Rational X Interface User's Guide

RATIONAL

Product Number: 4000-00436

Rev. 1.0, November 1990 (Software Release Rev10_7_2)

This document is subject to change without notice.

Note the Reader's Comments forms at the end of this book, which request the user's evaluation to assist Rational in preparing future documentation.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Preface

The *Rational X Interface User's Guide* describes how to use the Rational® X Interface (RXI) to access the Rational Environment™ through a workstation. This guide assumes a basic familiarity with the Rational Environment, the workstation and its operating system, and use of the workstation mouse.

The information presented in this guide is designed apply to all versions of RXI, regardless of the workstation being used. However, information for specific workstations is available in a series of supplements to this guide. Each supplement presents several kinds of information: the terminal type designation and how to use it; the workstation keyboard layout and how it relates to Rational Environment operations; and a keymap showing specific keystrokes for Environment operations.

You should receive the supplement for your workstation with your regular documentation. If you do not receive it, or if you want information on what other supplements are available, see your Rational representative.

This guide is organized as follows:

- Chapter 1, "Key Concepts," gives an overview of RXI usage and describes the workstation keyboard and mouse.
- Chapter 2, "Getting Started with RXI," explains how to invoke the RXI terminal emulator, how to manage the special RXI window, and how to access the Rational Environment.
- Chapter 3, "Using the Environment through RXI," describes how to use menus and the workstation mouse to communicate with the Environment.
- Chapter 4, "Customizing RXI," describes how to modify RXI on your workstation to suit your own needs. Customizing RXI includes adding user-defined menu items and changing the RXI window font.

This guide assumes that both the X Window System™ and RXI are already installed on the workstation. If X Windows is not installed, see your system administrator.

Contents

1 Key Concepts

The Rational Environment™ is an environment for developing and maintaining large software projects written in Ada. The Environment runs on the Rational R1000® Development System and can be accessed from many different types of X terminals and workstations by means of the Rational X Interface (RXI) terminal emulator application. The relationship among these systems and applications is represented in Figure 1-1.

As indicated in Figure 1-1, the X Windows server is layered above the operating system on the workstation. The X Windows server supports a *window manager* as a client. Together, X Windows and the window manager support the display and management of multiple windows on your workstation screen. These windows provide multiple workspaces through which you can access various tools and applications.

The RXI application is a terminal emulator that operates with the X Window System™. From the workstation, you can connect to the R1000 through an RXI terminal emulator window using any convenient communication path, such as Telnet, if the workstation is connected to the R1000 via Ethernet.

All of the usual window-management operations provided by the local window manager can be used to manipulate the RXI windows along with those from other X Windows applications.

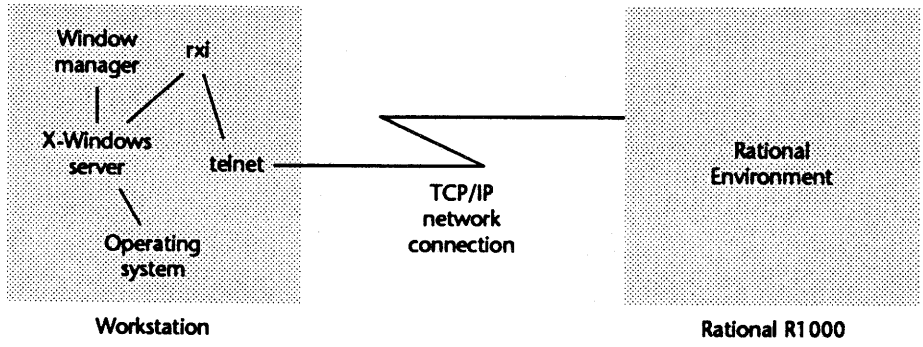

Figure 1-1 Conceptual Diagram of Workstation and Rational R1000

This guide assumes that the workstation and the R1000 are already on the network, that the X Window System, the window manager, and RXI have already been installed on your workstation, and that X Windows and the window manager are already running.

WINDOW MANAGEMENT

The RXI application is a terminal emulator. RXI allows the user to interact with the Rational Environment through the X Window System. RXI accomplishes this by translating R1000-generated screen formatting commands into X Window System commands.

RXI also implements mouse-driven menu, selection, and pick operations; it translates these operations into character sequences or commands for the Rational editors. When the cursor is in an RXI window, the menu and mouse buttons invoke commands specific to the R1000 context. Outside the RXI window, the mouse and menus are in the X Window management context. Within an RXI window, you may have one or more R1000 Environment windows, all belonging to the same R1000 session.

After you are logged into the Environment through an RXI window, two windowing systems are in effect:

- The workstation's windowing system is controlled by the X-based window manager. The window manager operations affect all X-based application windows, including RXI windows. The window manager operations include using the mouse and menus to change X-based windows into icons, resize them, and hide them behind other X-based windows.
- The R1000 Environment's windowing system affects only those Environment windows belonging to a single session and contained within a single RXI window. Environment window operations include using commands or specially bound keys to delete, expand, and reposition Environment windows relative to the other Environment windows, all within the RXI window.

  Environment window operations are described in Environment documentation such as the *Rational Environment User's Guide* and *Rational Environment Basic Operations.*

OF MICE AND KEYBOARDS

Through an RXI window, you can communicate with the Environment using the workstation keyboard and mouse. That is, after you log into the Environment through an RXI window:

- The keys on the workstation keyboard invoke Environment commands.
- The workstation mouse can be used for positioning the Environment text cursor, for performing various Environment operations, and for displaying and selecting from pop-up menus.
- Menus are available as alternatives to the keyboard for entering many commands.

The following two sections describe the workstation mouse and keyboard in greater detail.

WORKSTATION MOUSE

The *mouse cursor* on the screen tracks the workstation mouse. Under many window managers, the location of the mouse cursor defines which workstation window is *active*—that is, which window can accept input from the keyboard. Other window managers use a click-to-type system in which you must click a mouse button while in a window to make it active.

When the mouse cursor is in an RXI window displaying the Environment, the mouse:

- Acts as a pointing device in the window
- Executes certain Environment operations directly
- Displays pop-up menus from which Environment operations can be executed

Mouse Buttons

The standard mouse can have either two or three buttons. Users typically refer to the mouse buttons as *left*, *middle*, and *right*, whereas programmers refer to the buttons as *button 1*, *button 2*, and *button 3*. On a three-button mouse, the left button is button 1, the middle button is button 2, and the right button is button 3. On a two-button mouse, the left button is button 1 and the right button is button 3, but pressing both buttons simultaneously corresponds to button 2, the "middle" button. (See Figure 1-2.) The *User's Guide* and its supplements will refer to the mouse buttons as left, middle, and right.

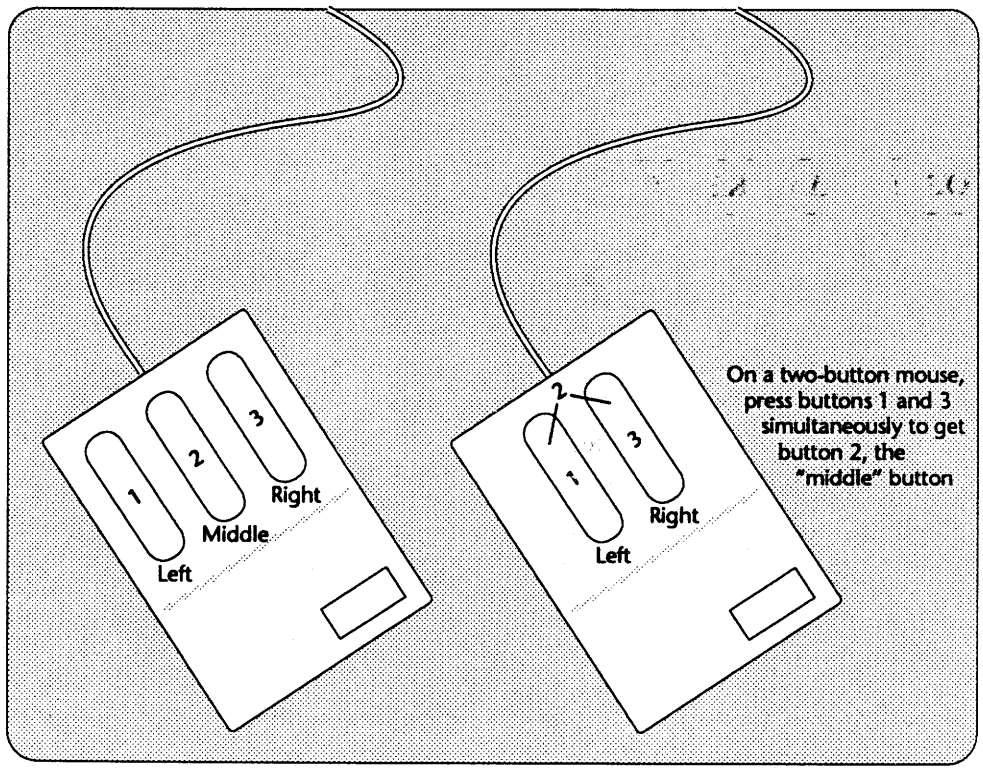

Figure 1-2 Mouse Buttons

You invoke mouse operations by pressing the appropriate mouse button(s) in one of the following three ways:

- By pressing and then releasing the button(s) once, after positioning the mouse cursor as desired. This is called *single-clicking*.
- By rapidly pressing and releasing the button(s) twice in succession, after positioning the mouse cursor as desired. This is called *double-clicking*.
- By pressing and holding the button(s), moving the mouse, and then releasing the button(s) when the mouse cursor is in the desired location. This is often called *click-and-drag* or *click-and-select*.

