LaTeX and WinEdt, Week 2

Configuring YAP and WinEdt

When WinEdt does its installation it also installs the screen previewer YAP. YAP is a separate piece of software designed by Christian Schenk in Germany to view the DVI files that are created by TeX and LaTeX. You can find out more than you want to know about YAP at the site

```
http://math.ndsu.nodak.edu/resources/tex/bibliography/yap/yap.html
```

The items I will describe in the next section are some basic features that you probably do want to know and take advantage of.

Some YAP Configuration

Open a .tex file in WinEdt and run LaTeX on it. This will create two files, one with the suffix .log and one with the suffix .dvi. The .log file tells what LaTeX did when it ran, and if all went well we probably don't care, the .dvi file is the DeVice Independent file that we will open in YAP.

If you ran LaTeXby using the Brown Bear and all is well, you should already be in YAP. If you ran LaTeX by using the LaTeX icon you need to go down on the toolbar menu to the DVI Previewer icon, which is the rightmost of the two icons labeled DVI.

The first configuration we will make will permit us to click on a part of the DVI file in YAP and bring us in WinEdt to the place in the LaTeX file that corresponds to it.

- Go to the View item on the YAP toolbar and open Option.
- Open the Inverse DVI Search item
- In the main box you should see the item

  WinEdt (auto-detected)

  and this should be the highlighted choice. If it is not, choose it.
- The command line box should contain

  ```
  "C:\Program Files\WinEdt Team\WinEdt\winedt.exe" "[Open(|%f|);SelPar(%l,8)]"
  ```

In earlier versions this needed to be added to this Command Line box, but it appears that in Version 2.62639, the one we have installed, it is already set up, although perhaps not chosen as the default.

- Click OK to return to the main YAP menu.
- If you want to see the links that connect the DVI file in YAP and the LaTeX file in WinEdt, choose Tools from the YAP toolbar, and then Source Links. The Line numbers in the right column refer to the lines numbered consecutively in the file.

Thee are many other configurations you can do in YAP, but we will postpone these until later.

Some WinEdt Explanation

WinEdt is a very powerful editor that can be configured to do many tasks. I will illustrate a few of the things I use which should give you some ideas about how you might want to configure your own editor. First, I recommend that you go to Help and read the information in the Contents section. You might want to skip the portion **Taking advantage of WinEdt's capabilities...** under the **Setting Started, TeX: Step-by-Step** on your first reading. It is a bit technical. Do look at the last few lines in this section, however, since they tell you the packages to include in a document if you want to produce some complicated mathematical symbols. The last one on this list is not part of the standard MiKTeX installation so if you LaTeXa file with this command, MiKTeX will use the network to find and install the correct files.

WinEdt was originally designed only as a text editor for creating TeX and LaTeXfiles. In its current version it is also suitable as an editor for programming construction, for HTML file generation, and many other tasks. We, however, will restrict our consideration to the LaTeX usage.

The first thing to note is that in the Default configuration there are a number of colors and other features used for various purposes. For example,

- Normal text is shown in black.
- When the Spell mode is on, a word in text mode that WinEdt thinks is misspelled is shown in red, or red and blue, and underlined.
- Strange words like WinEdt, TeX, LaTeX, and YAP, that are known to WinEdt, are shown in yellow.
- LaTeX control sequences in text mode are shown in blue of various shades and intensities.
- Objects set in mathematics mode have a shaded background.
- LaTeX control sequences in math mode are shown in red of various shades and intensities.
- Commands within

```
\begin and \end
```

statements are underlined and in green.

- Items within

  ```
  \begin and \end
  ```

  statements are have a shaded background.

- `\documentclass`
  is set in dark green and within its braces the item is set in red, whereas the

  ```
  \usepackage
  ```

  items are set in dark green with their items set in blue.

It is not important to remember the specific colors, but to consider them as toggles. When you see a color change, be sure that something has happened that you expect, and when you expect something to happen, like going between text mode and mathematics mode, note that there has been some color change.

Notice there is a light blue arrow in a shaded background at the left margin of the WinEdt sheet that points to the line the cursor is currently on. Go to the shaded background and right click on it. You will see a menu with a number of items on it. The only one we will look at this time is the Show Line Numbers. Left click on this item and you will see the lines numbers of the items in the document. However, it doesn't number the actual lines, since for LaTeX it doesn't matter how many words are in a lines. It numbers items that it knows are distinct entities. These are the lines referred to in the YAP Inverse Search scheme. Notice also that blank lines are numbered as well.

