Appendix A. Installation and Troubleshooting

Authors: *Joseph T. Buck*
Alan Kamas

Other Contributors: *Christopher Hylands*
Edward A. Lee
Kennard White

A.1 Obtaining Ptolemy

Ptolemy is currently available for Sun 4 (Sparc), DecStation (MIPS), and HP (Snake) architectures. Installing the system requires 90 Megabytes of disk space for Ptolemy (more if you optionally remake). Ptolemy also requires at least 8 Megabytes of physical memory. The latest information about Ptolemy can be found using the command:

```
finger ptolemy@eecs.berkeley.edu
```

You may also send any questions on obtaining Ptolemy to the same email address: "**`ptolemy@eecs.berkeley.edu`**". If you have questions about using or enhancing Ptolemy, you should use the ptolemy mailing list. See the below for details.

A.1.1 Access via the Internet

Ptolemy is available *without support* at no charge to organizations on the Internet via FTP. Source code, executable, and documentation are all available at the FTP site. The procedure for obtaining Ptolemy via FTP is as follows:

a. FTP to Internet host "**`ptolemy.Berkeley.EDU`**".

b. Login as "anonymous"; give your full email address as the password.

c. cd pub

d. Follow the directions in the "README" file there.

A.1.2 Access via ordinary mail

Organizations without Internet FTP capability can obtain Ptolemy *without support* from:

EECS/ERL Industrial Liaison Program Office
Software Distribution
205 Cory Hall University of California at Berkeley
Berkeley, CA 94720
(510) 643-6687
email: ilpsoftware@eecs.berkeley.edu

There is currently a handling fee of $250 for US orders, $265 for foreign. These prices are subject to change at any time. Please contact the ILP office at the address above for the exact price. Printed documentation is included.

A.1.3 Obtaining documentation only

The printed documentation is available separately from the tape. Please contact the ILP office at the address above for the current price.

A.2 Ptolemy mailing lists

There are two mailing lists for Ptolemy users:

`ptolemy-hackers@ohm.eecs.berkeley.edu`

`ptolemy-interest@ohm.eecs.berkeley.edu`

Both mailing lists follow the same administrative procedure. To be added or deleted from a mailing list, send a letter with the subject **`ADD ME`** or **`DELETE ME`** to *MaillingListName*–**`request@ohm.berkeley.edu`**. For example, to be added to mailing list "ptolemy-hackers," send email with the subject ADD ME to "ptolemy-hackers-request@ohm.berkeley.edu"

- **`ptolemy-hackers`**: A forum for Ptolemy questions and issues. Users of the current release who have a Ptolemy question, comment, or think they've found a bug should send mail to ptolemy-hackers. Ptolemy users in companies and universities around the world who are interested in discussing Ptolemy post and read from the ptolemy-hackers group.
- **`ptolemy-interest`**: A mailing list for Ptolemy announcements. Messages about new releases, bug fixes, and other information of interest to all Ptolemy users is posted here. Note: This is not a discussion group, and you cannot post to this mailing list. Message volume is very light. All users of Ptolemy should belong to this group in order to stay current with work being done on Ptolemy. To be added to the group, send mail with the subject **`ADD ME`** to: "ptolemy-interest-request@ohm.berkeley.edu".

A.3 Installation for Ptiny Ptolemy

Ptolemy is a large software system that relies on a properly configured software environment. There are a few things that can go wrong in getting Ptolemy running. There is some minimal configuration required by each user in order to use the graphical interface. The information for doing this is given in “Setup” on page 3-1. Here we give the basic information required to get from an FTP archive or distribution tape to being able to run the system.

Note that this information may change after this manual is printed. Thus, the most up-to-date instructions are included in ASCII form with the distribution.

A.3.1 Basic Ptolemy installation

Note, in order to install Ptiny Ptolemy you will need 12 Megabytes of disk space.

First change directories to the place you will put Ptolemy. If you have the tar files from the FTP site, move them to this directory. If you have Ptolemy on tape, do the following commands (change the device name if your tape drive has a different name):

a. **`mt -f /dev/nrst8 rewind`**

b. **`tar xf /dev/nrst8`**

Now, there should be two tar files in your current directory.

