```
Theorem~\ref{<Label>}
```

This reference could be anywhere in the text, even before the item being referred to.

## 4 Chapters and Sections

The format for chapter and section headings are

```
\chapter{<Chapter name>}\label{<Label>}
```

and

```
\section{<Section name>}\label{<Label>}
```

Do not write the chapter or section number as part of the name, since the number is inserted automatically. To refer to the chapter or section, write

```
Chapter~\ref{<Label>}
```

and similarly for sections.

## 5 Theorems and the like

The format for theorems is

```
\thm\label{<Label>}  
<Statement of theorem>  
\ethm
```

To refer to the theorem, write

```
Theorem~\ref{<Label>}
```

If you want to give the theorem a name, for example “The Pythagorean Theorem,” use the format

```
\thm[<Name of Theorem>\label{<Label>}  
<Statement of theorem>  
\ethm
```

For other theorem type environments, we have

```
\lem\label{<Label>} ... \elem      (Lemma)  
\coro\label{<Label>} ... \ecoro     (Corollary)  
\prop\label{<Label>} ... \eprop     (Proposition)  
\conj\label{<Label>} ... \econj     (Conjecture)  
\clm\label{<Label>} ... \eclm      (Claim)  
\expl\label{<Label>} ... \eexpl     (Example)  
\remk\label{<Label>} ... \eremk     (Remark)  
\alg\label{<Label>} ... \ealg      (Algorithm)  
\defn\label{<Label>} ... \edefn     (Definition)
```

If you want a theorem or the like that is not numbered (though it can still be given a name), use

```
\thmn\label{<Label>} ... \ethmn     (Theorem)  
\lemn\label{<Label>} ... \elemn     (Lemma)  
\coron\label{<Label>} ... \ecoron    (Corollary)  
\propn\label{<Label>} ... \epropn    (Proposition)  
\conjn\label{<Label>} ... \econjn    (Conjecture)  
\clmn\label{<Label>} ... \eclmn     (Claim)  
\expln\label{<Label>} ... \eexpln   (Example)  
\remkn\label{<Label>} ... \eremkn   (Remark)  
\algn\label{<Label>} ... \ealgn     (Algorithm)  
\defnn\label{<Label>} ... \edefnn   (Definition)
```

The environment for proofs is

```
\demo ... \edemo
```

If a proof has an alternative title (e.g. “Proof of Theorem 1.2.3”), then use the following environment, where the alternative title could include a reference to a theorem or the like

```
\demoname{alternative title} ... \edemoname
```

## 6 Numbered Displayed Equations

The format for a numbered displayed equation is

```
\begin{equation}\label{<Label>}
<Equation>
\end{equation}
```

Do not write the equation number, since that is done automatically. You also do not put in  $\$ \dots \$$  in this case.

To refer to the equation, write

```
Equation~\ref{<Label>}
```

## 7 Some Standard T<sub>E</sub>X Mistakes

- To write a set such as

$$\{x \in \mathbb{R} \mid x^2 < \pi\}$$

we write

```
\{x \in \mathbb{R} \mid x^2 < \pi\}
```

Observe that the curly brackets are written `\{` and `\}`, and that the vertical line is written `\mid`, NOT `|` (which would produce the wrong spacing).

- To write an inner product such as

$$\langle x, y \rangle$$

we write

```
\langle x, y \rangle
```

Observe that the brackets are written `\langle` and `\rangle`, NOT `<` and `>` (which would produce the wrong spacing).

- To write a function such as

$$f: A \rightarrow B$$

we write

`f\colon A \to B`

Observe that the colon is written `\colon`, NOT `:` (which would produce the wrong spacing).

- DO NOT USE `\` when writing text. It is the wrong way to start new lines and paragraphs, and will often mess things up. Use `\` only inside special constructions such as `align`, `matrix`, etc.