Modifiers and Mouse Buttons

Mouse buttons can also be used in combination with modifier keys on the keyboard. The modifier keys are [Shift], [Control], and [Meta]. (See "Workstation Keyboard," below, for the general location of the modifier keys. Also see the supplement for your workstation for an explanation of [Meta].) To use one or more modifiers with a mouse key:

1. Press and hold the appropriate modifier key(s).
2. Press the mouse buttons(s) in the appropriate manner. During click-and-drag operations, you can release the modifier key(s) at any time after pressing the mouse button(s).

Different combinations of mouse and modifier keys are bound to different operation classes. For example:

- Unshifted mouse buttons and combinations using only the [Shift] modifier are bound to operations that are available in an active RXI window only when you are logged into the Environment.
- Combinations using the [Control] or [Control] and [Shift] modifiers are bound to operations that are available in any RXI window, whether or not you are logged into the Environment.
- Combinations involving the [Meta] modifier are bound to window management operations. Note that some of these combinations also are modified by [Shift] and/or [Control].

A Cursor Is a Cursor...

The mouse cursor is distinguished from the Environment cursor by function and shape. The mouse cursor determines which window is active and indicates where mouse operations occur. The mouse cursor often changes shapes depending on its location. Within RXI windows, the mouse cursor is shaped like a capital I. In the background portion of the screen (often shaded gray), the mouse cursor is usually shaped like a capital X. When pop-up menus are displayed, the mouse cursor is shaped like an arrow. The mouse cursor may take on various other shapes during window management operations.

In contrast, the Environment text cursor is always a vertical rectangle that is solid black in an active RXI window or merely an outline in an inactive RXI window. The Environment text cursor marks the point at which keyboard input is inserted into an Environment window.

WORKSTATION KEYBOARD

In an active RXI window displaying the Environment, the keys on the workstation keyboard are defined by the Environment. The keymap supplied for your workstation can be modified to suit personal preferences or for use with other keyboards. (See Chapter 4, "Customizing RXI.") *Physical keys* are the actual keys as they exist on the keyboard; keycaps for the physical keys are often permanently marked with the physical key name. *Logical keys* are mapped from the physical keys to a logical function within the Environment; the function invoked by the key may differ from the action implied by the physical keycap marking. Logical key names are often printed on the *keyboard overlays* provided for specific terminals.

General Keyboard Layout

Environment key definitions are grouped together in specific areas on the workstation keyboard. The important groups of keys are:

- The *modifier keys*, which include [Shift], [Control], and [Meta]. Modifier keys typically are located in each of the two lower corners of the main keypad. Modifier keys are used in combination with other keys (for example, function keys) or with mouse buttons. (See the supplement for your product.)
- The four *cursor keys*, usually grouped at the lower-right side of the standard QWERTY keyboard. The cursor keys move the Environment cursor. They can be used alone or in combination with modifier or item keys. They indicate the direction of cursor movement with arrows on the keycap ([←], [→], [↑], [↓]).
- The *auxiliary keys*, which include keys immediately above the cursor keys and which are also usually to the right of the main keyboard. The physical keycap names are keyboard-dependent, but they are frequently bound to the logical keys: [Promote], [Complete], and [Format]. (See the supplement for your product.)
- The *item keys* may include keys above the auxiliary keys and in the top row of the numeric keypad, as well as other keys, depending on your specific keyboard. Item keys are used in combination with other keys (for example, with alphanumeric, punctuation, and cursor keys). Item keys denote items you frequently need to reference in Environment editing operations: words, lines, regions, and so on. These keys are referred to by their logical key names: [Object], [Region], [Word], and so on.
- The *function keys*, often labeled [F1], [F2], [F3], and so on, may be located to the left or across the top of the keyboard. These keys can be used alone or in combination with modifier keys.

Many of these keys or key combinations are bound to specific Environment commands, as indicated on the Rational keyboard overlay for your specific keyboard. (See the supplement for your product.) You can define your own keys and key combinations and change existing bindings. (See Chapter 4, "Customizing RXI.")

Executing Key Combinations

Item and modifier keys are not used alone but in combination with other keys, such as alphanumeric or function keys. (Note that modifier keys can also be used with mouse buttons. This is covered in "Modifiers and Mouse Buttons," above.)

Item-Operation Key Combinations

Key combinations made with item keys, often called *item-operation* key combinations, consist of two sequential keystrokes. That is, to execute an item-operation key combination:

1. Press and release the item key.
2. Press and release the second key (the *operation* key).

In the notation for an item-operation combination, the two keys are separated by a hyphen to remind you that the keystrokes are sequential—for example, [Window] - [D].

Modified Key Combinations

Key combinations made with modifier keys, called *modified* key combinations, consist of two or more overlapping keystrokes. That is, to execute a modified key combination:

1. Press and hold the desired modifier key or keys.
2. While holding down the modifier key(s), press the second key.

In the notation for a modified key combination, the keys are shown without a hyphen to remind you that the keystrokes overlap—for example, [Control] [D].

Combinations with different modifiers allow a family of related commands to be bound to a common modified key. For example, on many keyboards, a family of commands that provides system help is bound to [Help] and its modified key combinations:

- [Help] displays help on selected Environment objects.
- [Shift] [Help] displays an explanation of the Help facility.
- [Meta] [Help] displays help on keys.
- [Control] [Help] moves the cursor into a Help window.

Entering Numeric Keypad Arguments

Many Environment commands bound to key combinations accept numeric arguments. Numeric arguments can be entered on the numeric keypad or by using the [Control] key with the numeric keys across the top of the QWERTY layout. For example, [Word] - [D] deletes one word. The following combination deletes four words, using the numeric keypad: [Numeric 4] - [Word] - [D]. The same example using the upper numeric keys would be: [Control][4] - [Word] - [D].

Keyboard Overlay

The standard Environment key definitions are printed on cards inserted into the keyboard overlay for the workstation keyboard.

In addition to listing commands for the function keys, the keyboard overlay lists other redefined workstation keys. These keys may be illustrated with diagrams representing clusters of keys. Each key in that diagram is labeled with the physical name, as printed on the physical keycap. Above that physical name, a logical key name may also be indicated.

Function keys and auxiliary keys typically are assigned as a family of key combinations and they have related logical names, one for each combination. The logical names for the families are listed above the command names on the overlay. The bottom name in each list is the logical name for the unshifted key, and the remaining names correspond to combinations using [Shift], [Control], and [Meta].

You can modify the mapping of keys to commands, as indicated in Chapter 4. You can also update your keyboard overlay with new paper inserts to document your changes.

2 Getting Started with RXI

This chapter describes how to call up and use RXI windows to log in and log out of the Rational Environment from a workstation. Before you can invoke the RXI, you must invoke the X Window manager program for your system.

LOGGING IN

Logging into the Environment through a workstation may require you to invoke several layers of software. (You may find it helpful to refer to Figure 1-1 in Chapter 1, "Key Concepts.")

Once you have logged into the workstation and initiated the X-Windows system, you can access the Environment as follows:

1. Use the `rxi` command from any shell or command window.
2. When an RXI window opens, connect to the Environment using the `telnet` command and the network name of your Environment host machine.
3. Log into the Environment as usual.

These steps are described in greater detail in the following sections.

Where to Start

Use the `rxi` command in an X-based command window. The command assumes that the server is running and simply creates an RXI window connected to that server. You can use the command to:

- Create an initial RXI window.
- Create additional RXI windows so you can log into multiple Environment sessions.

Entering the `rxi` Command

Enter the `rxi &` command (or simply `rxi` under IBM® AIX™) in any X-based window that contains a command prompt:

```
> rxi & [Return]
```

or:

```
> rxi [Return]
```

After you press [Return], wait a few seconds. Either the mouse cursor will change shape or the RXI window will appear. At this point, some systems require you to click a mouse button to complete the operation. This creates a standard-sized Environment window.