The Ptiny Ptolemy tar file was compressed with gzip. Thus, you will need the program gzcat to uncompress it. If you do not have gzcat on your system, don’t worry. We have provided a gzcat binary with the Ptiny Ptolemy distribution. Do the following:

a. **`zcat gzcat.ARCH.tar.Z | tar xf -`**

Where **`ARCH`** is **`sun4`** if you have Sun Sparc machine, **`mips`** if you have a DEC Station, and **`hppa`** if you have an HP precision architecture machine.

Now that you have gzcat, you can use it to untar Ptiny Ptolemy:

b. If you are running the DEC Station (MIPS) version of Ptolemy:
`gzcat ptiny0.5.mips.tar.gz | tar xf -`
If you are running the Sun 4 (Sparc, IPC, IPX, etc.) version:
`gzcat ptiny0.5.sun4.tar.gz | tar xf -`
If you are running the HP Precision Architecture (HP PA) version:
`gzcat ptiny0.5.hppa.tar.gz | tar xf -`

Note that is possible to install binaries for multiple platforms on the same file system because different directories are used for each set of binaries. Just execute whichever of the above commands apply.

A.3.2 Special considerations for use under Open Windows

Ptolemy was developed using the X11R4 or X11R5 distributions from MIT. Although Ptolemy runs fine under Open Windows 2, there are problems with running the Ptolemy graphical interface with Open Windows version 3.0. Some users have had no problems at all, but others have had intermittent problems such as “bad match” errors. We believe this may be a problem with the X-server supplied with the Open Windows 3.0, but the error is elusive and we have not yet tracked it down.

In order for all utilities included with this distribution to work under Open Windows (either version), you must install the shared libraries for the Athena widgets (the freely redistributable widget set from the MIT X11 distribution), which are provided with this distribution under the **ptiny/athena.sun4** directory. To install them, become root and copy all files in that directory into **/usr/openwin/lib** (or, if you have installed Open Windows in a non-standard place, into **$OPENWINHOME/lib**). If you do not wish to do this, you could leave them in place and have every Ptolemy user change their **LD_LIBRARY_PATH** environment variable to search the **ptiny/athena.sun4** directory before **/usr/openwin/lib**. Consult the Unix manual entry for the **ld** program to learn more about **LD_LIBRARY_PATH**.

After installation, the ptolemy account will contain several scripts for starting up X11R5 (Xrun), Open Windows with *olwm* (Xrun.ow), or Open Windows with *twm* (Xrun.ow.twm).

Some users have reported problems with running **xedit** under Open Windows after setting their **LD_LIBRARY_PATH**. One workaround is to set the **PT_DISPLAY** environment variable to use an editor other than **xedit**, such as **vi**.

A.3.3 Testing the Installation

After starting up X or openwin, do the following:

a. **setenv PTOLEMY your_new_ptiny_dir**
   where your_new_ptiny_dir is the new ptiny directory that you created when you untarred Ptiny Ptolemy.

b. **cd $PTOLEMY**

c. **set path = ($PTOLEMY/bin $path)**

d. **cd demo**

e. **pigi**

Follow instructions in “Starting Ptolemy” on page 3-2.

The dot files (**.cshrc**, **.login** etc.) in the home directory are set up according to the tastes of the Ptolemy authors, and according the standard use of windowing software in the Ptolemy development group. You may want to use these files as guide as you customize your own installation.

A.4 Troubleshooting

This section lists common difficulties encountered when Ptolemy is installed or run. This list is, of course, by no means complete. If you do not find your particular problem here, refer to “Additional Resources” section below.

A.4.1 pigi fails to start when put in the background

A common problem occurs when **pigi** is started in the background and the user has the line

```
stty tostop
```

in their **.login** or **.cshrc** file. This command configures the terminal to halt any process that is running in the background when it tries to write to the terminal. One possible fix is to run **pigi** in the foreground. Another possible fix is to eliminate this command from your **.login** file.

A.4.2 pigi fails to start with a message about not finding fonts

The default fonts for **vem** are specified in the file **$PTOLEMY/lib/pigiXRes9**, and also **pigiXRes9.bw** and **pigiXRes9.cp**. These files define a set of X window resources. The **pigiXRes9.bw** file is used if **pigi** is started with the **-bw** option. The **.pigiXRes9.cp** file is used in **pigi** is started with the **-cp** option. The definitions in these files can be overridden by the user. For example, a user who prefers to use microscopic fonts could set the X resource as follows:

```
Vem*font:  *-times-medium-r-normal--*-120-*
```

If, however, the fonts defined in these files on not available on the system, then the Ptolemy installer should change them in the files $PTOLEMY/lib/pigiXRes9*.

