Chapter 5.
Debugging Programs on IRIX Systems

To debug MPI 3.1 programs on IRIX systems, you can use the XMPI graphical interface or either of the Silicon Graphics debuggers, CVD (ProDev WorkShop) and DBX. For information on using XMPI, see Section 5.3.

You cannot simply start an MPI program from the debuggers because MPI 3.1 uses Array Services to start an MPI application and the application is a collection of IRIX processes. Instead, you must start the program with `mpirun` or `xmpi` commands as you normally do, then attach to the running program with the debugger.

This startup method is inconvenient, because you cannot initially start an MPI application in a stopped state. If your program is long running, this may not be a problem because there will be time to obtain the process IDs and attach the debugger to them. If your program is short, or you need to debug it soon after MPI initialization, code a call to a sleep routine to allow time to attach to the program.

The debugging procedures in this chapter assume that you have added a sleep routine to the beginning of your program.

Procedure 3: Using the ProDev WorkShop debugger, CVD

Use the following procedure to debug an MPI program with the ProDev WorkShop debugger, CVD:

1. Modify the MPI application to call a sleep routine.

To modify the MPI process, begin with a preliminary call to `SLEEP` before calling the MPI initialization routine. The following example illustrates a modified Fortran program:

```
CALL SLEEP(60)

CALL MPI_Init ()
```

2. Start your MPI program after building it.

3. Obtain process IDs.

Be sure to obtain the processes with your program name, not the `mpirun` process, from the output of the following command:

`ps -u` *`userid`*

4. Attach to your application by using the CVD debugger. Attaching by using CVD, as follows, places you in the sleep library routine in a stopped state:

`cvd -pid` *`pid`*

Note: To find your program source, use the `cvd up` command to move up the stack several times.

5. Instruct the CVD debugger to automatically attach to child processes of the program as they are created (forked) by doing the following:

a. Select `Admin --> Multiprocess View`.

b. `Select Config --> Preferences`.

c. Select `Attach to forked processes` and `Copy traps to forked processes`.

6. Use the CVD debugger to set a breakpoint, if you wish.

7. Instruct CVD to automatically attach to your application's processes by selecting `Continue`. After your selection, processes are listed in the `Multiprocess View`. From this view, you can start CVD windows for other

processes to debug different MPI ranks.

The initial attached process remains running in `MPI_Init` for the duration of your program. For example, if you specify four processes on the `mpirun` command line, five processes will be listed in the `Multiprocess View`.

Caution: Do not stop the `MPI_init` process. Stopping `MPI_init` can halt MPI communications.

Procedure 4: Using the DBX debugger

Use the following procedure to debug an MPI program by using the DBX debugger:

1. Follow steps 1 through 3 in Procedure 3.

2. Attach to your running program by using the following `dbx` command:

```
dbx -p pid
```

3. Debug child processes as they are created.

Use the following command to tell DBX that you want to debug child processes as MPI creates them:

```
set $promptonfork = 2
```

4. Continue your program by entering the `continue` command.

5. As each process is created, set a breakpoint in it manually, then continue it. The DBX debugger stops each child process separately. For more information on debugging multiprocess programs with DBX, enter `help mp` and `help hint_mp_debug` on the `dbx` command line.

5.1 Core File Contents

If an MPI program aborts, the `core` file contains only the aborting process. The remaining processes in the application continue to run. You can inspect the `core` file using either cvd or dbx, but with either debugger, you will see only information for the failing process.

5.2 Running Distributed Applications

When an application is distributed across multiple nodes in an array, you must run multiple copies of cvd to debug the process on each node. See Array Services documentation for more information.

Use the following `cvd` command to attach to a particular process on a specific node:

```
cvd -host hostname-pidpid
```

The Array Services commands `ainfo`, `array`, and `aview` also provide information on MPI processes.