A number of useful options are available with the `rxi` command. These are described in "rxi Command Options," below. If your system includes the "man" facility, you can enter `man rxi` at a command prompt for a complete online description of these options.

At the prompt in the RXI window, enter the `telnet` command followed by your R1000 name and press [Return] to connect to the R1000 and log into the Environment as usual.

rxi Command Options

The `rxi` command has a number of options. Two useful and important options are:

- The `-geometry` option, which allows you to set the size and position of the RXI window without using the mouse. For example, the following command creates an RXI window that is 80 columns wide and 24 lines long at coordinates (0,0)—that is, at the upper-left corner of the screen:

  ```
  > rxi -geometry 80x24+0+0
  ```

  Note that the two coordinates (`+0+0`) specify the number of pixels along the horizontal (x) and vertical (y) axes, respectively. If both coordinates are positive, distance is measured from the upper-left corner of the screen. A negative x coordinate measures distance from the right side of the screen; a negative y coordinate measures distance from the bottom of the screen. Thus, the coordinates `+0-0` position a window in the lower-left corner of the screen.
- For window managers that produce a window title bar, the `-title` option allows you to give a specific title to the RXI window. (The default title is "RXI".) This title will also identify the window if you turn it into an *icon* using the window manager, unless you separately specify an `-icontitle` option. For example, the following command creates an RXI window titled `Debug` on a typical UNIX® system:

  ```
  > rxi -title Debug &
  ```

  RXI window titles need not be unique.

Table 2-1 summarizes `rxi` command parameters and their default values. Fonts must be of fixed width and of the same size.

Table 2-1 rxi Command Options

Option	Description	Default
-132	Allows 80 <-> 132 column escape sequences	False
-ar	Autoraise option	False
-b	Inner border width	2
-bd	Border color	Black
-bg	Background color	White
-bw	Border width	2
-C	Forces output for /dev/console to appear in this window	False
-cr	Cursor color	Black
-cu	Enables a bug fix for the UNIX curses package	False
-display	Indicates which X server to use	Display Environment variable
-e	Executes a specified command	Nil
-fb	Bold font	Fixed-screen-b-11
-fg	Foreground color	Black
-fn	Normal font	Fixed-screen-r-11
-geometry	Window geometry {width}X{height}{{+-}xoffset{+-}{yoffset}}	80xmm
-i	Enables iconic startup	False
-icontitle	Enables jump scroll	False
-icongeometry	Enables logging	False
-lf	Log file	Xtermlog.XXXX
-ls	Makes the shell a login shell	False
-mb	Turns the margin bell on	False
-ms	Mouse color	Black
-name	Changes RXI's resource name	RXI
-nb	Bell margin	10
-rv	Turns reverse video on	False
-rw	Turns reverse wraparound on	False
-s	Enables asynchronous scrolling	True
-sb	Enables the scrollbar	False
-si	Disables repositioning the scrollbar at the bottom on input	False
-sk	Causes the scrollbar to position at the bottom on a key	False
-sl	Save lines	64
-sn	Status normal	False
-st	Status line	False
-title	Title (Use to change window title)	RXI
-vb	Enables visual bell	False

CUT AND PASTE IN RXI WINDOWS

You can use the mouse to cut and paste between many combinations of X-based windows. For example, if you are logged into two Environment sessions through two RXI windows, you can cut and paste between these sessions. You can also cut and paste between RXI and other X-based windows; however, the mouse operations used may differ slightly depending on whether the mouse cursor is in an RXI window or in another type of X-based application window.

In the following procedures, note that you can release the [Control] and [Shift] buttons after pressing the mouse button.

To cut a region (basic method):

1. Position the mouse cursor at the beginning of the region you want to cut.
2. To cut a region from an RXI window, press the [Control] and [Shift] modifiers and then press and hold the left mouse button.
3. Move the mouse cursor to the end of the desired region. As you move the cursor, the region you are cutting is displayed in reverse video.
4. Release the mouse button.

To cut a region (alternative method):

1. Position the mouse cursor at the beginning of the region you want to cut.
2. Define the beginning of the cut region by pressing the [Control] and [Shift] modifiers and then single-clicking the left mouse button.
3. Move the mouse cursor to the end of the desired region.
4. Define the end of the cut region by pressing the [Control] and [Shift] modifiers and then single-clicking the right mouse button.

   The cut region is displayed in reverse video.

To extend a cut region:

1. After defining a cut region using either method above, move the mouse cursor to the new end of the cut region.
2. Redefine the end of the cut region by pressing the [Control] and [Shift] modifiers and then single-clicking the right mouse button.

   The cut region is displayed in reverse video.

To paste a region:

1. Position the mouse cursor in the window and at the exact position in which you want to paste the region.
2. Press the [Control] and [Shift] modifiers and then single-click the middle mouse button.

As a result, the region is inserted into the window. In an RXI window, the region is inserted after the Environment cursor. (Note that you can paste a region into an Environment window only when the Environment object is open for editing.)

SOME USEFUL MENUS

Three control menus are available within RXI windows: the RXI X11 menu, the Modes menu, and the RXI Recognition menu. The RXI X11 menu contains menu items that control the RXI application itself. The Modes menu sets the terminal emulation modes of the RXI application. The RXI Recognition menu contains terminal-specific keyboard configurations as recognized by an R1000 system.

RXI X11 Menu

The RXI X11 menu controls various aspects of the RXI program. One important item on this menu is `Quit`, which kills the RXI application and deletes the RXI window from the screen. *Note:* The RXI X11 Menu's `Quit` does not log you out of the Environment. Log out first by using the Environment Menu's `Quit`, and then use the RXI X11 Menu's `Quit` to kill RXI.

To use the RXI X11 menu:

1. Position the cursor in an RXI window.
2. Press the [Control] modifier and press and hold the left mouse button.
3. Move the mouse cursor to the desired menu item.
4. Release the mouse button.

Modes Menu

The Modes menu controls various features of the RXI program's terminal emulation. This menu contains items for terminal emulation modes that can be toggled on or off. A checkmark next to an item indicates that the corresponding mode is currently on. For a description of the items on this menu, see the "man" pages online.

An important item on the Modes menu is the `Permanent Environment Menu` item, which keeps the Environment menu on display as long as you are logged into the Environment. When this item is on, the Environment menu automatically appears to the right of the RXI window when you log into the Environment.

To set a mode such as `Permanent Environment Menu` on the Modes menu:

1. Position the cursor anywhere in an RXI window.
2. Press the [Control] modifier and press and hold the middle mouse button.
3. Move the mouse cursor to the menu item for the desired mode.
4. Release the mouse buttons to toggle the mode's state. A checkmark indicates that the mode is presently on.

RXI Recognition Menu

The RXI Recognition menu contains all of the different predefined R1000 terminal types that can be recognized by a fully configured R1000. The menu item with a checkmark indicates the current recognition setting. At login, this is the terminal type the R1000 would recognize.

To change the terminal type that the R1000 will recognize during the next login:

1. Position the cursor in an RXI window.
2. Press the [Control] modifier and press and hold the right mouse button. The RXI Recognition menu will appear.
3. Move the mouse cursor to the desired terminal type.
4. Release the mouse button.

ENDING AN RXI SESSION

You can *close* RXI windows that you do not need, removing them from the screen altogether. You can close an RXI window in either of two ways: by entering a shell `exit` command or by using the `Quit` item from the RXI X11 menu. (See "Some Useful Menus," above.)

You may also be able to close an RXI window with your window manager menu.

To close a window using the `exit` command:

1. Log out of the Environment.
2. Make sure an operating system command prompt is displayed in the RXI window you want to close.
3. At the command prompt, enter the `exit` command.

The window is removed from the screen.

To close a window from the RXI X11 menu:

1. Position the mouse cursor anywhere in the RXI window you want to close.
2. Pop up the RXI X11 menu by pressing the [Control] modifier and then pressing and holding the left mouse button.
3. Move the mouse cursor to choose the `Quit` menu item. When chosen, this menu item is highlighted.
4. Release the mouse button.

The window is removed from the screen.

Closing an RXI window destroys any Telnet connection to the R1000, but it does not automatically log you out of the Environment. Therefore, it is recommended that you close only RXI windows in which you have not yet connected to the Environment or in which you have already logged out. Note that you can cause the Environment to log you out automatically when a disconnect signal is received from Telnet. To do this, you or your system administrator must set an Environment line-specific switch using the Terminal.Set_Logoff_On_Disconnect procedure.

3 Using the Environment through RXI

After logging into the Environment, you can use all of the usual Environment facilities to accomplish your work. Many frequently used Environment commands are bound to workstation keys. See the supplement for your system for a quick-reference list of commands and keybindings. You also can invoke Environment commands using the workstation mouse and menus.