The fonts for tk (and hence, the fonts for most of the dialog boxes) are specified in **$PTOLEMY/lib/tcl/ptkOptions.tcl**. These may similarly require changing at some sites. In the worst case, if many standard fonts are not available, it may be necessary to redefine the default fonts built into the tk source code, and recompile tk.

A.4.3 pxgraph fails to come up or displays a blank window

If the **pxgraph** program is given exceptional numbers, such as the IEEE floating-point **Inf**, **-Inf**, or **NaN** (“not a number”), then it will issue a cryptic error message “problems with input data” and will fail to display a plot. The stars that use **pxgraph** are supposed to intercept this and pop up an error message in a window. However, on some platforms, as of this writing, this does not work. On such platforms, the error message, unfortunately, goes to the standard output, which may be buried several layers deep in your windowing system. It is also possible for the standard output to be lost, so that no error message appears. Thus, if you get such a **pxgraph** failure, look for instabilities in your Ptolemy schematic that would cause it to produce such exceptional numbers.

A.4.4 Old flowgraphs do not work (facets are inconsistent)

A **pigi** schematic contains references to icons. These icons are referenced by their location in the file system, typically using either an absolute path, a path relative to user’s home directory, or a path relative to the environment variable **PTOLEMY**. If the master for any of these icons is not in the expected place, **vem** will issue a warning, telling you that the facet is inconsistent. In place of the icon in the schematic there will be blank space. To find out what icon masters are missing, run the program **masters**. Instructions for doing this are given in "Copying and moving designs" on page 3-43. Invalid masters will be labeled "**INVALID**". You must replace the invalid reference with a reference to a valid master.

One typical scenario for users upgrading from version 0.4.1 of Ptolemy to version 0.5 is that they will have references to masters in the directory tree **~ptolemy**. But Ptolemy 0.5 may be installed somewhere else. One solution is to use the masters program to replace "**~ptolemy**" with "**$PTOLEMY**".

If you have flowgraphs dating back to the 0.3.1 version, you will encounter some significant incompatibilities. In particular, several stars that were included in the 0.3.1 release of Ptolemy have been renamed. The objective was to adopt a consistent naming policy across all domains. One major benefit of this is the ability to migrate universes and galaxies from one domain to another. A consequence, however, is that some of your universes and galaxies will have inconsistencies (they will point to icons that no longer exist in the current release). Please see the file "$PTOLEMY/lib/rename/README." This file explains how to upgrade your universes and galaxies to the current release.

A.4.5 Error: ld.so: libXext.so.4: not found

You have not installed the shared library needed by Ptolemy when it is used under Open Windows. See the "Open Windows" section in the installation guide.

A.4.6 Problems starting up clients such as xedit under Open Windows

Use the **PT_DISPLAY** variable to use a different editor. See the "Environment Variables" section in the Interactive Graphical Interface chapter of the User's Manual.

A.4.7 Problems with the colormap

Some applications, for example FrameMaker 4.0, allocate as many colors as they can from the colormap when they start up. This may force applications that are started later (such as **pigi**) to use a very restricted set of colors. When you start **pigi**, if the welcome window appears in black and white, then you may have such a situation. If the situation is worse, and there are not enough colors in the colormap for **vem** to start, then you may not even get this far. One solution is simply to exit the offending application (e.g. FrameMaker), and start **pigi** over. A better solution is to configure the offending application to use fewer slots in the colormap. We have found that for FrameMaker 4.0, the following X resources (placed in your **.Xdefaults** file) usually solve the problem:

```
Maker.targetExactColors: 2
Maker.minimumExactColors: 0
Maker.targetColorCube: 4
Maker.minimumColorCube: 1
```

This still leaves FrameMaker with a very rich set of colors to use. You may need to replace the 2 or 4 with smaller numbers if you have other color-intensive applications running (such as root window pictures of beaches in Tahiti).

A.4.8 Ptolemy simulation runs do not stop

In the SDF domain, particularly, it is possible to have multirate systems where a single iteration is very large number of firings of stars. This happens when the number of samples produced or consumed by various connected stars in the system are mutually prime, or they have very large least common multiples. If a simulation is taking an unreasonable amount of time, then look for such mutually prime numbers. Sometimes, in such circumstances, it can take a long time for the simulation to respond to pushing the "stop" button. It should, however, eventually respond.