5.3 Using XMPI

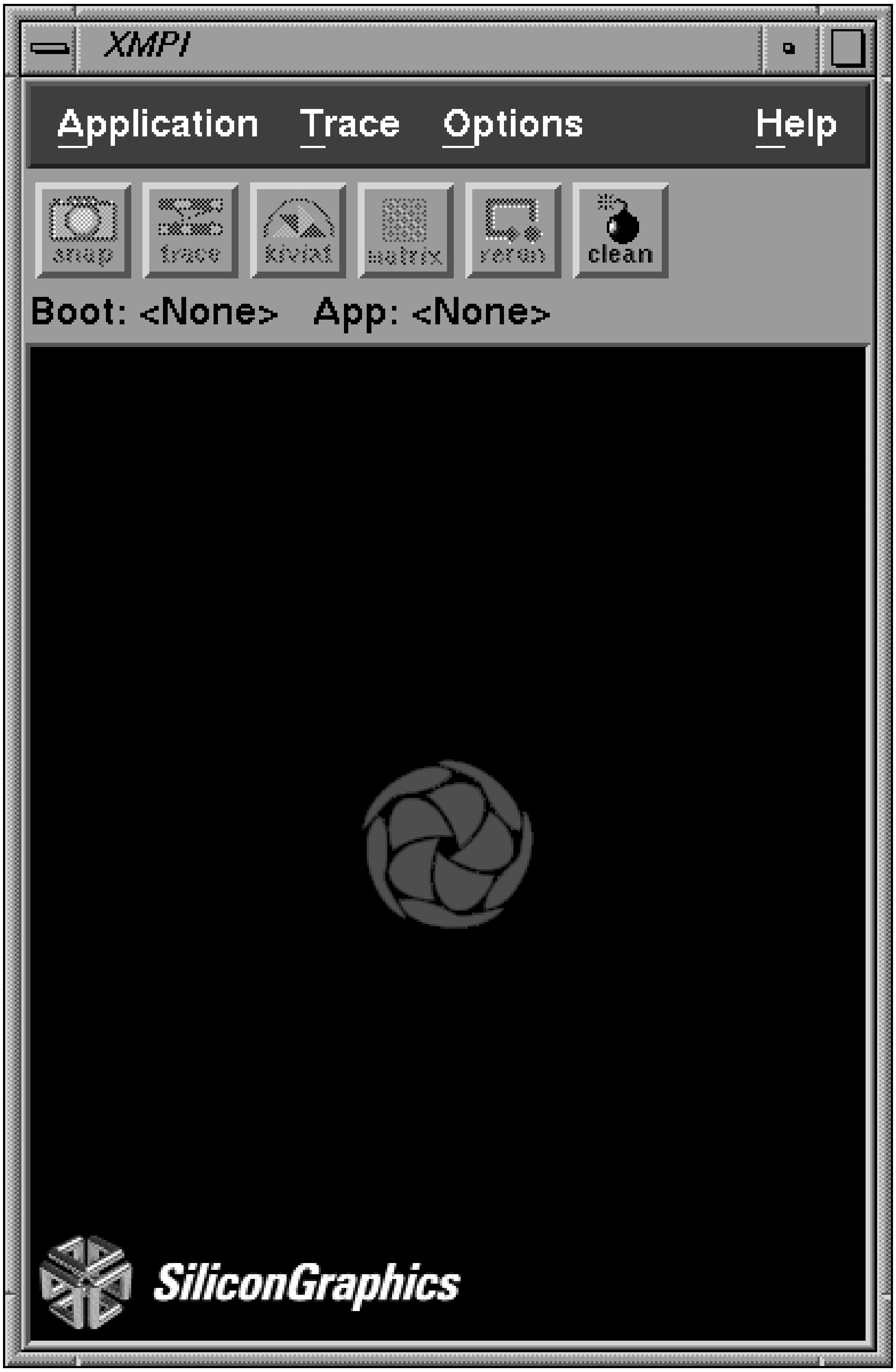
The XMPI graphical interface starts MPI applications and allows you to view message passing events that occur during program execution. It can also save a file of these events if you want a permanent record. XMPI must be running on any node on which it is used.

If you understand the concepts of MPI, you can probably master XMPI by starting it, experimenting with its menus, and browsing its help displays. But the information in this section acquaints you with XMPI menus and dialogs to make your exploration of XMPI more focused and efficient. It also explains how to use menus and dialog boxes to do several commonly performed tasks.

5.3.1 Starting XMPI

To start XMPI, type the `xmpi` command in a shell window. After your entry, the XMPI main window is displayed as follows:

Figure 1. XMPI main window



The XMPI menu bar offers the following menus:

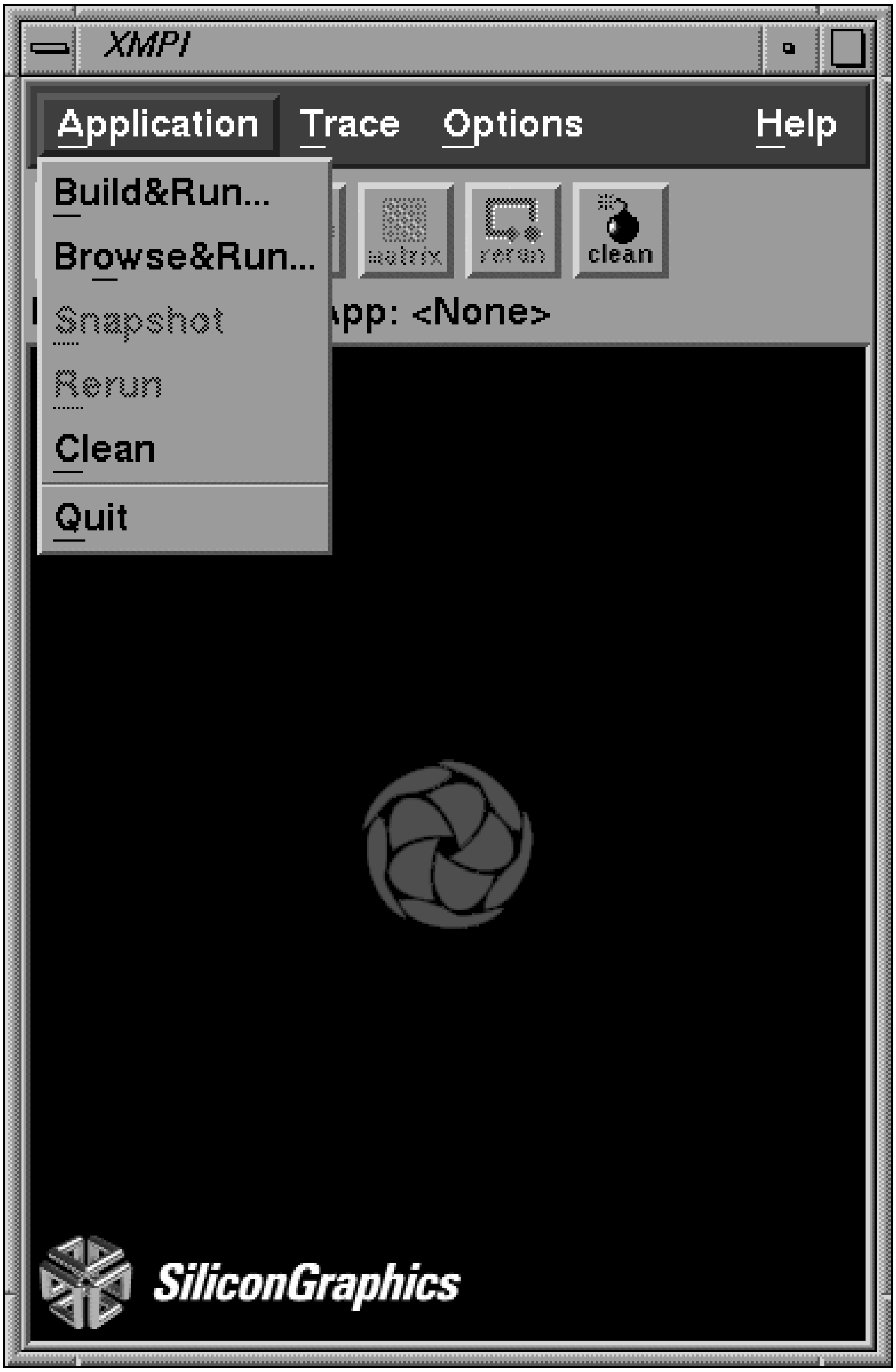
- The `Application` menu contains options for running MPI applications and managing other aspects of program execution. Section 5.3.2, describes each of the options on this menu and contains procedures for running MPI applications with XMPI.
- The `Trace` menu contains options for viewing and managing trace files. Section 5.3.3, describes the options on this menu and contains procedures for viewing traces.
- The `Options` menu contains implementation−specific functions that are not relevant to the Silicon Graphics version of XMPI.
- The `Help` menu provides a list of topics that you can select to obtain information about XMPI features and functions. Section 5.3.4, explains how to access additional XMPI documentation.