Some WinEdt Configuration

WinEdt can be configured to do a lot of different things, and I will be introducing these piece-by-piece during the next few weeks. Today I will just give you some basic configuration options.

The Status Line

The line at the bottom of the page is called the Status Line.

- The question mark on the left links you to a help menu on the Status Line. Actually an unregistered copy of WinEdt displays the string ? ... in the first panel. Once you enter your registration this panel shrinks to ? and clicking on it brings up this help.

- The next box will show either A or B, and give you access to two different locations in your document. Suppose it reads A and you are at the last line of your document. If you click on A it will turn to B. Suppose that you now go the the first line of your document. Then in View A you will be at the last line, and in View B at the first line, until you change these views.

- The next box tells you the location of the cursor by actual line number and position from the left margin.

- The next box tells the number of actual lines in the document. Notice that by clicking on this box you will toggle from line number on to line numbers off. If you are in Wrap mode, toggling might change the number of lines in the document, since the document will be realigned because of the decreased left screen margin.

- Clicking on the next box will tell you when the document was last Modified, or state that it is a Read Only document.

- There are various modes of Wrapping that can be used in WinEdt. I will show how to configure this after we finish with the Status Line. This box will toggle the Wrapping feature on and off.

- I don't know what toggling Indent does, and Help doesn't help. I suspect this will become clear later.

- The next box permits you to toggle between Insert and Overwrite modes. You probably want Insert.

- The next box permits you to toggle between Line mode and Block mode. You will generally want Line mode but I will illustrate in class some instances where Block mode is useful.

- You can Toggle the Spell Checker on and off. You can also add your own spell dictionary to complement or replace the default dictionary.

- The next box tell what type of file mode your document is in. It will likely read TeX.

- The next two boxes may be blank. They are used to tell you the Date and Time. Here is how you can activate this feature. It is our first WinEdt configuration.

  - Go on the WinEdt toolbar to Options and then Appearances.
  - Go to Time and Date.
  - Put checks in the boxes you want to make active.
  - If you would line to display the items differently make the appropriate modifications. For example, changing the date line to ddd mmm-d-yyyy will interchange the date and the month in the display.
  - Click on the Date box in the Status Line to enter the date and time at the cursor location.
  - Click on the Time box in the Status Line to enter only the time at the cursor location.

- The next box turns the Inverse Search feature in YAP on and off. If you turn it off, the box will be blank. You should leave it on.

- The final box tells what the current Project is called. It will display WinEdt.prj until you change it by entering a new project. I will detail how to do this later.

Some Additional Configurations

There are many additional ways to configure WinEdt for you personal use. In this section I will give two of these that I find particularly useful.

Soft Wrapping in WinEdt

Wrapping of text is a handy and common feature of all word processing programs. It refers to automatically starting a new line when a line is filled with type. However, there are some problems with wrapping when using a system that has flexibility on character spacing. For example, you will likely want to set displayed equations on separate lines so that if there is an error in the setting it can be easily found. In this situation you will want the word processing program to honor the line break you gave it and not wrap the lines. A scheme of this type is known as soft wrapping. In WinEdt it does one additional thing. It never wraps when you are in mathematics mode.

I strongly recommend that you set the wrapping feature of WinEdt on and that you set up the wrapping to be in Soft Wrapping mode. You can do this as follows.

- Close all current TeX documents that are open in WinEdt.

- Go on the WinEdt Toolbar to **Options/Settings**, and in the Setting Menu open Wrapping.

- Be sure that the **Soft Wrapping for File Modes:** box is checked.

- If you want Soft Wrapping to apply to all documents, make sure that the line below the **Soft Wrapping for File Modes:** is empty.

- If you want Soft Wrapping to apply only to TeX documents, then be sure that only **TeX** is in the the line below the **Soft Wrapping for File Modes:** .

- Leave the other boxes as set in the Default settings unless you know what you are doing. See the WinEdt help document for complete details on wrapping.

- Click OK to return to the WinEdt window.