A.4.9 Generated code in CGC fails to compile

The targets in CGC are configured by default with reasonable guesses about the compile and link options that are required to compile the code. However, the actual options required depend on your system configuration. For instance, your default **`cc`** compiler may not have been configured to automatically find the X11 include files. You might, therefore, get an error message the Xlib.h cannot be found. You should find out where on your system Xlib.h is installed, and add a compile option of the form "-L*path_name*" where *path_name* is the full path of the directory containing the file. Alternatively, you can create a symbolic link from /usr/include to the directory (you must have root permission to do this).

Certain compiles will change their behavior depending on the values of certain environment variables:

- The hppa cc compiler will use the **`CCOPTS`** environment variable. If your X11 libraries were at /usr/sww/X11R5/lib, then one could exit pigi, set the variable
  `setenv CCOPTS -L/usr/sww/X11R5/lib`
  and restart pigi. Then when you compile CGC demos with hppa cc, you should be able to find the proper libraries. See the hppa cc man page for more information.
- GNU gcc will also use certain environment variables. The **`LIBRARY_PATH`** variable may help:
  `setenv LIBRARY_PATH /usr/sww/X11R5/lib`
  See the gcc documentation for more information.

A.4.10 Ptolemy won't recompile

You may be using a substantially different version of the Gnu compiler. The system is most likely to build if you use the same tools that we used originally. The Gnu tools we used are supplied with the Ptolemy distribution.

A.4.11 Missing symbols while linking pigiRpc

On the sun4 with libg++-2.5.2, if you compile pigiRpc with the g++ **`-O`** option, then you may have missing symbols while linking **`pigiRpc`**. The workaround is to upgrade to libg++-2.5.3. We shipped libg++-2.5.2 with the 0.5beta, but we are shipping libg++-2.5.3 with the final 0.5 release.

A.4.12 Error involving xedit

The program **`xedit`** is a program provided with Ptolemy to display files on your screen during a Ptolemy run. If there is some problem with **`xedit`** that you cannot solve, or you do not like **xedit**, it is possible to change the display program used by Ptolemy. To do this, you need to set the **PT_DISPLAY** environment variable before you run **pigi**. For example, to use **PT_DISPLAY** to view files in a new xterm window with the *vi* editor, do

```
setenv PT_DISPLAY "xterm -e vi %s"
```

The value of the **PT_DISPLAY** variable is a printf format string with one %s in it. That %s is replaced with the file to be viewed, for look-inside commands and to view generated code. The default value is **"xedit %s"**.

A.4.13 Emacs confuses .pl files with Prolog

The .pl extension used to define Ptolemy stars is the same extension used for the Prolog language. Some text editors, such as emacs, have special modes for editing Prolog files. These modes are inappropriate for editing Ptolemy files. You can create the following line your .emacs file in your home directory:

```
(setq auto-mode-alist (cons '("\\.pl$" . c++-mode) auto-mode-alist))
```

A.4.14 The window manager crashes

The window manager **twm** sometimes crashes when you are running Ptolemy. We do not know why. Our solution is to simply restart it. You may wish to make sure your configuration does not log you out when the window manager exits.

A.4.15 Path and/or Environment Variables not set in “debug” pigi

When running Ptolemy’s interactive graphical interface with the debug option:

```
pigi -debug
```

The path is not set correctly, or environment variables are not at their normal values. This is caused by the Gnu debugger, GDB, overwriting values set by the pigi start-up script. From the GDB manual:

```
*Warning:* GDB runs your program using the shell indicated by your
`SHELL' environment variable if it exists (or `/bin/sh' if not). If
your `SHELL' variable names a shell that runs an initialization file-
-such as `.cshrc' for C-shell, or `.bashrc' for BASH--any variables
you set in that file affect your program. You may wish to move setting
of environment variables to files that are only run when you sign on,
such as `.login' or `.profile'.
```

If your .cshrc file specifies a value for a variable, it will override anything in the pigi start-up script. If this is the case, perhaps setting SHELL to /bin/sh before firing off the debugger will fix the problem.