The buttons below the menu bar duplicate selections on the `Application` menu. When you are viewing a trace, it is often easier to use these buttons than the menu selections.

5.3.2 The `Application` Menu

The `Application` menu, shown in Figure 2, contains options for running and managing MPI applications. Notice that this menu contains two selections for running MPI applications: `Build&Run...` and `Browse&Run...`. Choose the one that best suits your needs.

Figure 2. `Application` menu



Use the `Application` menu options for the following tasks:

- Use the `Build&Run...` option to run a job on distributed hosts or to define new execution parameters for the job. With this option, you can save the execution parameters to a file and use the parameters when you run the application later.

- Use the `Browse&Run...` option to run an application when a file of its execution parameters was previously created with the `Build&Run...` option.

- Use the `Snapshot` option on a running program to record information on trace events at a given instant in time.
- Use the `Rerun` option to clean any remnants left from a previous job and run it again.
- Use the `Clean` option to cancel a running MPI application and remove status information about its processes from collection tables.

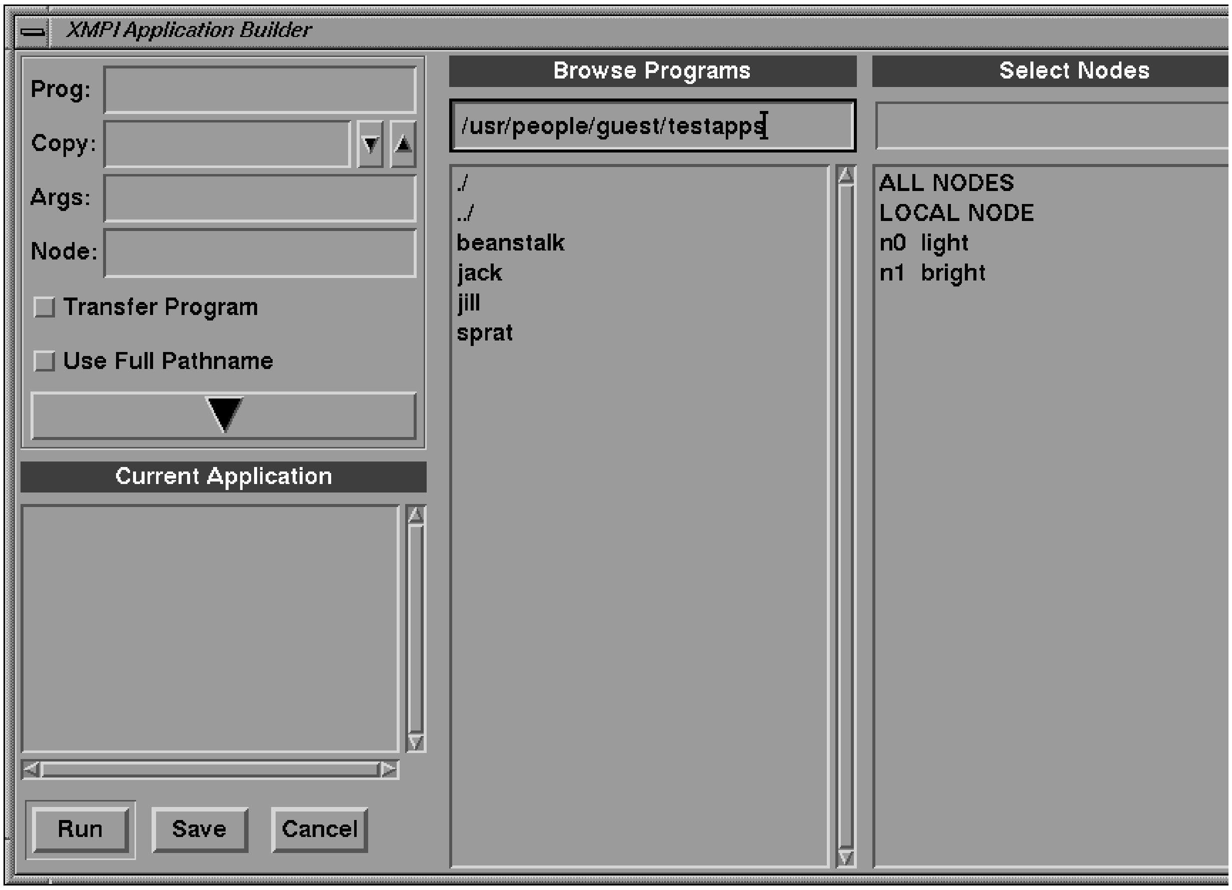
After an application is launched, you can use the `Snapshot` option to collect information on process events at any instant in time. Use the `Clean` option to stop a process without stopping XMPI.