## 8 Miscellaneous Mathematics Symbols

For convenience, there are a few some macros for commonly used mathematical symbols included in `bardproj.sty`. Don't use these macros without `bardproj.sty`.

To write the standard symbols for various sets of numbers, we have

Real numbers: `\rr`  
Complex numbers: `\cc`  
Rational numbers: `\qq`  
Integers: `\zz`  
n-dimensional space: `\rrr{n}`

To write  $f: A \rightarrow B$  write

`\func fAB`

If the name of the function, or the domain or codomain, consist of more than one character, put them in curly brackets.

To write  $g \circ f$ , write

`g \rc f`

All the expressions in this section must be in math mode.

## 9 Comments

To make a comment to yourself in your `.tex` document that is ignored when the document is compiled, use the `%` character. Any line in the `.tex` document that starts with `%` is ignored; if a `%` is inserted in the middle of a line, all the text to the right of the `%` symbol is ignored.

Unfortunately, the standard LaTeX program does not give a method for commenting out long sections of the .tex file, other than using lots of % symbols. There is a nice LaTeX package called `verbatim.sty` that solves this problem. This package can be downloaded from <http://math.bard.edu/bloch/bardtex.htm>. Place the package in the same folder in your computer as your .tex file. To use this package, you must first insert

```
\usepackage{verbatim}
```

into your file prior to

```
\begin{document}
```

Then, simply type `\begin{comment}` before the text that you want commented out, and `\end{comment}` after the text.

## 10 Typesetting Simulated Computer Code

The simplest way to write a small amount of simulated computer code is to use the `verbatim` environment. *Everything* within this environment is typeset in typewriter font exactly as it appears in the .tex file—obeying spaces and line breaks as in the .tex file, and not recognizing any special symbols.

To obtain

```
This is      verbatim text, where $ signs are
just $ signs, and commands such as \huge are ignored.
```

we simply write

```
\begin{verbatim}
This is      verbatim text, where $ signs are
just $ signs, and commands such as \huge are ingored.
\end{verbatim}
```

For arbitrarily long verbatim text, the LaTeX package called `verbatim.sty` must be used. This package can be downloaded from <http://math.bard.edu/bloch/bardtex.htm>. Place the package in the same folder in your computer as your .tex file. To use this package, you must first insert

```
\usepackage{verbatim}
```

into your file prior to

```
\begin{document}
```

Then, use the `verbatim` environment as above.

There is another package, called `moreverb.sty`, that contains some very useful variants of the `verbatim` environment; these environments can be used to make typesetting simulated computer code easier. This package can be downloaded from <http://math.bard.edu/bloch/bardtex.htm>. To use this package, you must first insert

```
\usepackage{moreverb}
```

into your file prior to

```
\begin{document}
```

The `verbatimtab` environment from the `moreverb.sty` package allows tab characters to be expanded to a given number of spaces (without this package,  $\text{\LaTeX}$  considers tab characters to be single spaces. In the `verbatimtab` environment, tab characters are by default set to equal eight space characters. In the following examples, the  $\triangleright$  character denotes a tab character.

To obtain

```
This      is      verbatimtab      environment.
```

we write

```
\begin{verbatimtab}
This>is>verbatimtab>environment.
\end{verbatimtab}
```

An optional argument after the definition of `verbatimtab` environment can be used to set tab characters to any desired number of spaces.

To obtain

```
This    is    verbatimtab    environment.
```

we write

```
\begin{verbatimtab}[4]
This>is>verbatimtab>environment.
\end{verbatimtab}
```

The `verbatimcmd` environment from the `moreverb.sty` package is generally like the `verbatim` environment, except that in the `verbatimcmd` environment any LaTeX command that start with a `\` is obeyed, and `{ }` is respected. Hence, in the `verbatimcmd` environment styles such as bold and italics can be used. Mathematics symbols can also be used in the `verbatimcmd` environment, though only those that start with the `\` symbol; therefore `$a^{x+y}$` would not be recognized (because neither `$` nor `^` starts with the `\` symbol), but it could be replaced with the less commonly used `\(a\sp{x+y}\)`, where `\( \)` are another way of starting and ending inline math mode, and `\sp` means superscript (we can also use `\[ \]` as another way of starting and ending display math mode, and `\sb` is used for subscript).