A.4.16 CGC demos won’t run

If **.** (dot, the current working directory) is not in your path, then when you run the CGC demos you may see:

```
sh: butterfly: not found
```

The workaround is to exit **pigi**, place **.** in your path, and restart **pigi**. For security reasons, placing **.** in your path after all of your other directories is best:

```
set path = ($path .)
```

A.4.17 EOF messages while using tar on Suns

There is a bug in the SunOS4.1.3 version of /bin/tar. Sometimes a command such as:

```
zcat atarfile.tar.Z | /bin/tar -xf -
```

may produce error message such as:

```
tar: read error: unexpected EOF
```

when reading from a pipe. One workaround is to use GNU tar, another is to uncompress the tar file, and then use /bin/tar on the uncompressed file:

```
uncompress atarfile.tar.Z; /bin/tar -xf atarfile.tar
```

A.4.18 Dynamic Linking fails

Ptolemy has the ability to load stars dynamically during run time. The stars are compiled into .o files and loaded with the Unix loader. Dynamic linking is tricky and dependent on the Unix loader. There are several reasons dynamic linking can fail:

- A tar overlay is available that allows dynamic linking on the hppa and hppa.cfront platforms. The tar overlay is available via anonymous ftp from ptolemy.eecs.berkeley.edu in the pub/ptolemy/ptolemy0.5/patches directory. See the README file in that directory for more information.
- If you are upgrading from an earlier release, including Ptolemy-0.5beta, be sure to remove all the .o files in the directory where the source files for you star is located.
- If you are using prebuilt Ptolemy binaries, be sure that you are using the prebuilt Gnu compiler that is also available with the Ptolemy binaries. The alternative is to rebuild Ptolemy with your local Gnu compiler, but be aware that versions earlier than gcc-2.5.6 and libg++-2.5.3 have bugs. gcc-2.4.x and libg++-2.4.x and earlier are know to have serious bugs, so you may want to upgrade. For more information, see the Gnu Installation section in this Installation document.
- If you are using prebuilt Ptolemy binaries and have the prebuilt Gnu compiler, be sure that either the Ptolemy distribution is available as /users/ptolemy, or you are setting the Gnu environment variables in $PTOLEMY/bin/g++-setup. Again, see the Gnu Installation section.
- If, at linktime, you see messages about undefined symbols, and the undefined symbols begin with _vt, then you probably have compiler version incompatibilities. Note that you can find out what files have the undefined symbols by using the Unix nm command. For example on Suns, the command `nm -o $PTOLEMY/lib.sun4/* | grep MySymbol` will find all the files that have symbols that contain the string MySymbol.
- Dynamic linking may not work on machines that have a different release of the OS. In particular, we found that dynamic linking with the prebuilt binaries failed on mips running Ultrix4.3 if our mips prebuilt binaries were created under Ultrix4.3A. The workaround is to relink Ptolemy on the local machine.
- If you are having problems compiling the star from Ptolemy, try running the compile by hand from the shell. Unfortunately, part of the compile command is not always visible in the vem window. To see the all of the compile command, try running a few vem

commands, such as ‘i’ (look-inside) to flush the vem buffer. Once you have produced a .o file, you can load the .o file into Ptolemy with the load-star command.

- If you can’t compile your star from the shell, try compiling a simple c++ program to verify that the compiler is working. Place the code below in a file called hello.cc and if you are using the Gnu compiler, try compiling it with **g++ -v**. The -v option will show the compiler steps.

```
#include <stream.h>
main()
{
  cout << "Hello, Ptolemy.\n";
}
```

A.4.19 The DE stars “And” and “Or” no longer work

These stars have been replaced by the Logic star. Use this new star instead.

A.4.20 Multi-porthole Galaxies Fail

If a galaxy contains a multi-porthole input or output, and the icon of the galaxy is named Foo.input=2, Foo.output=2, etc., the galaxy will fail to compile. This is because Ptolemy behaves as if anything ending in input=X or output=X must be a star. Avoid using names like “Foo.input=3” for galaxies.