Procedure 5: Using the `Build&Run...` option

Use the following procedure to define parameters for MPI programs and launch them from XMPI:

1. Select the `Build&Run...` option from the `Application` menu. When you select this option, the `Application Builder` dialog box is displayed (shown in Figure 3). This dialog box accepts MPI program parameters, queues the programs, and launches them.

Figure 3. `Application Builder` dialog box



2. Specify program parameters in the left column of the dialog box.

Enter the name of the MPI application, the number of copies that you wish to run, program arguments, and the node on which the application is to run. (You can also select the program and node from the `Browse Programs` and `Select Nodes` lists at the right.)

To change the directory or node parameters, go on to Procedure 5, Step 3. If directory and node parameters are satisfactory, go directly to Procedure 5, Step 4.

Note: The `Transfer Program` and `Use Full Pathname` buttons are not implemented for this release.

3. Change the program directory and MPI node, as necessary.

By default, XMPI assumes that you want to run a program located in the directory in which you entered the `xmpi` command. If the `Browse Programs` column does not contain the program that you want to run, enter a different directory name at the top of this column, then select the program that you want from the new list that is displayed after your entry. If the program that you want to run is on a different node, use the `Select Nodes` column.

The `Select Nodes` column contains a list of the available hosts on which you can run an MPI application. You can select an individual node, all nodes, or the just the local node from the available nodes list. You can also add a node to the list by entering its name in the entry field at the top of the column. The node that you add must be listed in the following file:

```
/usr/lib/array/arrayd.conf
```

4. Save the execution parameters, as necessary.

To store the execution parameters that you entered in a file, click the `Save` button. When you select this button, you are prompted to enter the name of the parameter file.

5. To queue the job, click the `commit` button (the widget containing a down arrow).

The `commit` button queues the job for execution. The job remains in the queue until you select the `Run` button.

6. Queue additional jobs, as necessary.

To specify various combinations of nodes and program parameters, and to queue the additional jobs, repeat steps 1 through 5.

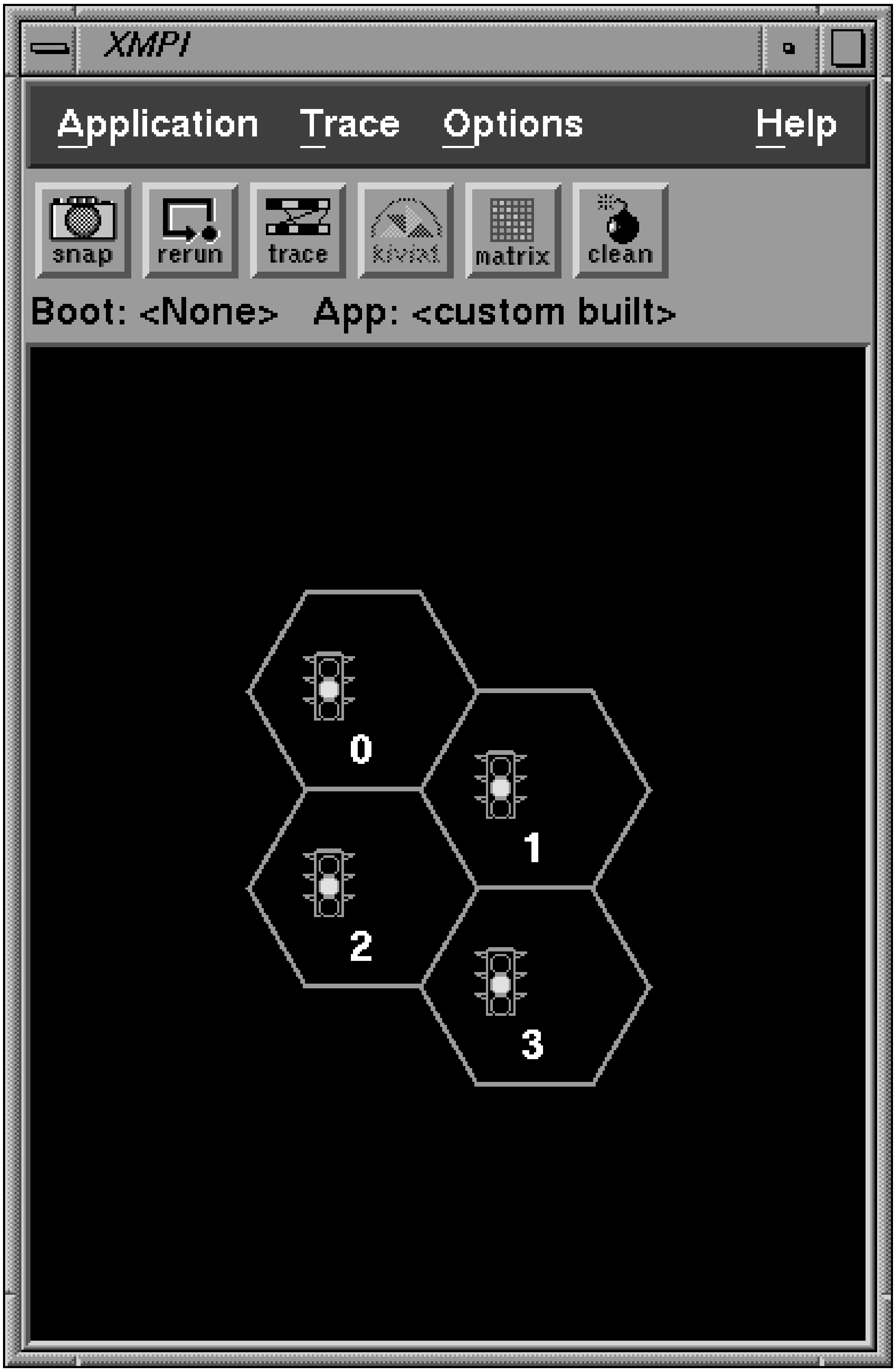
To remove a job from the queue, select the job by double−clicking its name in the queue and then press the `Delete` key.