To obtain

This is  $a^{x+y}$  in `verbatimcmd`.

we write

```
\begin{verbatimcmd}
This is \(\a\sp{x+y}\) in {\bf verbatimcmd}.
\end{verbatimcmd}
```

The `listing` environment from the `moreverb.sty` package is like the `verbatim` environment, except that it numbers lines. The `listing` environment has two parameters, the first (in square brackets) specifies the step between numbered lines (a value of 1 makes every line numbered), and the second parameter (in curly brackets) specifies the number of the first line. The `listing` environment handles tab characters similarly to the `verbatimtab` environment, though tab characters are unalterably set to equal eight space characters.

To obtain

```
This is the environment that
8 numbers lines. In this example,
every other line is numbered,
10 and the first line is set at line 7.
```

we write

```
\begin{listing}[2]{7}
This is the environment that
numbers lines. In this example,
```

```
every other line is numbered,  
and the first line is set at line 7.  
\end{listing}
```

The `listingcont` environment from the `moreverb.sty` package is like the `listing` environment, except that it does not have any parameters, and it continues numbering lines where the previous `listing` or `listingcont` environment left off. The `listing*` and `listingcont*` environments from the `moreverb.sty` package are very much like the `listing` and `listingcont` environments respectively, except for two differences: they write `\_` for each blank space in the original text, and they do not handle tab characters.

## 11 Title Page

When you are ready to put your project into final form, add the title page as follows.

For a Bard undergraduate senior project, insert

```
\titlepg{<Project title>}{<Your name>}{<Month>}{<Year>}
```

into your text immediately after

```
\begin{document}
```

For a Bard M.A.T. mathematics research project, insert

```
\titlepgmat{<Project title>}{<Your name>}{<Month>}{<Year>}
```

into your text immediately after

```
\begin{document}
```

Make sure you put in your own title, name, and month and year of graduation. This command will automatically create a title page, and will automatically make a table of contents.

## 12 Table of Contents and List of Figures

When you are ready to put your project into final form, insert

```
\tableofcontents
```

into your text immediately after the title page.

If you need a list of figures, insert

```
\listoffigures
```

into your text immediately after the title page.

## 13 Abstract, Dedication and Acknowledgments

To put an abstract in your project, insert

```
\abstr
```

after the titlepage and before the table of contents, and then put in the text of your dedication. The appropriate heading, pagebreaks and the like will be done automatically.

It is customary (though not required) to put in a dedication and/or acknowledgments in your senior project. These should be put right after the title page, before the body of the text.

To format a dedication, insert

```
\dedic
```

and then put in the text of your dedication. The appropriate heading, pagebreaks and the like will be done automatically.

To format acknowledgments, insert

```
\acknowl
```

and then put in the text of your acknowlegments. The appropriate heading, pagebreaks and the like will be done automatically.

## 14 Double spacing

The final draft of your project needs to be double spaced. You might or might not want your drafts to be double spaced. For double spacing, insert

```
\doublespace
```

into your text immediately prior to the body of the text, after the title page, acknowledgments and dedications.

When you double space your text, you might wish to insert

```
\singlespace
```

just before your bibliography, so that the bibliography is single spaced.

## 15 Figures

There are two ways to deal with figures. One is to leave spaces for the figures, and then paste them into the printed out copy of your project; the other is to import computer generated figures directly into the .dvi file.