USING THE MOUSE FOR BASIC ENVIRONMENT OPERATIONS

The mouse provides a shortcut for performing basic Environment operations such as traversal (viewing objects in the Environment's library system), selection (highlighting objects for other operations to act upon), and cursor placement (scrolling and/or moving the Environment cursor to the current location).

The following instructions refer to the left, middle, and right mouse buttons. Recall that you get the middle button on a two-button mouse by pressing both the left and right buttons simultaneously. (See Figure 3-1.)

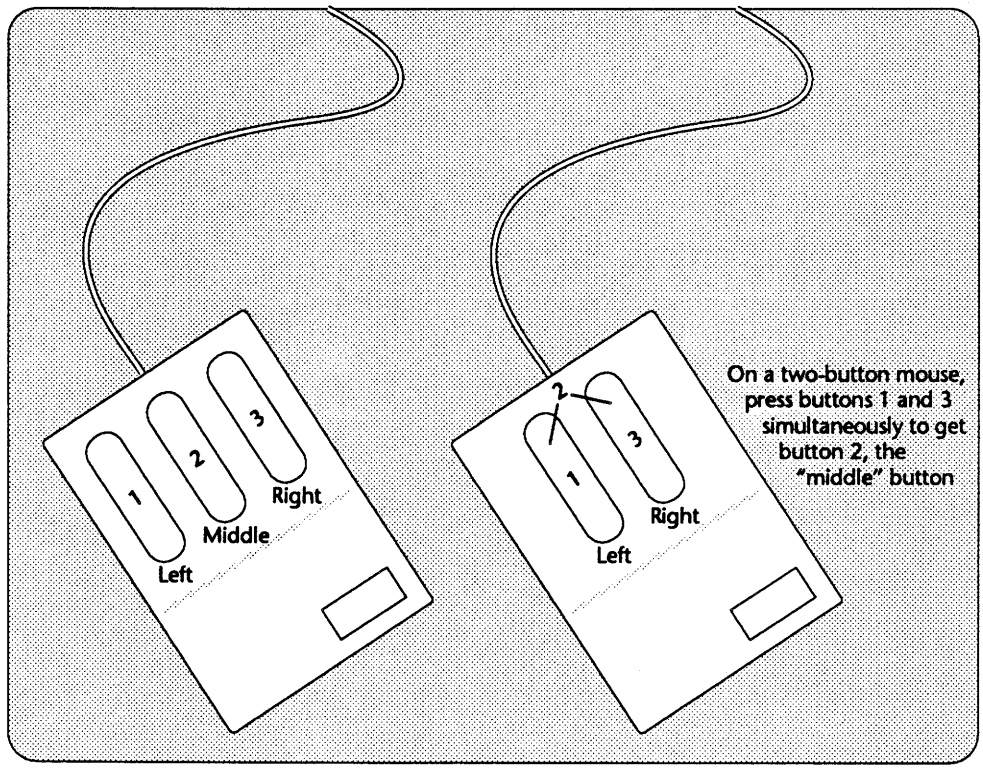

Figure 3-1 Mouse Buttons

Traversal Operations

To visit an object (for example, an object listed in an Environment library or an identifier in an Ada unit):

1. Position the mouse cursor over the name of the desired object. (The Environment cursor can be anywhere in the RXI window.)
2. Click the left mouse button once.

These steps are equivalent to positioning the Environment cursor over the object and pressing [Definition].

To visit an object's parent (for example, to visit the library that encloses a displayed Ada unit):

1. Position the mouse cursor on the object. (The Environment cursor can be anywhere in the RXI window.)
2. Click the right mouse button once.

These steps are equivalent to positioning the Environment cursor over the object and pressing [Enclosing].

Selection Operations

To select a given object or structure:

1. Position the mouse cursor anywhere in the desired object or structure. (The Environment cursor can be anywhere in the RXI window.)
2. Double-click the left mouse button.
3. Successively double-click the left mouse button to select increasingly higher-level structures around the mouse cursor.

The foregoing steps are equivalent to positioning the Environment cursor on an object and pressing the [Object] - [←] combination.

To select a smaller object or structure than the one currently selected:

1. Position the mouse cursor anywhere in the current selection. (The Environment cursor can be anywhere in the RXI window.)
2. Double-click the right mouse button.
3. Successively double-click the right mouse button to select increasingly lower-level structures around the mouse cursor.

The foregoing steps are equivalent to positioning the Environment cursor on an object and pressing the [Object] - [→] combination.

To select an arbitrary region of text:

1. Position the mouse cursor at the beginning of the desired region. (The Environment cursor can be anywhere in the RXI window.)
2. Press and hold [Shift] and then press and hold the left mouse button.
3. Move the mouse cursor to the end of the desired region.
4. Release the left mouse button and [Shift].

The foregoing steps are equivalent to positioning the Environment cursor and pressing the [Region] - [A] and [Region] - [E] combinations.

Note that the Environment cursor does not need to be in a structure in order to select it using the mouse. The Environment cursor is moved automatically into the selection by the mouse selection operation.

To turn off a selection:

1. Position the mouse cursor anywhere in the selection.
2. Double-click the middle mouse button.

The foregoing steps are equivalent to positioning the Environment cursor and pressing [Control] [X].

Environment Cursor-Placement Operations

To move the Environment cursor to any location within the RXI window:

1. Position the mouse cursor at the desired location.
2. Double-click the middle mouse button.

The foregoing steps are equivalent to using the arrow keys and/or the [Window] - [↑] and [Window] - [↓] key combinations.

To scroll an image within an Environment window:

1. Move the Environment cursor into the window you want to scroll. (Note that the mouse cursor can be anywhere in the RXI window.)
2. Press and hold [Shift] and then press and hold the right mouse button.
3. Move the mouse cursor in the direction you want to scroll. For example, to scroll up (toward the top of the object displayed in the window), move the mouse cursor toward the top of the screen. You can scroll up, down, left, or right by moving the mouse cursor accordingly.

The foregoing steps are equivalent to using [Image] - [↑], [Image] - [↓], [Image] - [←], and [Image] - [→].

Other Useful Operations

To abandon an object:

1. Position the mouse cursor on the object you want to abandon (or on the object name in the library).
2. Press the [Shift] modifier and double-click the middle mouse button.

The foregoing steps are equivalent to using the [Object] - [G] combination, which releases the object without saving changes and deletes its Environment window.

To pop up a menu for Environment operations:

1. *Either* single-click *or* press and hold the middle mouse button.
2. For details, see the following section, "Using Menus for Environment Operations."

USING MENUS FOR ENVIRONMENT OPERATIONS

You can use the mouse to *pop up* an Environment menu from which many Environment operations can be invoked. In addition, two levels of *pull-right* menus are available from many of the entries on the pop-up menu. By walking through these menus, you can invoke any command that is listed on the keyboard overlay plus a number of additional commands.

Popping Up the Environment Menu

To pop up the Environment menu, press and hold the middle mouse button. The menu titled `Environment` appears at the mouse cursor location, and the cursor shape changes to an arrow. (See Figure 3-2.)

To select a menu item, highlight the desired item by moving the mouse cursor to it and then releasing the button.

Each menu item preceded by an asterisk (*) is associated with a specific Environment command that is executed when you select that item. For example, selecting the `*Traverse` menu item executes the Common.Definition command, which is the same as pressing the [Definition] key.

Each menu item followed by a right arrow (=>) has a pull-right menu associated with it. Pull-right menus contain additional choices, typically a family of related commands. For example, the pull-right menu for `*Traverse` contains such items as `*Definition`, `*Enclosing`, and `*Other Part`, which invoke the appropriate traversal commands. The first item in the pull-right menu invokes the same command as the parent menu item.

To remove the Environment menu without executing any commands, leave the mouse cursor in the menu header and release the mouse button. Alternatively, you can move the mouse cursor out of the menu and release the mouse button.