If you change your mind and decide not to run any applications, click the `Cancel` button to close the `Application Builder` dialog box without taking action on your entries.

7. To start the application, click the `Run` button.

The XMPI banner is replaced by a display of the running application, shown in the following figure:

Figure 4. XMPI display of a running application

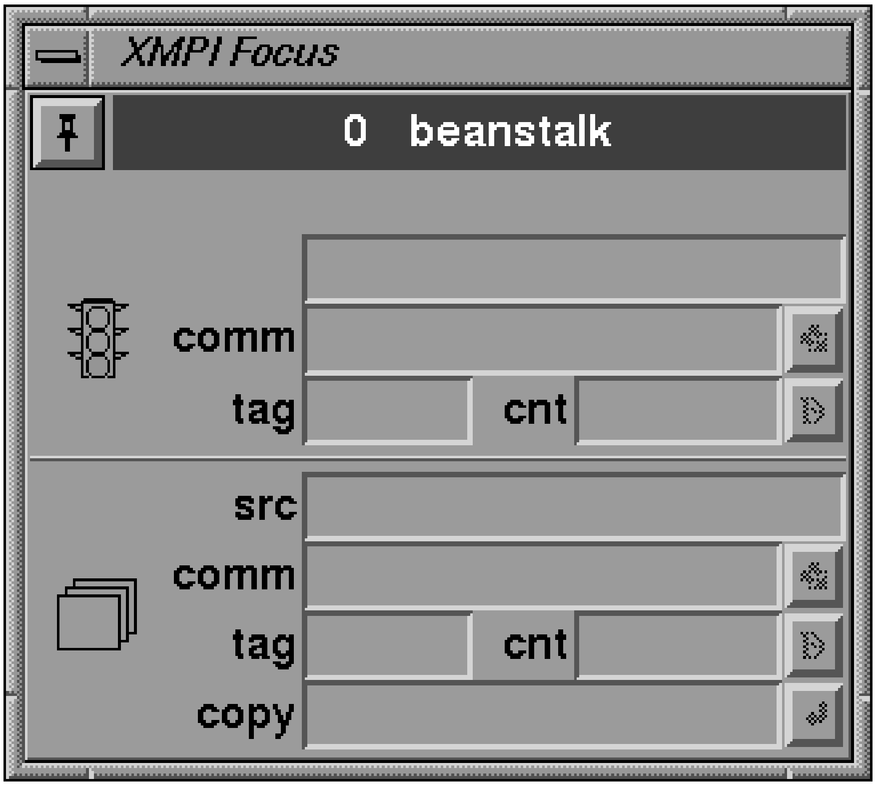


Each process in the job is represented by a hexagonal icon. The color of the traffic light in the icon indicates the following process status:

- Green indicates that the process is running but it is not executing an MPI routine.
- Yellow indicates that an MPI routine call is in progress and no blocking is occurring.
- Red indicates that an executing MPI routine is blocking.

You can display a record of events for any process by clicking its icon. The individual record for each process is shown in the following figure:

Figure 5. Individual record of events for an MPI process



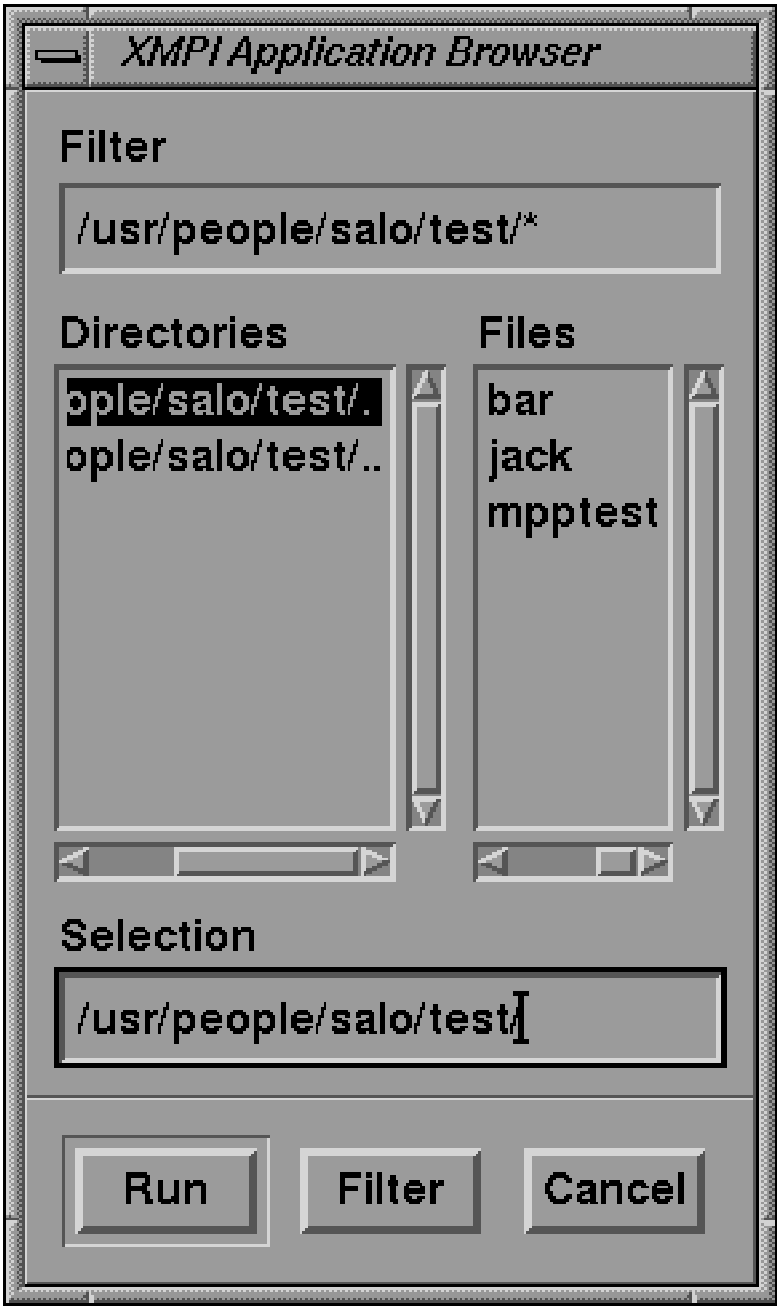
8. Perform the following functions as necessary:

- To refresh the information on the record and in the process icons, click the `snap` button on the main window (or choose the `Snapshot` option from the `Application` menu).
- To clear the information and remove records from the window, click the `clean` button.
- To clear the information and restart the application, click the `rerun` button.

Procedure 6: Using the `Browse&Run...` option

When you select `Browse&Run...`, the `Application Browser` dialog box (shown in Figure 6) is displayed. Notice from this figure that this dialog box has no entry fields for execution parameters. With this option, execution parameters are read from a file that was previously created by selecting `Build&Run...` (see Procedure 5, Step 4).

Figure 6. `Application Browser` dialog box



To run an MPI application with the `Browse&Run...` option, use the following procedure:

1. Choose a directory and program name from the `Directories` and `Files` lists in the `Application Browser` dialog box.

2. Change the directory, as necessary.

XMPI assumes that the program that you want to run is in the directory in which you entered the `xmpi` command. If the `Directories` and `Files` lists do not contain the directory that you want, enter a directory name in the `Filter` field at the top of the dialog box. Then select the program from the new ones displayed in the `Files` list.

3. Click the `Run` button to start the application.

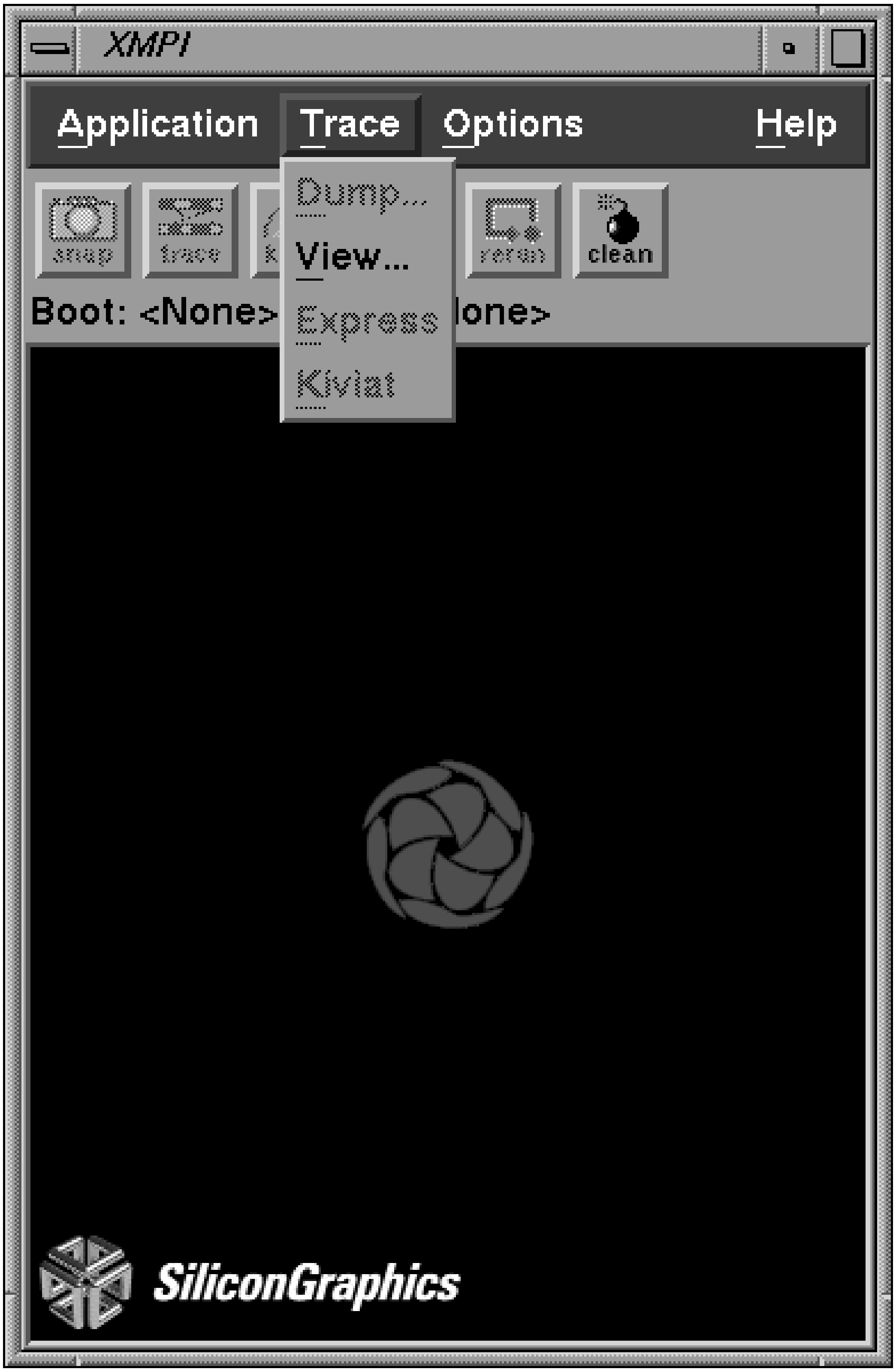
After your selection, the XMPI banner is replaced by a display of the running application. See Figure 4, and the discussion that follows it for information on monitoring the application after it starts.