A.4.21 Problems with Mac X and Ptolemy

Some people have had difficulties running Ptolemy with Mac X. Tze-Wo Leung of Bell-Northern Research suggests the following setup for Mac X when using Ptolemy:

- My hardware setup is: MacIIci w/ 20 MBytes DRAM, 210 MB HD and 21” two page display/21gs Radius B&W monitor. The UNIX server is a HP715 workstation.
- The ‘Display:’ option in the ‘Edit Remote Command’ MUST be set to ‘(2) Color Rootless’ mode.
- Increasing the memory allocation of MacX to 4 MBytes also helps. The memory size can be increased by first exiting MacX, then using the ‘getinfo’ option in the System 7 pull-in menu to pop-up the file info window where one can change the memory allocation.
- In the control panel, the characteristics of the monitor are set to ‘Colors’ and ‘256’.

A.4.22 Ptolemy Ports to other platforms

Ptolemy has been ported to a few other platforms. For the most part, these new ports are somewhat untested.

- Patches for a Linux port are available via anonymous ftp from ptolemy.eecs.berkeley.edu in pub/ptolemy/ptolemy0.5/contrib. The Ptolemy group at UC Berkeley does not have access to a Linux machine, so this port has not been tested by us.
- A Silicon Graphics port is in the works. Ptolemy was working under IRIX4.0.5H on the SGI, but we have since upgraded to IRIX5.2. Once the port is done, an announcement will be made.

- A DEC Alpha port has been considered, but there is no Gnu g++ compiler on this platform yet. Also, the Ptolemy group does not have direct access to an Alpha machine. Some work was done with translating the mips binaries to Alpha binaries with the **mx** program, but this work is incomplete.

A.4.23 Postscript error while printing Ptolemy manual files

Postscript gives a dictfull error, apparently inside one of the EPS illustrations. This problem has been reported mainly with the CG56 and CG96 chapters of the User's Manual when printed on older postscript printers.

Apparently, the Frame document header doesn't allow for a lot of def's inside EPS illustrations. If you use a PS implementation that won't expand dictionaries automatically, the dictfull error will result. Fortunately, there is a simple workaround because Frame has put in a whole lot of def's that don't do anything useful. If you are having this problem, find the lines:

```
/FMBEGINEPSF {
      end
      /FMEPSF save def
      /showpage {} def
% See Adobe's "PostScript Language Reference Manual,
% 2ndEdtion",page714.
% "...the following operators MUST NOT be used in an EPS file:"
      /banddevice {(banddevice) FMBADEPSF} def
      /clear {(clear) FMBADEPSF} def
      ... lots more like this ...
      /b5 {(b5) FMBADEPSF} def
      FMNORMALIZEGRAPHICS
```

The lines above begin about line 770 in each *.ps file. Delete the roughly forty lines containing "FMBADEPSF". Thanks to Tom Lane for finding and solving this problem.

A.5 Known Bugs

There are a number of known bugs that we have not had a chance to fix.

A.5.1 Bugs in Vem

- When graphical animation is turned on and a simulation is running, if you execute the "look-inside" command, **vem** terminates the pigiRpc application.
- Labels that end with a carriage return are neither displayed nor printed correctly.
- Editing icons stimulates a memory leak in vem that can make the process grow quite big. After editing a few dozen icons, you may wish to exit **vem** and restart.
- When you edit a label, a dialog box frequently appears with the message "ambiguous layer" and a checklist of colors. Check and color and continue. The choice of color has no effect.
- The "layer" command only works when editing icons, not when editing schematics.
- The "copy-objects" command will copy from one window to another only when editing icons, not when editing schematics.
- If you obtain the **other.src** tar files and attempt to build **vem** from scratch, do not

use **`gcc`**, use **`cc`** instead. **`Vem`** binaries built with **`gcc`** may fail to start up, or may crash later with memory errors.

A.5.2 Bugs in pigi

- The “run-all-demos” command does not always work. In particular, it fails on most Tcl/Tk demos, because these assume that a run control panel is present, but the run-all-demos command does not open a control panel. Also, some demos deliberately trigger an error, in order to demonstrate some failure modes in the domain. The run-all-demos command quits when it encounters such an error.
- It is possible to make more than one icon to represent a given star. For instance, the icon And and And.input=2 refer to the same star, but the former can have any number of inputs, while the latter has exactly two inputs. As of this writing, this facility does not work for galaxies. You should create only one icon for each galaxy.
- Simulations that run for a very long time may not respond quickly enough to pushing the stop button in the control panel.
- If you try to run the CGC demos that generate sound, and you do not have a properly configured audio device, or you do not have write permission to /dev/audio, then you get a cryptic and uninformative error message, rather than something useful.
- The CGC Multi-processor demos have Berkeley machine names hardwired into them. These demos probably won’t work at other sites unless the machine names are changed.
- The CGC tcltk BallAsync demo fails on the mips platform. We believe that this is due to interactions between tcltk and the select() function. This bug only affects demos which use the sleep star on the mips platforms.
- The sim-CG96 target was not compiled into the release. Thus, the only CG96 target available is the default-CG96 target. Unfortunately, the demos use the sim-CG96 specific IO stars, so they cannot be run with the default target. To use the sim-CG96 target, add the line below to the file $PTOLEMY/mk/stars.mk and rebuild pigi.
  CG96TARGETS = $(CG96T)/Sim96Target.o