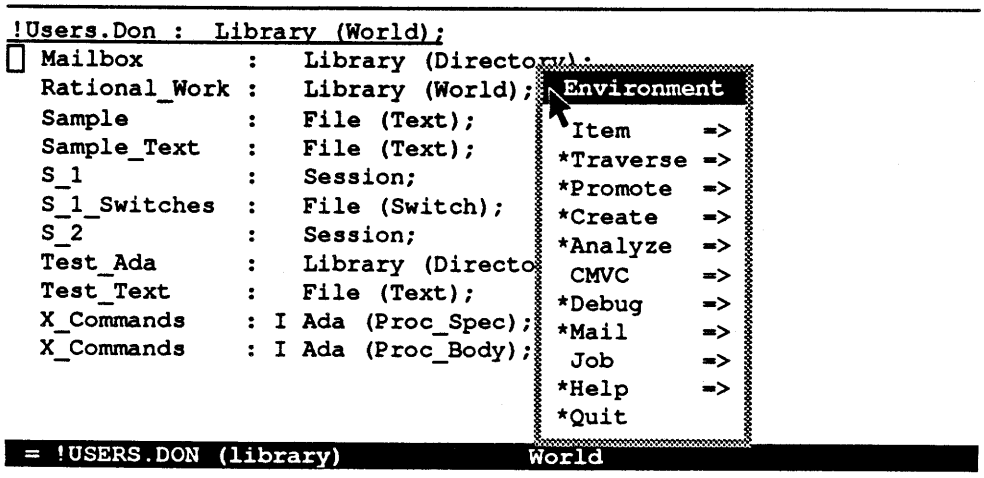

Figure 3-2 Pop-Up Environment Menu

Executing a Command from the Environment Menu

The basic steps for executing a command from the Environment menu are:

1. Pop up the Environment menu (press and hold the middle mouse button).
2. Move the mouse cursor onto the desired menu item. Only items that have an asterisk preceding them are directly executable. When you have chosen an item, a box appears around it.
3. Release the mouse button. The appropriate command is invoked and the menu disappears.

Many Environment commands are sensitive to the position of the Environment cursor. Therefore, you may need to position the Environment cursor before popping up the Environment menu. For example, to create a command window off a given Environment window:

1. Position the Environment cursor in the desired window (double-click the middle mouse button).
2. Pop up the Environment menu (press and hold the middle mouse button).
3. Move the mouse cursor onto the `*Create` menu item until a box appears around that item, as shown in Figure 3-3.
4. Release the mouse button. As a result, the menu disappears and a command window is created (as if you had pressed the [Create Command] key).

Note that some menu items, such as `*Quit`, do not execute a command directly. Instead, they open a command window containing the appropriate Environment command. You can fill in parameters, as necessary, and promote the command window using either the [Promote] key or the `*Promote` menu item.

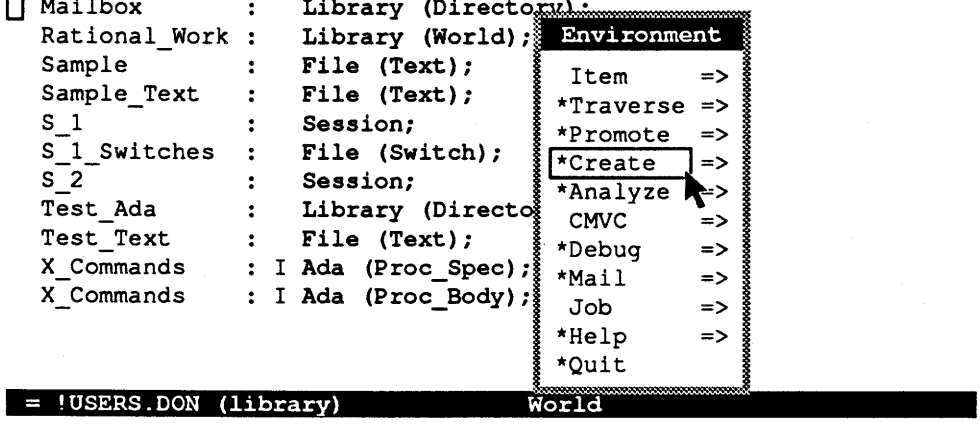

Figure 3-3 Choosing *Create on the Environment Menu

Walking Through Menus

Menu items followed by an arrow symbol (=>) have pull-right menus associated with them. You can bring up a pull-right menu by moving the cursor onto the => symbol. This is called *walking through* a menu item.

For example, to create something other than a command window, you must walk through the `*Create` menu item to display further items. In particular, to create a text file:

1. Pop up the Environment menu (press and hold the middle mouse button).
2. Move the mouse cursor to the => following the `*Create` menu item. A box appears briefly around the => and then the pull-right menu appears. The header of the pull-right menu is the same as the menu entry you walked through.
3. Move the mouse cursor to the desired menu item—for example, `*Create Text`. As before, a box appears around the chosen item. (See Figure 3-4.)
4. Release the mouse buttons. `*Create Text` opens a command window containing the Text.Create command.
5. After filling in the appropriate parameters, you can return to the Environment menu again and execute the `*Promote` item.

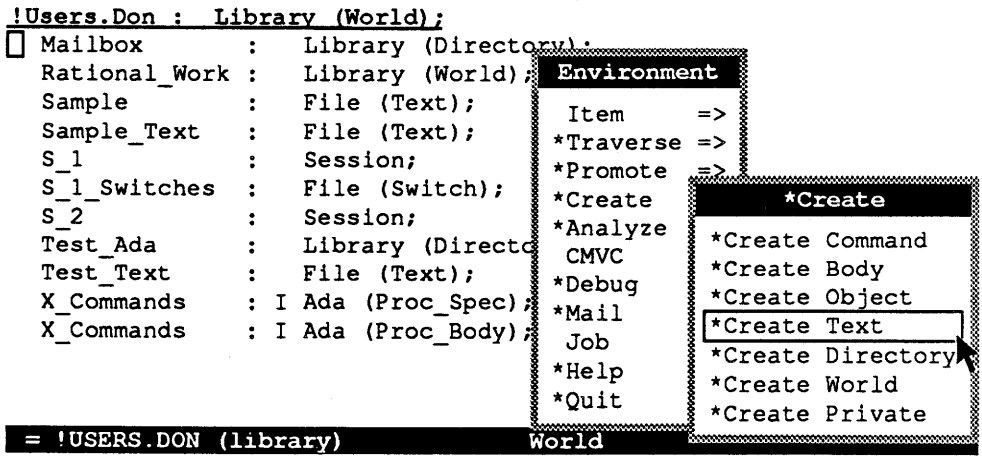

Figure 3-4 Walking Through `*Create` on the Environment Menu

For some items on the Environment menu, there are up to two levels of pull-right menus. For example, to display your home library, you can start at the `*Traverse` item and walk through two pull-right menus to choose the `*Home Library` item, as shown in Figure 3-5.

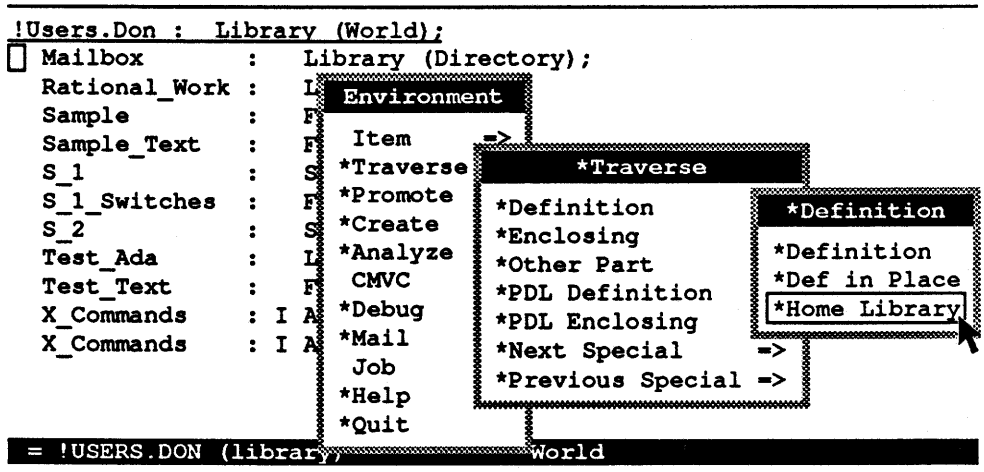

Figure 3-5 Walking Through All Three Menu Levels

Keeping Menus on Display

When you press and hold the middle mouse button, menus are visible only as long as you hold the mouse button. As soon as you release the mouse button to execute a command, the menus disappear.

Sometimes it is useful to keep menus on display, particularly when you want to invoke the same menu item multiple times. For example, if you are stepping through a program using the debugger, you can keep the pull-right menus containing `*Debug` and `*Run` on display.

To keep the Environment menu on display:

Single-click the middle mouse button (instead of pressing and holding).

To remove the Environment:

Move the mouse cursor outside the menu and single-click the middle mouse button.

To keep pull-right menus on display:

Walk through the desired menu items while the Environment menu is on display.

To remove a pull-right menu:

Move the mouse cursor from the pull-right menu back into its parent menu.

Executing Commands from Displayed Menus

While menus are on display, you can invoke menu items in several ways. The way you invoke an item determines whether menus disappear and whether commands are executed or prompted for.

To execute a command and leave the menus on display:

Choose the desired menu item (on any level of menu) and single-click the left mouse button.