5.3.3 The `Trace` Menu

Whenever you launch an MPI application with XMPI, it automatically gathers trace information from the MPI daemons on the host nodes. The `Trace` menu (shown in Figure 7) contains options for viewing information collected from these daemons. Use the following `Trace` menu options to perform these tasks:

- `Dump` saves trace information in a file.
- `View` displays information that was previously saved in a trace file.
- `Express` displays trace information while the application is executing.
- `Kiviat` displays a trace file in a Kiviat diagram.

Figure 7. XMPI `Trace` menu



Procedure 7: Creating and viewing trace information

Use the following procedure to create and view a trace file:

1. Run the MPI application.

To launch the MPI program, use the `Build&Run...` or the `Browse&Run...` option from the `Application` menu. (For more information, see Procedure 5, or Procedure 6.)

2. Dump the trace data to a file.

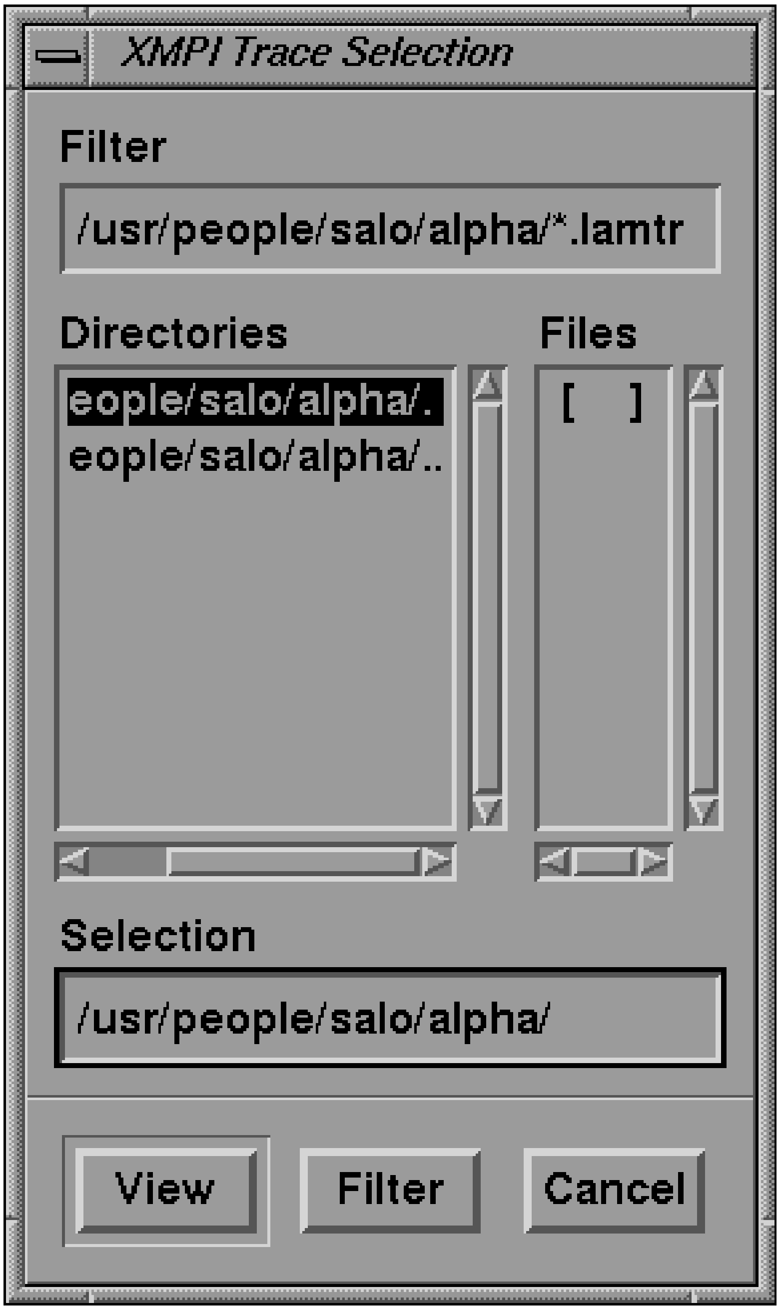
To create a file containing the trace information, choose the `Dump` option from the `Trace` menu. When you choose this option, a dialog box is displayed so that you can specify the name of the trace file. If you do not include a directory in your specification, the trace file will be stored in the directory in which you entered the `xmpi` command.