When you open a document be sure that the Wrap icon in the Status Line of WinEdt is on. Then soft wrapping should occur.

Configuring PDFTEXify to Include EPS Graphic Files

This section is optional, but contains some valuable information for those who want to create PDF files that include Encapsulated PostScript (EPS) Files. These are the industry standard graphics files, which can be created, for example, with Adobe Illustrated.

The TEXify option in WinEdt is a very convenient way to process LATEX files. Because there may be internal references in a LATEX document, it might be necessary to run LATEX on a document more than once so that these references are correct. For example, my Test-JDF.tex file needs LATEX to run twice since it automatically gives page numbers on the right top margin as **Page xx of yy Pages**. The first time LATEX is run, it determines the number of pages in the test. The second time LATEX is run it adds the correct page number headings to the pages.

The TEXify option (Little Brown Bear) detects how many times LATEX needs to be run and then runs it that many times. Following this, if all is well, it opens YAP and displays the DVI file. I have used this option when I have run LATEX in class, and suggested it for your own implementation, so by this time you should be familiar with the process, if not the exact details of its workings.

There is also an option called PDFLATEX that creates PDF files instead of DVI files, and then sends these to Adobe Acrobat for viewing. This is useful because PDF files are the industry and government standard, even the IRS files are PDFs. The procedure is the same as in LATEX it is just that the output is in the more widely used PDF format rather than the DVI format.

However, although I want PDF files as my final output, PDFLATEX would not work for me. My documents commonly include a lot of graphics, and professional graphics files are commonly in Adobe Encapsulated PostScript (EPS) format. This is an ASCII, that is, person-readable, format and these files are often quite large. They can be converted to the much smaller binary, that is, machine-readable, PDF format by using Adobe Acrobat software (which is not freeware). PDFLATEX did not recognize EPS graphic files, so for it to be useful all the EPS files had to first be converted to PDF files. Since my books might contain thousands of graphics files, this is not a minor problem.

I got around the problem by creating my own version of PDFLATEX that took a TeX file, created a DVI file, then created a PostScript (PS) files that included the graphic files, and finally used the Adobe Acrobat software to convert this to a PDF file. This worked, but I was never able to get it to include the multiple runs of LATEX that makes TEXify so useful. Clearly, it would be useful to have a PDFTEXify option as well that would incorporate the nice features of TEXify but create PDF files as output, and additionally permitted the inclusion of graphics files in EPS format.

I contacted the developer of WinEdt, Aleksander Simonic, and told him about my problem, and in this latest version of WinEdt he has included an option to solve this problem. You may never need to include EPS graphics files in your documents, but if you want to have the option of doing so, here is how.

First you need some (free) software, called **GhostScript** for creating PostScript files.

- Go to the internet site

`http://pages.cs.wisc.edu/~ghost/`

- Click on the line **GPL Ghostscript**.
- This should take you to the site

  `http://pages.cs.wisc.edu/~ghost/doc/GPL/index.htm`

- Go to the link

  `http://sourceforge.net/projects/ghostscript/`

- At that site you will find a green box that says **Download GhostScript**. Find the line that says GPL GhostScript 8.70 and click on it.
- Transfer the file to where ever you like to save download files and run the .EXE file to install GhostScript.
- Accept the default setting, and note that 40 meg of hard drive space is required.
- When the installation finishes there will be a window tell you where your personal information is located.
- You probably don't need to read the Readme file, just exit out of Explorer.

Now we need to do a minor configuration in WinEdt and we are done.

We need to change PDFTEXify, so go to

Options/Execution Mode/TeX Output

Then change PDFTEXify Output to the last entry, that is, to

```
tex -> ps -> pdf (dvips + Ghostscript)
```

and click OK.

Now if you run using the the Brown Bear with the Adobe symbol in the background it will

- Close the current PDF file if it is open.
- Run LATEX as many times as it needs in order to get all the references correct,
- Create the DVI file from the TeX file,
- Create a PS (PostScript) file from the DVI file,
- Convert the PS file to a PDF file, converting both text and graphics,
- Save the PDF file in the directory with the TEX file,
- Open the PDF file in Adobe Acrobat.