To execute a command and make the menus disappear:

Choose the desired menu item (on any level of menu) and single-click the middle mouse button.

To prompt for the command associated with a menu item:

Choose the desired menu item (at any menu level) and single-click the right mouse button. This creates a command window containing the command and leaves the menus displayed.

Summary of the Environment Menu Items

Many of the items on the Environment menu are the same as the "family names" listed on the keyboard overlay for the function keys. These include Traverse, Create, Promote, Debug, Cmvc, Help. There are some differences, however, between the Environment menu and the keyboard overlay:

- The `*Analyze` item on the Environment menu combines the command families associated with the logical keys named Show and Semanticize.
- The `*Promote` item on the Environment menu combines the commands invoked by the logical keys named Promote, Demote, and Install.

The following list summarizes the kinds of menu items that are found on pull-right menus for each Environment menu entry. This list also indicates what happens when each menu item is invoked directly from the Environment menu. (Recall that "*" means "direct command.")

Item: Provides a menu-based way of entering item-operation key combinations. The first pull-right menu has an entry for each item key ([Object], [Region], [Window], and so on); the menu for each item key lists the relevant operations.

When invoked from the Environment menu: no effect.

***Traverse:** Provides commands for getting around the hierarchic Environment structure and commands for moving the cursor between underlined items or prompts.

When invoked from the Environment menu: same as [Definition].

***Promote:** Provides commands for changing Ada unit state, completion, formatting, and opening for edit.

When invoked from the Environment menu: same as [Promote].

***Create:** Provides commands for creating Environment objects such as libraries, Ada units, text files, and so on.

When invoked from the Environment menu: same as [Create Command].

***Analyze:** Provides commands for analyzing Ada programs and for getting other kinds of information (explaining error messages, showing system information, and so on).

When invoked from the Environment menu: same as [Semanticize].

CMVC: Provides commands for configuration management and version control, including operations for checking objects in and out and displaying CMVC information.

When invoked from the Environment menu: no effect.

***Debug:** Provides commands for debugging Ada programs.

When invoked from the Environment menu: invokes Debug.Current_Debugger.

***Mail:** Provides commands for displaying and replying to mail, if this optional product is installed.

When invoked from the Environment menu: same as Mail.Edit.

Job: Provides commands for managing jobs, including operations for disabling, enabling, and killing jobs, and for printing.

When invoked from the Environment menu: no effect.

***Help:** Provides commands for displaying Environment help.

When invoked from the Environment menu: same as [Help].

***Quit:** Provides commands for quitting from the Environment and setting the Ignore_Changes parameter to True (False is the default).

When invoked from the Environment menu: creates a command window containing the Quit command with Ignore_Changes set to False.

Note that menus can contain items for commands that are part of products other than the Rational Environment. For example, menu items such as `*Mail` or `*PDL Promote` point to commands from Rational Network Mail and the Rational Design Facility, respectively. If these products are not installed on the R1000 to which you are connected, their menu item names typically will not be present.

4 Customizing RXI

This chapter describes how to customize RXI by changing the defaults. You can make *systemwide customizations*, which affect all users on a shared file system, or *personal customizations*, which affect only individual users. The defaults control screen layout, window size, menu items, and keybindings.

WHERE DEFAULTS ARE DEFINED

Customizable defaults for RXI are defined both in the Environment and on the workstation. The following sections summarize where to find the definitions of these defaults.

In the Environment

In the Environment, the !Machine.Editor_Data.*<XTerminalType>*_Commands keymap procedure defines bindings for keys, mouse buttons, and menu items. You can modify this procedure to change systemwide default bindings.

Alternatively, you can create in your home library a keymap procedure named *<XTerminalType>*_Commands to define your own bindings without affecting systemwide defaults. Note that your personal *<XTerminalType>*_Commands procedure need specify only those bindings that *differ* from the systemwide bindings.

On the Workstation

The workstation contains several files that specify systemwide defaults for customizable aspects of the RXI program. These include the RXI.Xdefaults file, a systemwide resource file that defines default values for menu and mouse characteristics. This file is read by RXI to determine interactions with the X server.

You can change systemwide defaults for RXI by modifying the RXI.Xdefaults file in the RXI *source directory*. After modifying the RXI.Xdefaults file in the RXI source directory, you must execute the `make install` (UNIX) command. This command installs the file in the appropriate *systemwide default directory* under the name `RXI` (see Table 2). Although you can modify the file directly, note that it is overwritten any time you execute `make install` in the RXI source directory.

Table 2 Systemwide Default Directory

Operating System	Location
Most UNIX systems	/usr/lib/X11/app-defaults
SunOS™ with OpenLook™	$OPENWINHOME/etc
VMS™	SYS$COMMON: [DECW$DEFAULTS.SYSTEM]

You can make personal customizations by creating a .Xdefaults file in your home directory. This file is used in conjunction with the systemwide default. You can create this file by copying and modifying the contents of the RXI.Xdefaults file, or you can include in the local file only the changes you require.

In summary:

- The modifiable systemwide defaults for RXI reside in *source_directory*/RXI.Xdefaults.
- After you run `make install`, the systemwide default is written to the systemwide default directory with the filename `RXI`.
- Optional personal defaults may be in ~/.Xdefaults.

CHANGING MENU AND WINDOW CHARACTERISTICS

The `rxi` command consults a resource file to determine menu fonts and borders, the timing of double-clicks on the mouse, and so on. You can change these features by:

- Editing the RXI.Xdefaults resource file in the RXI source directory (and then entering the `make install` command). This file is always referenced first.
- Creating and editing an .Xdefaults resource file in your own home directory. If it exists, this file is read second, and conflicting values override those in RXI.Xdefaults.
- Making both kinds of change.

Regardless of its location, the resource file contains expressions setting the default values for various features. The meaningful parts of these expressions begin with an asterisk (`*`) or `RXI*`. In the user home directory file .Xdefaults the lines begin with `RXI*`; in the systemwide RXI.Xdefaults file, the lines begin with `*`. Lines in the RXI.Xdefaults file are preceded by descriptive comments (the comment symbol is `#`). For example:

```
#--What is the maximum time (in milliseconds) we should wait for a
#  double click?

*doubleClick:  250
```

You can customize features by changing the values in these expressions; for example, you can change the timing of double clicks from 250 to 300 as follows:

```
*doubleClick:  300
```

It is recommended that you inspect the RXI.Xdefaults file in the RXI source directory to familiarize yourself with the features this resource file controls. Two important changes you can make in the resource file are described in "Changing the Default RXI Window Size," and in "Adding Environment Menu Items," later in this chapter.

CHANGING THE DEFAULT RXI WINDOW SIZE

By default, when you enter the `rxi` command, you get a new window of the default size. You may want to change the default size—for example, to obtain a 60x24 window (60 columns, 24 lines). There are several ways to accomplish this:

- To change the default just for yourself, add this line to your personal .Xdefaults resource file:

  ```
  RXI*geometry: 60x24
  ```

- To change it for everyone using a particular workstation, replace the systemwide geometry specification in the RXI.Xdefaults file with this entry:

  ```
  *geometry: 60x24
  ```

- To change it for everyone using the Environment on a particular R1000, you can replace the entry for this *<XTerminalType>* in the !Machine.Editor_Data.Terminal_Types file with this entry:

  ```
  <XTerminalType> XRTERM 24 60
  ```

  (Refer to package Terminal in the System Management Utilities (SMU) book of the *Rational Environment Reference Manual* for more information about the Terminal_Types file.)

CHANGING KEYBINDINGS

As mentioned earlier, you can rebind workstation keys to different Environment commands by either of the following:

- Editing the systemwide keymap procedure, !Machine.Editor_Data.*<XTerminalType>*_Commands, and then promoting it to the installed state.
- Creating and editing a personal keymap procedure called *<XTerminalType>*_Commands in your home library. The most convenient way to create a personal keymap is to:
  1. Create an Ada unit in your home library.
  2. Copy the contents of the file !Machine.Editor_Data.*<XTerminalType>*_User_Commands into it. This file contains a skeleton keymap procedure in which you can enter appropriate bindings.
  3. Promote your keymap procedure to the installed state. These bindings will take effect the next time you log in.

Regardless of its location, the high-level structure of a keymap procedure is a case statement whose alternatives represent the three kinds of command *action*:

```
case Action is
   -- Opens a command window containing a command.
   when Prompt =>
     ...
   -- Executes the command without prompting.
   when Execute =>
     ...
```

```
   -- Interrupts a currently executing job
   -- in order to execute the command.
   when Interrupt =>
     ...
end case;
```

Nested within each high-level alternative are case statements that provide bindings for each key and key combination. For example, in the following sample systemwide keymap, the bindings for [Definition] and [Definition In Place] appear under the Execute alternative, as shown in the following partial case statement. The ellipses (...) in this fragment stand for other case statement alternatives:

```
when Execute =>
   case Key1 is
      ...
      ...
      when F5 =>
         Common.Definition (In_Place => False, Visible => True);
      when S_F5 =>
         Common.Definition (In_Place => True, Visible => True);
      ...
      ...
      when others => null;
   end case;
```

Note that these sample keybindings may not apply to your keyboard.

