Appendix B. Introduction to the X Window System

Pigi uses the X window system, version 11 release 6 (X11R6). It will also run correctly under X11R4, X11R5 and under Sun OpenWindows. This appendix provides just enough information about using X so that if you are not familiar with it you will nonetheless be able to get started. Complete documentation, however would be helpful. Complete guides to the X window system are available in many technical bookstores. If you are familiar with X, and your user account is properly configured to run X11, you need not be concerned about this appendix. `Pigi` will automatically load the resources it needs for operation. If you wish to customize these resources, it may be useful to scan this section. Resources are explained below.

B.1 A model user's home directory

A strength (and weakness) of the X window system is that it is infinitely configurable. Making reasonable use of it requires a number of files in the user's home directory. Usually, these files are hidden by giving them a name that begins with a period (`.`), such as `.xinitrc`. You can see these files by listing your directory contents with the `-a` option:

```
ls -a
```

Even if you are new to Unix systems, with a new account, your system administrator will have put a number of such hidden files in your home directory when he or she sets up your account. Hopefully, these files have been chosen to give you reasonable behavior immediately. If not, however, this appendix can help you set up your account for running Ptolemy.

The home directory of your Ptolemy installation (either `~ptolemy` or `$PTOLEMY`) is designed to serve as the home directory of a model user. We will refer to the directory by the name `$PTOLEMY`. Here is a listing of the contents of this directory:

```
% ls -a
./              .cshrc          .twmrc
../             .login          .xsession
.Xdefaults      .plan
```

This user is configured according to the tastes of the authors of Ptolemy. To configure your own account to behave the same way, you can copy all the dot files (hidden files whose names begin with a period) from the directory `$PTOLEMY` into your home directory. Unless you are a new user, however, you will probably not want to overwrite features already defined in your own dot files. In this case, you will need to extract the features you desire from the dot files in `$PTOLEMY` and add them to your own. To make this easy to do, we explain below the salient features of each of these files.

B.2 Running Pigi using Sun's OpenWindows system

Pigi will run under Sun's OpenWindows Version 2 or 3, assuming the Athena widgets have been installed (see the installation instructions). We assume as a starting point that you are already configured to run OpenWindows; *do* not attempt to use the `.login` and `.cshrc` in the ptolemy account in this case. First, make sure that your .cshrc file includes `$PTOLEMY/bin` in your path.

You will need to have the `olwm` window manager run in the mode where focus follows the mouse (in this mode, your keystrokes go to whatever window your mouse points to); to do this, find the line in your `.xinitrc` file that invokes `olwm`; add the option `-follow` to the `olwm` command. If, for some reason, you must run `olwm` in the click-to-focus mode, you must click the mouse in each window before pigi will accept keyboard commands in that window; it is possible but annoying to use pigi this way. An alternative is to add the following line to your `.Xdefaults` file:

```
        OpenWindows.SetInput:followmouse
```

B.3 Starting X

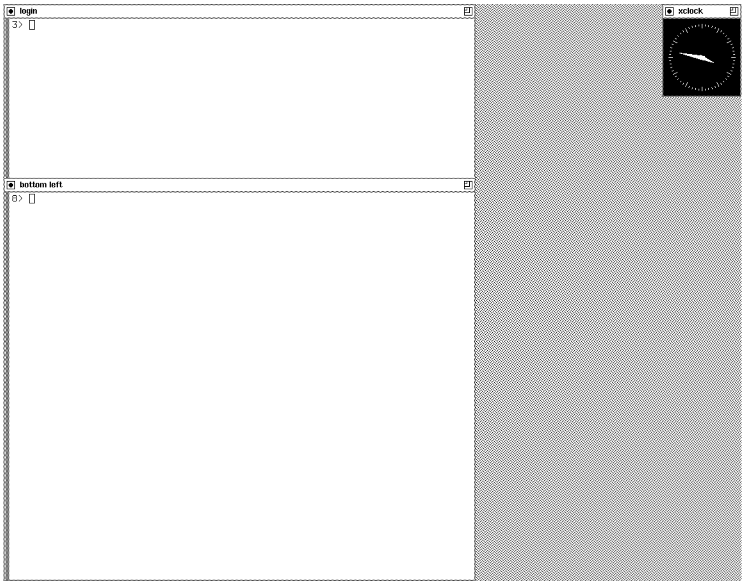
There are two principal ways that X may be configured:

- The X server may be running all the time. In this configuration, when no one is logged in a single window together with a background appears. You don't have to worry about starting X. A daemon program called *xdm* is responsible for logging you in.
- The user may be responsible for starting X — in this case, when you log in you get a bare workstation tube.

