- What version of Ptolemy you are running. If you are not running the latest version, you may want to upgrade and see if your bug has been fixed already.
- Whether you are using prebuilt binaries, or if you compiled your binaries yourself.
- If the problem is with the compiler, and you are using the Gnu compiler, you should also state whether you are using the prebuilt compiler we provide, or your own version. You should also try running `gcc -v` to see what version of the compiler you are running.
- It is best for us if you can reproduce the bug in a small test case and send us your `~/pigiLog.pt` file, along with a detailed description of what you did and what happened.
- The best place to mail a bug report is to send it to the Ptolemy-hackers mailing list at `ptolemy-hackers@ptolemy.eecs.berkeley.edu`.
- If the bug consists of a problem with facets, you might want to uuencoded these facets and include them in your mail message. Note that there is a 40K size limit to mail to Ptolemy-hackers, so if your mail message is large, then you may want to send it only to `ptolemy@ptolemy.eecs.berkeley.edu`. To uuencode a facet and a galaxy:

```
tar -cf /tmp/facets.tar yourfacet yourgalaxy
uuencode /tmp/facets.tar facets.tar > facets.uu
```

- Then include the `facets.uu` file in a mail message. You may get some leverage out of compressing the `facets.tar` file before uuencoding it.

`-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.
- Zooming in on bar graphs doesn't work right.
- There is no way to produce hardcopy without running pxgraph interactively.
- Specifying the colors for data sets using X resources does not work.

G.5 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 and is available in our WWW and FTP sites. 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@ptolemy.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`. Archives of the mailing list are also available.

A third source is the Ptolemy Usenet news group `comp.soft-sys.ptolemy`.

A fourth source on the latest information about Ptolemy is the Ptolemy World Wide Web (WWW) server accessible by programs such as Netscape. The Universal Resource Locator (URL) for the Ptolemy WWW server is `http://ptolemy.eecs.berkeley.edu`. The WWW pages contain on-line access to a quick tour of Ptolemy, a list of publications, a collection of technical papers, information on the members of the development team, and links to WWW information on related tools. The WWW interface is a superset of our FTP server discussed below.

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.

G.6 Submitting a bug report

Ptolemy is offered without support, but we are nonetheless interested in hearing your feedback with regard to bugs. A good bug report consists of the following elements:

- What OS you are running under (e.g. SunOS4.1.3, Solaris2.5.1, HPUX-10.01).a

G.4.2 Bugs in pigi

- 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.
- `$PTOLEMY/bin/pigi` is a link to `$PTOLEMY/bin/pigiEnv.csh` which may add to your path. Under certain circumstances, you may get the message 'Warning: ridiculously long PATH truncated'. To work around this problem, try shortening your path. One trick is to use shorter pathnames in your path, perhaps using symbolic links.
- If your X server runs out of colors, `pigi` may crash. One workaround is to not run color hogs like `netscape`, or to limit the number of colors `netscape` can use.
- The compile-SDF target fails if a galaxy has incrementally linked in stars. The problem is that the `.h` file for the incrementally linked in star cannot be found by compile-SDF. As a workaround try editing the makefile in `~/PTOLEMY_SYSTEMS` and adding the directory where the incrementally linked stars is. Then build the binary by hand.
- Ptolemy 0.6 contains two new data types in the kernel: `String` and `File`. These two new data types should be useful in creating Design Flow Management (DFM) universes. However, there are only a few DFM demos.
- The Networks Of Workstations active messages (`NOWam`) CGC target is slow, but it does run under Solaris.
- To use the Dynamic Dataflow Code Generation (`cgddf`) CGC target, two files must be recompiled. The `cgddf` changes in `PortHole.cc` and Higher Order Functions (`HOF`) are incompatible, the `HOFMapGr` star breaks. The changes to `parScheduler.cc` are a fairly radical shift in how Ptolemy handles disconnected graphs. These changes could break other parallel schedulers. The script `$PTOLEMY/bin/mkcgddf` will build a `pigiRpc.ptrim.debug` binary that contains the `cgddf` target. This script is basically untested, but should work.
- The compile-SDF target cannot handle star parameters that use the Tcl Interpreter to evaluate Tcl commands.

