Appendix C

Glossary

This is a glossary of specialized terms used in Prism.

alias — An alternative name for a Prism command. You can set up these alternative names via the **`alias`** command.

breakpoint — An event that stops execution of a program at a specific location, when a condition is met, or when a variable or expression changes its value.

call stack — The list of procedures and functions currently active in a program.

command window — The pane at the bottom of the main Prism window, in which messages are displayed and the user can issue commands.

context — In printing and displaying data, the active elements of a variable or expression. Prism handles active elements differently from inactive elements in certain data representations.

current file — The source file currently being displayed in the source window. When the program is first loaded, this is the file that contains the **`main`** function. This can change during execution, or as a result of actions you perform in Prism. The current file determines the scope that Prism uses in identifying line numbers and variables.

Copyright © 1991–1994 Thinking Machines Corporation

current function	The function or procedure displayed in the source window. This is the **`main`** function when a program is first loaded. It can change during execution, or as a result of performing certain actions in Prism. Prism uses the current function to determine the scope it should use in identifying variables.
data navigator	A component of a visualizer that lets you manipulate the display window relative to the data being visualized.
dialog box	A window used by Prism to obtain information from or provide information to the user.
display window	The pane within a visualizer that shows the data.
event	A breakpoint or trace, along with associated actions, that the user creates to control the execution of a program.
event list	The list of events, displayed as part of the event table or via the **`show events`** command.
event table	A table that lists the events that are to take place during the execution of a program, and provides mechanisms for adding, editing, and deleting these events.
execution pointer	The greater-than symbol (**>**) that appears in the line number region and points to the next line in the source window to be executed.
history region	The area of the command window where Prism displays messages and responses to commands.
I-beam	An I-shaped graphical image that appears in a text entry box to show that text can be entered in it.
immediate action	An action that takes place as soon as a menu selection is chosen; no dialog box or window is displayed to obtain further information from the user.

Version 2.0, April 1994
Copyright © 1991–1994 Thinking Machines Corporation

keyboard accelerator A sequence of keystrokes that performs an action without the need to display a menu.

line-number region The area to the left of the source window in which line numbers are displayed. The user can set breakpoints in this region.

location cursor A graphical image that represents the focus of keyboard actions; it is displayed as a box surrounding the selected object.

menu bar The line of text across the top of the main Prism window. A pulldown menu is associated with each word along this line; you can choose items from these menus to perform actions in Prism.

mnemonic A single letter (generally the first letter) underlined in a menu title or menu selection; by typing this letter, you can display the menu or (when the menu is displayed) choose the menu selection.

mouse pointer The graphical image (for example, an arrow head) that appears on the screen and represents the current location of the mouse.

performance advisor An analysis that Prism provides of performance data.

Prism resource A variable that controls an aspect of Prism's behavior. Default values for many Prism resources appear in Prism's file in your system's **`app-defaults`** directory. You can change these defaults by specifying new values in your X resource database, or by using the **Customize** utility.

qualified name A version of the name of a variable or function that identifies it more completely within a program. For example, **`` `foo`bar`x ``** identifies the variable **`x`** in the function **`bar`** in the source file **`foo`**. Names can be either *partially qualified* or *fully qualified*.

Copyright © 1991–1994 Thinking Machines Corporation

resize box — The small box in the corner of many windows; you can drag this box to resize the window.

resolving names — The procedure by which Prism determines which (of possibly several) variables or procedures with the same name it is to use (for example, in an expression).

resource — In performance analysis, one of the subsystem components (for example, CM cpu time, Send/Get communication) for which Prism measures a program's usage.

scope pointer — The – symbol in the line-number region. It indicates the beginning of the scope that Prism uses for identifying variables.

source window — The pane in Prism's main window where source code is displayed.

subsystem — In performance analysis, one of the independent systems for which Prism measures a program's usage. For a CM-2 or CM-200, they are the front end and the CM. For the CM-5, they are the partition manager and the nodes. The front end and partition manager are referred to as *serial* subsystems. The CM and the nodes are referred to as *parallel* subsystems.

tear-off region — The area beneath the menu bar in Prism's main window. You can move frequently used menu selections and Prism commands to this region to make them more accessible.

triggering condition — An action that causes an event to take place. Triggering conditions include reaching a program location, a logical condition becoming true, and the value of a variable changing.

visualizer — A window that displays scalar or parallel data using one of several visual representations available in Prism.

Version 2.0, April 1994
Copyright © 1991–1994 Thinking Machines Corporation

watchpoint An event that occurs when the value of a variable changes.

Copyright © 1991–1994 Thinking Machines Corporation