As indicated above, most function keys are represented in the keymap using their keycap names (for example, F5). Furthermore, modified key combinations are represented as single keys with prefixes C_ for [Control], S_ for [Shift], and M_ for [Meta]. Thus, [Definition In Place] may be represented as S_F5 for [Shift] [F5]. The enumeration type describing all key names is located in the file !Machine.Editor-_Data.<*XTerminalType*>_Key_Names.

Item-operation combinations are represented as sequences of multiple keystrokes and so are bound using further nested case statements. For example, the binding for [Window] - [↓] appears as follows:

```
when Execute =>
   case Key1 is
      ...
      when Window =>
         case Key2 is
            ...
            when Down | S_Down => Editor.Window.Next;
            ...
            when others => null;
         end case;
      ...
      when others => null;
   end case;
```

Note that item keys are represented in the keymap using their logical names.

You can change keybindings by editing the existing case statements in the systemwide keymap or by inserting the appropriate case statement alternatives into your own keymap. Your personal <*XTerminalType*>_Commands procedure need specify only those bindings that differ from the systemwide bindings.

ADDING ENVIRONMENT MENU ITEMS

You can add up to ten entries to the top-level Environment menu. New entries can be single items or complete menus. New entries will appear on the Environment menu above the standard menu items. Examples and suggestions are presented in comments within the RXI.defaults file.

Adding submenus and menu items involves two steps: defining the submenus and menu items in the appropriate .Xdefaults file, and binding the executable items (as if they were keys) to Environment commands in the appropriate *<XTerminal-Type>*_Commands procedure.

To begin, it is recommended that you specify at least one submenu in the Environment menu (or in any other menu), following the example in the resource file. As many as ten new submenus can be defined in any menu, and submenus can be nested to four levels. The new submenus are designated menu0 through menu9. For example:

```
RXI*env.menu0:    Local Menu

# transmits nothing, creates a pull-right submenu
```

Then define some items within the new menu. For example:

```
RXI*env.menu0.item0:    Printer Queue

# transmits Menu_Pick.'0'.'0'
```

The text following the colon (:) is displayed in the new menu.

If the new items are to be systemwide additions, you should add entries by editing RXI.Xdefaults in the RXI source directory. In this case, you should plan to add the corresponding Environment commands to the !Machine.Editor_Data.*<XTerminalType>*_Commands procedure.

If the new items are for a personal customization, you should edit ~/.Xdefaults in your home directory on the workstation and the *<XTerminalType>*_Commands procedure in your home library on the Environment.

Specify the Environment command that will be bound to each new item by editing the desired keymap procedure on the Environment. To continue the example:

```
 ...
when Execute =>

    case Key1 is
         ...
        when Menu_Pick =>
             ...
            case Key2 is
                 ...
                when '0' =>
                    -- These three lines replace the supplied
                    -- response "Io.Echo".
                    case Key3 is
                        when '0' =>
                            Queue.Display (Class => "all");
                        ...
```

Note that keybindings for menu items are shown with the key named `Menu_Pick` with entry numbers following. Remember to promote the edited keymap procedure to at least the installed state.

For additional help with adding submenus and items, see the commented sample additions in the RXI.Xdefaults file.

Index

A

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

W

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197

Rational X Interface User's Guide Supplement

IBM RISC System/6000 Workstation

RATIONAL

Product Number: 4000-00485

Rev. 1.0, November 1990

This document is subject to change without notice.

Note the Reader's Comments forms at the end of this book, which request the user's evaluation to assist Rational in preparing future documentation.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Contents

RXI User's Guide Supplement: IBM RISC System/6000 Workstation

This supplement is designed to augment the *Rational X Interface User's Guide*. The *User's Guide* gives conceptual information that applies to all platforms on which RXI is installed; this supplement provides material specific to running RXI on the IBM® RISC System/6000™. This supplement also applies to the IBM Xstation 120. Note that this supplement does not present information in context; rather, it assumes a familiarity with the concepts presented in the *User's Guide*.

Included in this supplement are the following kinds of information: the particular terminal type designation that is referred to with a general term in the *User's Guide*; the keyboard layout and a description of how it relates to Rational Environment™ operations; and a keymap indicating the specific keystrokes for Environment commands.

Please keep this document with the *User's Guide*. It is recommended that both documents be placed in the same binder.

RISC SYSTEM/6000 TERMINAL TYPE

When you log into the Environment from a workstation, the Environment must recognize the type of workstation you are using. Each supported workstation has a terminal type designation. For the RISC System/6000, the designation for the terminal type is Xr6us.

This designation is used in two circumstances: when logging in from nonstandard RXI windows and when using the !Machine.Editor_Data.Xr6us_Commands file. These are explained below.

Logging In

Chapter 2 of the *User's Guide*, "Logging In from Nonstandard RXI Windows," contains instructions for entering your terminal type at the appropriate prompt. Recall that the prompt for the terminal type indicates the terminal type last used. The following example indicates that the last terminal connected to the port was a Facit terminal:

```
Enter terminal type (FACIT):
```

Chapter 2 instructs you to enter your terminal type at this prompt. To log in from a RISC System/6000, enter `xr6us` (case is not significant):

```
Enter terminal type (FACIT): xr6us [Return]
```

As indicated in the *User's Guide*, you can also enter the dimensions of the window after the terminal type designation:

```
Enter terminal type (FACIT): xr6us 60 80 [Return]
```

The Xr6us_Commands File and the Terminal_Types File

Chapter 4 of the *User's Guide*, "Customizing RXI," contains instructions for changing key bindings and adding Environment menus using the file !Machine.Editor_Data.<*XTerminalType*>_Commands. To change key bindings for the RISC System/6000, edit the file Xr6us_Commands as explained in Chapter 4.

Similarly for the section "Changing the Default RXI Window Size" in Chapter 4, when you are instructed to edit the terminal-type entry in the !Machine.Editor-_Data.Terminal_Types file, edit the `xr6us` entry within that file.

KEYBOARD LAYOUT

Environment key definitions are grouped together in specific areas on the workstation keyboard, as shown in Figure 1.

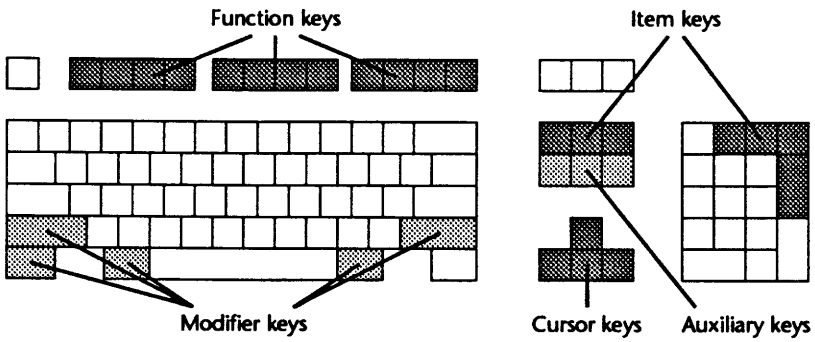

Figure 1 Layout of Environment Operations

The important groups of keys are:

- The *modifier keys*, including [Shift], [Control], and [Meta] ([Meta] is bound to the [Alt] key). Modifier keys are located in each of the two lower corners of the main keypad. Modifier keys are used in combination with other keys (for example, function keys) or with mouse buttons.
- The four *cursor keys* grouped at the lower-right side of the keyboard. The cursor keys move the Environment cursor. They can be used alone or in combination with modifier or item keys. They indicate the direction of cursor movement with arrows on the keycap ([←], [→], [↑], [↓])
- The *auxiliary keys*, including keys immediately above the cursor keys. The keycaps are labeled [Delete], [End], and [Page Down], but are usually referred to by their *logical key names*: [Promote], [Complete], and [Format].
- The *item keys*, including keys grouped with the auxiliary keys (labeled [Insert], [Home], and [Page Up]), and three more keys in the top row of the numeric keypad (labeled [/], [*], and [-]). Item keys are used in combination with other keys (for example, with alphanumeric, punctuation, and cursor keys). Item keys denote

items you frequently need to reference in Environment editing operations: words, lines, regions, and so on. These keys are referred to by their logical key names: [Object], [Region], [Window], and so on.