G.4.3 Bugs in tycho

Ptolemy 0.6 contains the first release of `tycho`, the Ptolemy syntax manager. Development of `tycho` is ongoing, so the version included in Ptolemy 0.6 is by no means the final version. In particular, there are many more features to be implemented.

- If you start up tycho with Ptolemy in the background (i.e, `tycho -ptrim &`), then exiting from the MATLAB console may hang. The workaround is to place your `tycho` process into the foreground. This is a known bug with MATLAB.

G.4.4 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`,

can be increased by first exiting Mac X, then using the ‘Get Info’ option under the File 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’.

G.4 Known bugs

There are a number of known bugs that we have not had a chance to fix. You may find useful information about architecture dependencies in `$PTOLEMY/mk/config-$PTARCH.mk` for the particular architecture you are running under.

G.4.1 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.
- Deleting wires that are too long (i.e., end of wire is past the terminal, but near it) can result in `vem` dumping core.
- 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.
- 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.
- `vem` or `pigiRpc` may fail to start up on 4 bit (16 color) screens. At UC Berkeley, we use 8 bit and 24 bit screens. Black and white (1 bit) support in vem is a little weak. If you are on a black and white screen, you may need to modify `$PTOLEMY/lib/pigiXRes9`. (Note that `$PTOLEMY/lib/pigiXRes9.bw` is read when the `-bw pigi` options is used to create black and white screen dumps from a color monitor. `pigiXRes9.bw` has very little to do with running on a black and white screen).
- If the `VEMBINARY` environment variable is set, then `bin/pigiEnv.csh` will use the contents of that variable as the `vem` binary. However, the binary must be named `vem`, or there will be minor problems with X resources, such is small fonts and incorrect snap. Most users will never use the `VEMBINARY` environment variable, it is primarily used to debug `vem`.
- If your X server runs out of colors, `vem` may crash. One workaround is to not run color hogs like netscape, or to limit the number of colors netscape can use.
- Bug fixes to `vem` now mean that `vem` is more picky about text label heights. Currently, the range for text label heights is 0 to 80. If you select text and then use E, to edit the label, you may see an error if your text height is greater than 80. One workaround is to delete the text and reenter it with a height that is within the proper range. Another workaround is to use an old `vem` binary to edit the height and set it to within the range.
- If you re-read a facet that has an open run window, you won’t be able to dismiss the run window. The workaround is to open a new run window.

These modes are inappropriate for editing Ptolemy files. You can add the following line to your .emacs file in your home directory:

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

Problems with the colormap

Some applications, for example FrameMaker 5, 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 have access to a very restricted set of colors. If when you start `pigi`, 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 or Netscape), and restart `pigi`. A better solution is to configure the offending application to use fewer slots in the colormap. We have found that for FrameMaker 5, 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).

The HP window system, VUE, may not display the correct colors when running Ptolemy. If the Vem window appears with white text on a tan background, or if the Ptolemy run window appears blue instead of tan, then VUE is getting the Ptolemy colors wrong.

The solution here is force VUE to use the regular Ptolemy X resources. Before starting `pigi`, in an xterm, do the following line:

```
xrdb -load $PTOLEMY/.Xresources
```

Then run `pigi`.

The window manager crashes

The window manager `twm` sometimes crashes when you are running Ptolemy. We do not know why. It seems to be an interaction with Tk. 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.

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: Mac IIci w/ 20 MBytes DRAM, 210 MB HD and 21” two-page display/21gs Radius B&W monitor. The UNIX server is a HP 715 workstation.
- The ‘Display:’ option in the ‘Edit Remote Command’ MUST be set to ‘(2) Color Rootless’ mode.
- Increasing the memory allocation of Mac X to 4 MBytes also helps. The memory size

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

You have not installed the shared library needed by Ptolemy when it is used under OpenWindows.

pigi fails to start and gives 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 modifications 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. You may find the X11 program `xlsfonts` useful.