3. Choose the trace file that you want to display.

After a trace file is created, you can view it by choosing the `View` option on the `Trace` menu. When you choose `View`, the `Trace Selection` dialog box (shown in Figure 8) is displayed so that you can specify the file to display.

If you want to display a trace file other than the one that was just created, use the `Directory` and `Files` lists to select a different trace file.

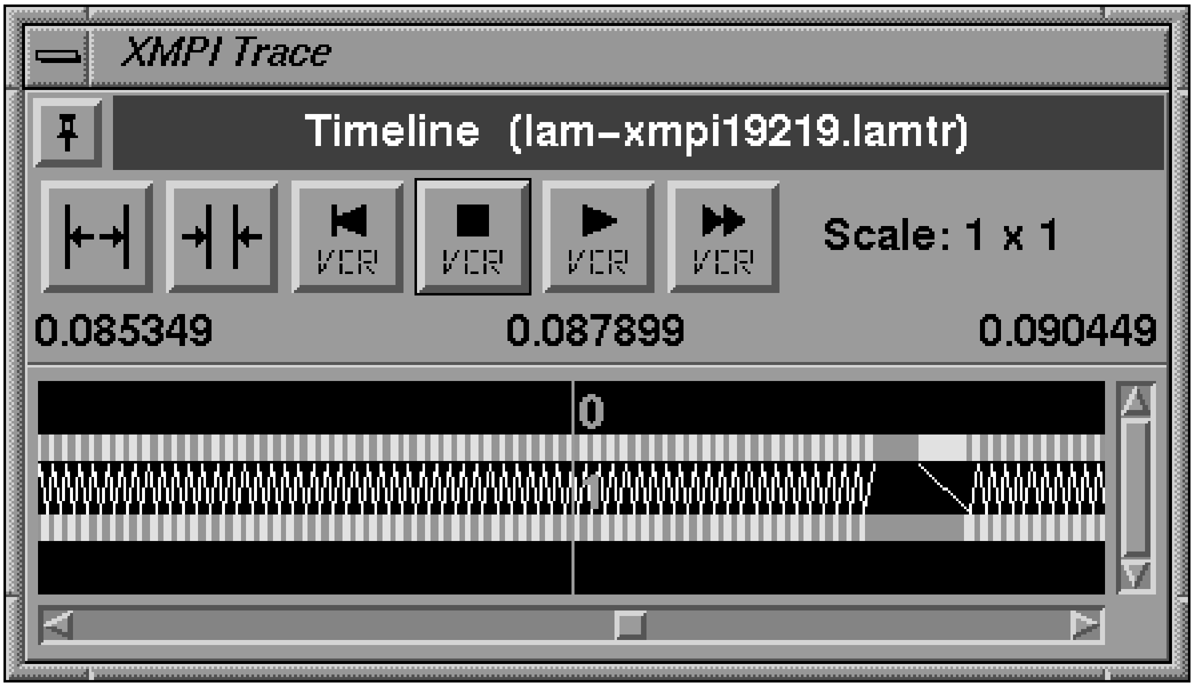
Figure 8. Dialog box for displaying trace files



4. View the trace file.

Click the `View` button to see a time line of the trace. Figure 9, illustrates a portion of a time line on two processes.

Figure 9. XMPI display of trace information



The horizontal bars reflect the state of each host over time. Arrows between bars represent messages that are being passed between hosts. To control the level of detail that is shown, use the `expand` and `compress` buttons (icons showing arrows pointing out and arrows pointing in above the display). To control the time−lapse display, use the `VCR` buttons; they start and stop, fast−forward, and rewind the trace.

Note: To zoom in on an area of the time line to see it in greater detail, position the cursor over the area of the time line that you want to expand. Press the right mouse button and sweep over the area of interest to frame it. Each time you create a new frame, the contents of the frame are magnified.

Procedure 8: Viewing a trace as an application runs

Use the following procedure to view a trace file while an application is running:

1. Run the MPI application.

To launch the MPI program, use the `Build&Run...` or the `Browse&Run...` option from the `Application` menu. (For more information, see Procedure 5, or Procedure 6.)

2. Choose the `Express` option on the `Trace` menu.

When you choose the `Express` option, a time line of the trace is displayed. For a description of the display, see Figure 9, and the information following the figure.

Note: To view a trace while an application is running, use the `Trace` button on the main window instead of the `Express` option.

5.3.4 `Help` Menu

The `Help` menu (shown in Figure 10) contains the primary documentation for XMPI. The `About XMPI` option gives the XMPI version number and copyright information. This option also contains a World Wide Web location at which you can find help on XMPI.

When you select the `Help` option from the `Help` menu, your Web browser launches and connects you to the Web site containing XMPI help. To access this Web page, the XMPI Web site must be accessible from the host from which you are using XMPI.

Figure 10. XMPI `Help` menu