A.5.3 Bugs in pxgraph

- If **`pxgraph`** is given only one point to plot, nothing appears in the plot window.
- If **`pxgraph`** is given exceptional numeric input, such as the IEEE floating point **`Inf`**, **`-Inf`**, or **`NaN`**, then it displays a blank window only.
- If **`pxgraph`** is given an empty file to plot, all it does is send a message “problems with input data” to the standard output.

A.5.4 HPPA specific bugs

As stated earlier, the hppa port in Ptolemy0.5 is fundamentally unchanged from the hppa port in Ptolemy0.5beta. There is a patch available for some of the bugs below. The patch is available via anonymous ftp from ptolemy.eecs.berkeley.edu, in the pub/ptolemy/ptolemy0.5/patches directory. See the README file for more information. We anticipate rereleasing the hppa port soon. Below is a list of known hppa specific bugs.

- Dynamic linking does not work on the hppa (Patch available).
- The hppa.cfront architecture is new port, and is known to have problems with random numbers. The hppa does not have a **`random()`** function, and the **`rand()`** function returns numbers over a different range than **`random()`**. There is a patch for this bug available via anonymous ftp from ptolemy.eecs.berkeley.edu at pub/ptolemy/ptolemy0.5/patches/pt-0.5-patch1.
- On the hppa, **`pxgraph`** does not print hardcopy perhaps because the hppa does not have an lpr command.
- On the hppa, compiling with **`gcc-2.5.8`** with the -O option fails because of an undefined symbol. This is a Gnu compiler bug.
- On the hppa, **`oct2ps`** core dumps, so printing facets probably won't work.
- On the hppa, the octtools **`masters`** program may core dump.
- On the hppa, problems have been reported in the de domain. The de:basic:real-time demo hangs (Patch available).
- On the hppa, under cfront, Xhistogram may have rounding problems.

A.5.5 GNU compiler bugs

If you write your own stars, then be aware that gcc-2.5.8 may not choose the expected cast. In particular, g++ has been know to cast everything to **`Fix`** if the explicit cast is omitted. The arithmetic is then performed using fixed-point computations. This will be dramatically slower than double or integer arithmetic, and may yield unexpected results. It is best to explicitly cast states to the desired form. For more information, see the ptlang chapter in the Ptolemy Programmers Manual.

On the mips, files must be compiled with the -G O option for dynamic linking to work. Unfortunately, under gcc-2.5.8, gcc spits out lots of warnings like:

```
warning: T_EOF not found in original or external symbol tables
```

These warnings may be ignored.

A.6 Additional Resources

The best, most complete source for information on Ptolemy is to be found in the Ptolemy manual, *The Almagest*. The manual is included in every distribution. In addition, the printed documentation is also available from the ILP office at the following address:

EECS/ERL Industrial Liaison Program Office
Software Distribution
205 Cory Hall
University of California at Berkeley
Berkeley, CA 94720
(510) 643-6687
email: ilpsoftware@eecs.berkeley.edu

A second source is the “ptolemy-hackers@ohm.eecs.berkeley.edu” mailing list. This list provides a forum for Ptolemy questions and issues. Users of the current release who have a Ptolemy question, comment, or think they’ve found a bug should send mail to ptolemy-hackers. See the “Mailing Lists” section above for details.

Another source is the Ptolemy FTP site. To access this, use anonymous FTP to ptolemy.eecs.berkeley.edu. The directory “pub/ptolemy/papers” contains some of the latest Ptolemy papers and articles. The directory “pub/ptolemy/mailing-lists” contains the logs from the “ptolemy-hackers” mailing list. Check there to see if your question has been asked before.