Ptolemy startup window only has an OK button

If the Ptolemy startup window does not have the Mr.Ptolemy bitmap and the copyright button, but instead the startup window is very small and has only an OK button, then you probably have font problems, see the section above for details about fonts.

Errors involving xedit

The program `xedit` is a program provided with Ptolemy to display text files on your screen during a Ptolemy run. Over time, we plan on using `tycho`, the Ptolemy syntax editor, as a replacement for `xedit`.

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 separate `tycho` window, do:

```
setenv PT_DISPLAY "tycho %s"
```

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"`.

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.

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 the facet is inconsistent, and there will be blank space in place of the icon in the schematic. 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 2-60. Invalid masters will be labeled "`INVALID`". You must replace the invalid reference with a reference to a valid master. The tcl script `$PTOLEMY/bin/ptfixtree` can be useful for changing large numbers of facets. `$PTOLEMY/bin/ptfixtree.tcl` file contains limited instructions on how to use it.

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

Ptolemy simulations do not stop

In the SDF domain, it is possible to have multirate systems where a single iteration fires a very large number 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 (e.g, rates such as 53:97). 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.

Multi-porthole galaxies fail

If a galaxy contains a input or output multi-porthole, 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.

Star is a compiled-in star and cannot be dynamically loaded

When you create a new universe (schematic), the domain assigned to the universe by default is SDF. If you create stars in the palette that are from another domain and then try to run the universe, you may get the error message

```
star 'Poisson' is a compiled-in star of domain DE. Cannot dynamically
load a compiled-in star class.
```

The solution is to change the domain of the universe by choosing edit-domain from the pigi menu (the keyboard short cut is 'd').

G.3 Window system problems

Below we discuss various problems we've seen between Ptolemy and the X window system.

tycho fails to start up, giving TCL_LIBRARY messages

There are several ways to start up tycho, the Ptolemy syntax manager. $PTOLEMY/bin/tycho is a link to a script that processes command line arguments and starts up the appropriate binary. If you type tycho -pigi, tycho starts up with a binary that includes the Ptolemy system. If you type just tycho, then tycho starts up with the generic itkwish binary that is built from the itcl sources, which does not include any of the Ptolemy system.

If your Ptolemy distribution is not at /users/ptolemy, and you are running from prebuilt binaries, then if you run tycho with the prebuilt generic itkwish binary, you may see messages about:

```
application-specific initialization failed: can't find /users/
ptolemy/tcltk/itcl/lib/tcl7.4/init.tcl; perhaps you need to install
Tcl or set your TCL_LIBRARY environment variable?
```

What's happening here is the itkwish binary we ship has the /users/ptolemy path hard coded into it and the binary is not finding the libraries it needs. The reason this happens is that we want tycho to be able to run outside of Ptolemy on machines that have only generic itkwish installed from the itcl distribution. There are a few workarounds:

- Create a link from /users/ptolemy to the Ptolemy distribution.
- Run tycho -ptiny instead. The tycho script assumes that the Tcl installation is located in $PTOLEMY/tcltk if it is called with the -ptiny, -ptrim or -pigi arguments, so the script will do the right thing.
- Set environment variables in a script and then call itkwish:

```
#!/bin/sh
TCL_LIBRARY=$PTOLEMY/tcltk/itcl/lib/tcl
TK_LIBRARY=$PTOLEMY/tcltk/itcl/lib/tk
ITCL_LIBRARY=$PTOLEMY/tcltk/itcl/lib/itcl
ITK_LIBRARY=$PTOLEMY/tcltk/itcl/lib/itk
IWIDGETS_LIBRARY=$PTOLEMY/tcltk/itcl/lib/iwidgets2.0
export TCL_LIBRARY TK_LIBRARY ITCL_LIBRARY
export ITK_LIBRARY IWIDGETS_LIBRARY
exec $PTOLEMY/bin/tycho $*
```

G.2 Common problems while running pigi

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, as of this writing, this does not work on all platforms. 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.

```
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 fix is to run `pigi` in the foreground. Another fix is to eliminate this command from your login files.

pigi fails to start up, giving shared library messages

On certain platforms, Ptolemy is built using shared libraries. In Ptolemy 0.6, the Solaris, HP and possibly Linux platforms use shared libraries. See the "Shared Library" appendix for more information about shared libraries.

At run time, if shared libraries cannot be found, you may see a message under HPUX-10.01 like:

```
/usr/lib/dld.sl: Can't find path for shared library: libuprintf.sl
```

Under Solaris 2.x, you might see:

```
ld.so.1: /users/cxh/pt/bin.sol2/vem: fatal: librpcserver.so:
can't open file: errno=2
```

The message that you see may vary but the problem is that the binary cannot find the shared libraries to which it was linked. There are a few reasons this could be happening:

- The shared libraries are not where the binary expects them to be. If you are running from prebuilt binaries and your Ptolemy tree is not at `/users/ptolemy`, then this may be the problem.

  If you are running from prebuilt binaries and your Ptolemy distribution is not at `/users/ptolemy`, then you may need to set an environment variable to indicate what path should be searched for shared libraries. Under HPUX, the environment variable is `SHLIB_PATH`; under Solaris, the environment variable is `LD_LIBRARY_PATH`. The file `$PTOLEMY/.cshrc` should contain the proper commands to set the appropriate environment variable, though you may need to uncomment some lines. For HPUX, you could type the following command. (Do not type a space after the \ at the end of the first line, just press `Return`):

  ```
  setenv SHLIB_PATH {$PTOLEMY}/lib.{$PTARCH}:\
        {$PTOLEMY}/octtools/lib.{$PTARCH}:{$PTOLEMY}/gnu/{$PTARCH}/lib
  ```

  For Solaris, you could type the following command (all on one line):

  ```
  setenv LD_LIBRARY_PATH {$PTOLEMY}/lib.{$PTARCH}:\
        {$PTOLEMY}/octtools/lib.{$PTARCH}:{$PTOLEMY}/gnu/{$PTARCH}/lib
  ```

It is best to place these commands in your `~/.cshrc` file. It is possible that you might have to add other directories to the shared library path. For example, the Ptolemy binaries are compiled with `/usr/openwin/lib` as the location of the X windows libraries. If your X windows libraries are in another directory, then you will need to add that directory to the shared library path. See "Window system problems" on page 7-182.

- You do not have a library with that exact name. You may have an earlier or later version. Recompiling Ptolemy from scratch is one solution. It may also be possible to set up symbolic links to the proper libraries from a directory on the shared library path.

Chapter 7. Troubleshooting

This section lists common difficulties encountered when installing and running Ptolemy. This list is, of course, by no means complete. If you do not find your particular problem here, refer to the section “Additional resources” on page 7-187.

The most recent version of this section can be found on the Ptolemy home page at `http://ptolemy.eecs.berkeley.edu/`.

G.1 Problems starting pigi

Mr. Ptolemy window does not come up

Ptolemy consists of two processes, `vem` and `pigiRpc`, that communicate via Remote Procedure Calls (RPC). `Vem` is the first process that starts up, and it produces a vem console window in the upper left corner of the screen and a green demo window just below it. When `pigiRpc` starts up, you should see a window in the middle of your screen that has the Mr. Ptolemy bitmap and a brief description of the binary you are running.

If the `pigiRpc` process fails to connect to the `vem` process, you won’t see the Mr. Ptolemy bitmap, and the shift-middle-button menus will not be active. This problem seems to be most common on Linux machines, in part because they are often not on a network. If you are running under Linux and your installation is configured to use the network, then you may need to rebuild Ptolemy from source.

If you are running on a machine that is not connected to a network, you will need to provide some network support for `pigi` to start up. `vem` and `pigiRpc` communicate with each other via RPCs, which require some intra-machine network support. One quick test is that you should be able to ping yourself:

```
/usr/etc/ping `hostname`
```

There are several workarounds to this. One is to add the name of your host to the loop back line in `/etc/hosts` (here we add the name *myhostname*):

```
127.0.0.1 localhost myhostname
```

Another solution is to use `route` to route packets to your host through the loopback interface. As `root`, type:

```
route add `hostname` localhost 0
```

See the `ping`, `netstat` and `route` commands for more information.

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