If you want to leave a space for a figure (which you will paste into the printed copy), use the format

```
\figspace{<Space>}{<Figure Label>}
```

where Space would be something like “1.5truein,” and Figure Label is the internal reference for the figure that you will use to refer to it. You should not put in the figure number, since that is done automatically by L<sup>A</sup>T<sub>E</sub>X. To refer to the figure elsewhere in the text, use the format

```
Figure~\ref{<Figure Label>}
```

If you want to import a computer generated figure directly into the .dvi file, the figure needs to be in a format that T<sub>E</sub>X can work with. If you use PCT<sub>E</sub>X, or most other PC implementations of T<sub>E</sub>X, the best format to use is .eps; PCT<sub>E</sub>X can also handle .jpg figures, though the way of doing so is particular to this implementation of T<sub>E</sub>X, in contrast to the method for .eps, which is universal. If you use T<sub>E</sub>XShop on the Mac, the necessary format is .pdf; the method of implementation for .pdf files in T<sub>E</sub>XShop is identical to the method for .eps file in most other T<sub>E</sub>X implementations.

How you actually create figures does not matter, as long as they are in the right format. A good program for creating .eps files is Adobe Illustrator. Graphs and the like created in Mathematica can also be saved as .eps files. Many programs can save files in .pdf format, for example Adobe Illustrator, and if you have Adobe Acrobat you can convert various types of files into .pdf format, including .jpg files.

To use .eps files (or .pdf files in T<sub>E</sub>XShop), you must first insert

```
\usepackage{graphics}
```

into your file prior to

```
\begin{document}
```

To insert an .eps file, use the format

```

\begin{figure}[ht]
\hfil\includegraphics{<Path+filename>}\hfil
\caption{<Caption>}\label{<Figure Label>}
\end{figure}

```

where the path+filename in Windows has the form

```
c:/mysubdir/mypic.eps
```

Notice that we use / instead of \ in T<sub>E</sub>X. In a Mac or Unix the path+filename has the form

```
/Users/username/myfolder/mypic.pdf
```

For T<sub>E</sub>XShop on a Mac, or any other implementation that used PDF<sub>La</sub>T<sub>E</sub>X, you use PDF files rather than EPS files.

The caption can be left blank if you do not wish to insert one (though you still need to write {} if you leave the caption blank). You should not write the figure number in the caption, since that is done automatically by L<sub>A</sub>T<sub>E</sub>X. To refer to the figure elsewhere in the text, use the format

```
Figure~\ref{<Figure Label>}
```

For .jpg files, when you use PCT<sub>E</sub>X, you must first insert

```
\input setbmp
```

into your file prior to

```
\begin{document}
```

To insert a .jpg file, use the format

```

\begin{figure}[ht]
\centerbmp{<Width>}{<Height>}{<Path+filename>}
\caption{<Caption>}\label{<Figure Label>}
\end{figure}

```

The width and height are lengths used to scale the figures, for example “3truein.” (Note that the width and height did not need to be stated for .eps and .pdf files, another reason that .eps and .pdf files are preferable.)

To view .jpg files in PCT<sub>E</sub>X, you must be in PS mode.

Here are some important formatting points to note when you use figures in a mathematics text:

- Each figure must be referred to by its figure number somewhere in the text. If a figure is not referred to, it will not necessarily be read.
- Each figure should be inserted **after** it is first referred to.
- Captions should be left blank, or should be minimal. Any explanation of a figure should be done in the text, not in the caption.

## 16 Bibliography

The best way to make a bibliography in L<sup>A</sup>T<sub>E</sub>X is to use the package called `amsrefs.sty` that solves this problem. This package, and a manual for the package, can be downloaded from <http://math.bard.edu/bloch/bardtex.htm>. Place the package in the same folder in your computer as your `.tex` file. To use this package, you must first insert

```
\usepackage{amsrefs}
```