Entering Keystroke Macros

Suppose that I would like to make a definition to set the numerator and denominator of a fraction which when executed will automatically return the cursor to the place where I can insert the numerator, as in

```
\frac{x}{}
```

where the x indicates where the cursor will be placed. It is a little tricky at first, but after doing it once it is quite easy and very powerful. I can do this as follows:

- Open a New Document
- Go on the WinEdt Toolbar to **Macros/Recorder**
- This will give a split screen with its own toolbar, which I will call the **Recorder Toolbar**. If you run the cursor over this toolbar you will see the functions associated with the icons.
- Click the left, **Record**, button to record the keystrokes I will give you. The ASCII characters associated with what you type will appear in the lower window. If you make a mistake, click the **Reset** icon on the Recorder Toolbar and start over.
- It is difficult for me to show this but I have left the cursor between the first set of braces by typing the left arrow key three times after I typed the text.

  ```
  \frac{}{}
  ```

- In the Recorder window your lines should look like

  ```
  Chr(92,"\");
  Chr(102,"f");
  Chr(114,"r");
  Chr(97,"a");
  Chr(99,"c");
  Chr(123,"{");
  Chr(125,"}");
  Chr(123,"{");
  Chr(125,"}");
  Do("[IfScrollLock(!""ScrollLeft(1);"",!""CMD(’Char Left’);"");]");
  Do("[IfScrollLock(!""ScrollLeft(1);"",!""CMD(’Char Left’);"");]");
  Do("[IfScrollLock(!""ScrollLeft(1);"",!""CMD(’Char Left’);"");]");
  ```

  You can read these to see that they are really doing what I want them to do.

- Go to the Recorder Toolbar and click on **Save**. You will be given a menu with the suggestion that you should Save the file as Recorded Macro.edt. **Don't do it!** Instead, move up one level in the directory and save the macro in the Macro file as

  frac.edt

- It is expected that you will screw this up a couple of times, but keep after it! When you think you have it right, go to a blank line on your document. Then go to **Macro/Play Macro Once** on the WinEdt Toolbar. You should get the macro you created.
- Close the Recorder Toolbar since we don't need it any more.
- OK, you might say, but how does this make life easier for me if I have to go to a menu every time I want to run the bloody thing? It doesn't, you need the second half of the lesson.
- We will now associate the fraction macro with a keystroke. I will use the keystroke **Ctrl f**, but you can use whatever you like. You should check the keystroke you intend to use first to see if it does something you would like to do, because you will over write this operation with the new associated macro.
- First we need to know some standards that WinEdt, and many computer programs use. When the characters %B are seen in a directory listing, WinEdt knows that this stands for wherever the home directory of WinEdt lives. Similarly, when the characters %b are seen in a directory listing, WinEdt knows that this stands for wherever the home directory of your version of WinEdt. For example, The WinEdt program lives on this machine in

  C:\Program Files\WinEdt

  so when WinEdt sees %B in a directory listing it substitutes these characters.

  It is important to know this since I will need to tell WinEdt where the macro that I want to associate with **Ctrl f** lives and what its name is. It is

  %b\Macros\frac.edt

- It might be good to read that last item over again to be sure it makes sense, then we are ready to go.
- On the WinEdt Toolbar go to **Options/Menu Setup**, and when the window appears left click twice on the **Shortcuts** option. This will give you list of already-defined shortcuts.
- Click on the second icon from the left, the **Insert New Item** option. Choose **Macro** from its list. You will be placed in a blank area entitled **Menu Items**
- Type

  frac

  in the blank area

- Go to the window on the right and type (very accurately)

```
Exe("%b\Macros\frac.edt");
```

This tells Winedt to execute the macro that is described.

- Go down to the Shortcut and hit on the keyboard

  Ctrl f

- Hit Ok to get to the Menu Setup window and the Ok again to get back to the document.
- Try it, that is, keystroke **Ctrl f**   It worked for me.

Your Turn

Make the following Keystroke Macros.

1. - Make a macro for

     ```
     \text{}
     ```

     and leave the cursor between the braces.
   - Save the Macro as text.edt in your Macros directory
   - Associate **Ctrl Q** with this Macro.
2. - Make a macro for

     ```
     \text{ \quad }
     ```

     and leave the cursor after the left brace.
   - Save the Macro as quadtext.edt in your Macros directory
   - Associate the keystroke **Crtl Alt Q** with this Macro.