Our sample configuration attempts to support both to some degree. If `xdm` is running, it will use the supplied `.xsession` file to set up windows. If `xdm` is not running, and the user is on a bare Sun tube, the `xinit` program will execute, starting the X server; the `.xinitrc` file is used in this case to configure windows. The supplied `.xinitrc` file reads:

```
        xrdb $HOME/.Xresources
        twm&
        xterm -geometry 80x30+0-0 -fn 9x15 -name "bottom left" &
        xclock -geometry 120x120-0+0 &
        exec xterm -geometry +0+0 -fn 9x15 -name "login" -ls
```

The first line reads X resources from the file `~/.Xresources`. Then the window manager called `twm` is started. The next three lines open windows using the `xterm` and `xclock` commands. The `.xsession` file works pretty much the same way for xdm installations. The screen will look something like this after start up:



The function of the mouse buttons after the windows have opened is determined by the `$PTOLEMY/.twmrc` file. This file is reasonably easy to interpret, even without any X experience, so you may wish to examine it and customize it. If your installation uses the X window system in some way that differs from our default, then whoever installed Ptolemy should have modified the `.login` file above to reflect this.

B.4 Manipulating Windows

If you already know how to use the X Window system and you are using your own window manager configuration rather than the standard Ptolemy configuration, you may skip this section. Assuming you use the file `$PTOLEMY/.twmrc` without modification, rather than your own window manager, the basic window manipulations are explained below. Note that there are often several different ways to accomplish the same objective. Feel free to experiment with key and mouse button combinations in various parts of the screen. First, you must identify the “meta” key on your keyboard. It may be labeled in any of various ways, including “meta”, “left”, “right”, “alt”, a small diamond, or any of a number of other possibilities. It is usually close to the shift and control keys. Most window manipulations require that you hold down the meta key while depressing mouse buttons.

Iconifying windows.

Depressing the meta key, and clicking the left mouse button in any window will iconify it. The window disappears, replaced by a symbol in the icon manager at the right of the screen. To get the window back, place the cursor in the appropriate slot of the icon manager, and click the left mouse button. An alternative iconifying mechanism is to click any mouse button on the icon symbol at the left of the window header.

Moving windows.

Holding the meta key and dragging with the middle mouse button will move a window. “Dragging” simply means to hold the mouse button while moving the mouse. Alternatively, you can drag the left button in the middle of the window header.

Resizing windows.

The meta key and right mouse button can be used to resize a window. Place the mouse cursor near a corner of a window, depress the meta key, and drag the right button. Without the meta key, any button in the rightmost corner of the window header will resize the window.

Mnemonic.

The window header has an iconify icon at the left, a blue bar for moving the window in the middle, and a resize symbol at the right. Correspondingly, without going to the window header, but using the meta key, the left mouse button will iconify a window, the middle button will move it, and the right button will resize it. Hence, the window header can be used as a mnemonic to help remember which mouse button has which function when the meta key is depressed.

Pick up and stuff.

In a text window, without the meta key, the mouse may be used to grab text and put it somewhere else, either in the same window, or in some other windows. This can be very useful to avoid copying long sections of text. The left mouse button, when depressed and dragged, highlights a region of text in a window, and stores the text. The right button can be used to modify the extent of the highlighted region. The highlighted text is "picked up", stored for future use. The middle button causes the highlighted text to be typed to whatever window has the mouse cursor.

New windows.

A useful command, defined in `$PTOLEMY/.cshrc`, is “term”. Typing this in any window, followed by a carriage return, opens a new terminal window with a reasonable size and color. You may change the size and location in the usual way.

Removing windows.

Most windows, including all pigi windows, can be removed by typing control-D with the mouse cursor in the window.

Resources.

A *resource* in X is a parameter that customizes the behavior of applications that run under X. For instance, resources determine the foreground and background color of your windows. Pigi requires that some resources be set. These resources are defined in the file `$PTOLEMY/lib/pigiXRes`, and are automatically merged with whatever other resources you may have defined. The merging occurs when you invoke the start-up script `$PTOLEMY/bin/pigi`. In addition to these required resources, there are many optional resources. Some of these are set in `$PTOLEMY/.Xresources`.

If you have not used the X window system before, you will probably not have a file with the name `.Xresources` in your home directory, and can simply copy the one from `$PTOLEMY`. This file defines some basic resources. If you already have a file with this name, then you can probably use the one you have as is.