into your file prior to

```
\begin{document}
```

The bibliography is put at the very end of the document, and looks like

```
\singlespace
```

```
\begin{bibdiv}
```

```
\begin{biblist}[\normalsize]
```

```
\addcontentsline{toc}{chapter}{Bibliography}
```

```
\markboth{Bibliography}{Bibliography}
```

```
\bib{G-K-P}{book}{
```

```
author = {Graham, Ronald},
```

```
author = {Knuth, Donald},
```

```
author = {Patashnik, Oren},
```

```
title = {Concrete Mathematics},
```

```
edition = {2},
```

```
publisher = {Addison-Wesley},
```

```
address = {Reading, MA},
```

```

date = {1994}
}

\bib{BA1}{article}{
author = {Banchoff, Thomas},
title = {Critical points and curvature for embedded polyhedra},
journal = {J. Diff. Geom.},
volume = {1},
date = {1967},
pages = {245--256}
}

\bib{SLOA}{report}{
author = {Sloane, Neal J. A.},
title = {The On-Line Encyclopedia of Integer Sequences},
eprint = {http://www.research.att.com/~njas/sequences},
}

\bib{WINT}{article}{
author = {Wintgen, P.},
title = {Normal cycle and integral curvature for polyhedra in
Riemannian manifolds},
book = {
series = {Colloq. Math. Soc. J\'anos Bolyai},
volume = {31},
title = {Differential Geometry},
editor = {So\'os, Gy.},
editor = {Szenthe, J.},
publisher = {North-Holland},
address = {Amsterdam},
date = {1982}
},
pages = {805--816}
}

\end{biblist}
\end{bibdiv}

```

Each new bibliographic item starts with the

`\bib{<Label>}{<Type of Item>}{`

where the `Label` is the label for the bibliographic item, and the `Type of Item` is one of `article`, `book`, `misc`, `report`, `thesis`. Some of the fields, such as `author` and `title` are necessary in every bibliographic entry, whereas others such as `series` and `editor` are optional. Note that if there are multiple authors, each one is listed separately, always with last name first. The order of the fields inside each bibliographic entry does not matter. Do not forget the commas between the fields. Also, unlike regular text, in the bibliographic entries  $\TeX$  is not bothered by a period that is not at the end of a sentence, and you do not need to do anything about such periods.

To refer to an item in the bibliography, use the formats

`\cite{<Label>}` or `\cite{<Label>}*{<Location>}`

where `Location` is something like “Chapter 3,” or “Theorem 1.2.3.”

# Sample Project Outline

```
\documentclass[11pt, oneside, reqno]{book}
\usepackage{amssymb, amsthm, amsmath, amsfonts}
\usepackage{bardproj}
\usepackage{graphics}
\usepackage{amsrefs}

\begin{document}

%For senior projects:
\titlepg{<Title of Project>}{<Your Name>}
      {<Month of Graduation>}{<Year of Graduation>}

%For M.A.T. mathematics research projects,
%uncomment the following, and remove the above:
%\titlepgmat{<Title of Project>}{<Your Name>}
%   {<Month of Graduation>}{<Year of Graduation>}

\abstr

<Text of abstract>

\tableofcontents

\listoffigures

\dedic

<Text of dedication>

\acknowl

<Text of acknowledgments>

\doublespace

\chapter{<Title of First Chapter>}
\label{chapA}
```

```

\section{<Title of First Section>}
\label{secA1}

<Text>

\section{<Title of Second Section>}
\label{secAB}

<Text>

\singlespace

\begin{bibdiv}
\begin{biblist}[\normalsize]

\addcontentsline{toc}{chapter}{Bibliography}
\markboth{Bibliography}{Bibliography}

\bib{<Label>}{book}{
author = {<Last Name, First Name>},
title = {<Title>},
publisher = {<Publisher>},
address = {<City>},
date = {<Year>}
}

\bib{<Label>}{article}{
author = {<Last Name, First Name>},
title = {<Title>},
journal = {<Journal Name>},
volume = {<Volume Number>},
date = {<Year>}
pages = {<Starting Page--Ending Page>}
}

\end{biblist}
\end{bibdiv}

\end{document}

```