- The *function keys*, comprising the keys [F1] through [F12], and three additional keys across the top of the keyboard: [Print Screen], [Scroll Lock], and [Pause]. These keys can be used alone or in combination with modifier keys.

Many of these keys or key combinations are bound to specific Environment commands, as indicated on the *keyboard overlay* and in the lists below. You can define key combinations for unbound keys and you can change existing bindings (see the *User's Guide*: Chapter 4, "Customizing RXI").

BASIC KEY MAPPINGS

This section lists some of the basic key mappings for the RISC System/6000, grouped by Environment function and name. New users of RXI may find it helpful to study this list. Note that key mappings can be customized for individual users and on a systemwide basis for all Environment users. Therefore, if your system has been modified, these lists may no longer apply.

The special keys [Control], [Shift], and [Meta] are pressed simultaneously with other keys. For example, [Control] [L] indicates that you hold down the [Control] key and then press the letter [L] key. Keys meant to be pressed *in sequence* are separated by a hyphen (-). For example, [Object] - [C] denotes a sequence of two keys.

The lists below use both logical and physical key names to refer to keys. A logical key name indicates the function of the key as used in the Environment, as opposed to its label or physical keycap name. In some cases, the two are the same. The correspondence between logical keys and physical keys is shown in Table 1.

Table 1 Logical Keys and Physical Keys

Logical Key	Physical Key
[Complete]	[End]
[Control]	[Control]
[Format]	[Page Down]
[Image]	[Numeric +]
[Line]	[Numeric /]
[Mark]	[Numeric -]
[Meta]	[Alt]
[Object]	[Insert]
[Promote]	[Delete]
[Region]	[Home]
[Shift]	[Shift]
[Window]	[Page Up]
[Word]	[Numeric *]

A keyboard overlay is distributed with the software. The overlay serves as a reminder of the key bindings for your workstation. The remainder of this supplement lists the key bindings by Environment package. As mentioned earlier, keys named consecutively are pressed simultaneously, and a hyphen between key names indicates that they are pressed in sequence. In some instances, key combinations require that you hold down two modifier keys and then press a third key.

For many of the commands below, there are additional key combinations that you may use. To see these bindings online, press [Meta] [Q] for the Help On Key facility; when the prompt `Press key to be described:` appears, enter the key combination. A help window appears, containing a description of the key combinations.

KEY MAPPINGS FOR COMMANDS FROM PACKAGE EDITOR

Command Name	Physical Key	Logical Key	Alternate Key
Char.Delete_Backward	[Backspace]		
Char.Delete_Forward	[Control] [D]		
Char.Delete_Spaces	[Control] [Backspace]		
Char.Quote	[Control] [']	[Control] [Tick]	
Char.Tab_Backward	[Control] [Tab]		
Char.Tab_Forward	[Tab]		
Char.Transpose	[Control] [T]		
Cursor.Down	[Numeric /] [↓]	[Line] [↓]	[Control] [N]
Cursor.Down (8)	[Control] [Alt] [N]	[Control] [Meta] [N]	
Cursor.Left	[Control] [H]		[←]
Cursor.Left (8)	[Control] [Alt] [H]	[Control] [Meta] [H]	
Cursor.Next	[Alt] [N]	[Meta] [N]	[Meta] [↓]
Cursor.Previous	[Alt] [U]	[Meta] [U]	[Meta] [↑]
Cursor.Right	[Control] [J]		[→]
Cursor.Right (8)	[Control] [Alt] [J]	[Control] [Meta] [J]	
Cursor.Up	[Numeric /] [↑]	[Line] [↑]	[Control] [U]
Cursor.Up (8)	[Control] [Alt] [U]	[Control] [Meta] [U]	
Hold_Stack.Next	[Home] [→]	[Region] [→]	[Meta] [C]
Hold_Stack.Previous	[Home] [←]	[Region] [←]	[Region] [H]
Hold_Stack.Push	[Home] [↓]	[Region] [↓]	[Control] [C]
Hold_Stack.Top	[Home] [↑]	[Region] [↑]	[Control] [Y]
Image.Beginning_Of	[Numeric +] - [Control] [A]	[Image] - [Control] [A]	[Image] - [A]
Image.Down	[Numeric +] - [↓]	[Image] - [↓]	[Control] [V]
Image.End_Of	[Numeric +] - [Control] [E]	[Image] - [Control] [E]	[Image] - [E]
Image.Find ("")	[Numeric +] - [?]	[Image] - [?]	[Control] [F6]
Image.Left	[Numeric +] - [←]	[Image] - [←]	[Image] - [H]
Image.Right	[Numeric +] - [→]	[Image] - [→]	[Image] - [J]
Image.Up	[Numeric +] - [↑]	[Image] - [↑]	[Control] [Z]
Key.Name	[Control] [Q]		[Control] [Shift] [F6]
Line.Beginning_Of	[Numeric /] - [Control] [A]	[Line] - [Control] [A]	[Control] [A]
Line.Copy	[Numeric /] - [C]	[Line] - [C]	[Control] [Meta] [C]

Command Name	Physical Key	Logical Key	Alternate Key
Line.Delete	[Numeric /] - [D]	[Line] - [D]	[Control] [Meta] [D]
Line.Delete_Backward	[Numeric /] - [Backspace]	[Line] - [Backspace]	
Line.Delete_Forward	[Numeric /] - [K]	[Line] - [K]	[Control] [K]
Line.End_Of	[Numeric /] - [Control] [E]	[Line] - [Control] [E]	[Control] [E]
Line.Insert	[Numeric /] - [I]	[Line] - [I]	[Control] [I]
Line.Join	[Numeric /] - [J]	[Line] - [J]	[Meta] [O]
Line.Open	[Numeric /] - [O]	[Line] - [O]	[Control] [O]
Line.Transpose	[Numeric /] - [T]	[Line] - [T]	[Control] [Meta] [T]
Macro.Bind	[Numeric -] - [F]	[Mark] - [F]	[Mark] - [F5]
Macro.Execute	[Numeric -] - [Delete]	[Mark] - [Promote]	[Meta] [X]
Macro.Finish	[Numeric -] - []]	[Mark] - []]	[Mark] - [E]
Macro.Start	[Numeric -] - [[]	[Mark] - [[]	[Mark] - [A]
Mark.Next	[Numeric -] - [→]	[Mark] - [→]	[Meta] [M]
Mark.Previous	[Numeric -] - [←]	[Mark] - [←]	[Mark] - [H]
Mark.Push	[Numeric -] - [↓]	[Mark] - [↓]	[Control] [M]
Mark.Top	[Numeric -] - [↑]	[Mark] - [↑]	[Mark] - [U]
Region.Beginning_Of	[Home] - [A]	[Region] - [A]	[Region] - [Control] [A]
Region.Capitalize	[Home] - [^]	[Region] - [^]	[Region] - [']
Region.Copy	[Home] - [C]	[Region] - [C]	
Region.Delete	[Home] - [D]	[Region] - [D]	[Control] [W]
Region.End_Of	[Home] - [E]	[Region] - [E]	[Region] - [Control] [E]
Region.Finish	[Home] - []]	[Region] - []]	[Control] []]
Region.Lower_Case	[Home] - [<]	[Region] - [<]	
Region.Move	[Home] - [M]	[Region] - [M]	
Region.Off	[Home] - [X]	[Region] - [X]	
Region.Start	[Home] - [[]	[Region] - [[]	[Control] [[]
Region.Upper_Case	[Home] - [>]	[Region] - [>]	
Screen.Clear	[Alt] [L]	[Meta] [L]	
Screen.Down	[Control] [Shift] [↓]		
Screen.Left	[Control] [Shift] [←]		
Screen.Redraw	[Control] [L]		
Screen.Right	[Control] [Shift] [→]		
Screen.Up	[Control] [Shift] [↑]		
Search.Next	[Control] [S]		
Search.Previous	[Control] [R]		
Search.Replace_Next	[Alt] [S]	[Meta] [S]	
Search.Replace_Previous	[Alt] [R]	[Meta] [R]	
Set.Designation_Off	[Control] [X]		[Control] [F12]
Set.Fill_Mode (False)	[Numeric +] - [X]	[Image] - [X]	
Set.Fill_Mode (True)	[Numeric +] - [F]	[Image] - [F]	
Set.Insert_Mode (False)	[Numeric +] - [O]	[Image] - [O]	
Set.Insert_Mode (True)	[Numeric +] - [I]	[Image] - [I]	
Window.Beginning_Of	[Page Up] - [Control] [A]	[Window] - [Control] [A]	[Window] [A]
Window.Child	[Page Up] - [→]	[Window] - [→]	[Window] - [V]
Window.Copy	[Page Up] - [C]	[Window] - [C]	
Window.Delete	[Page Up] - [D]	[Window] - [D]	

Command Name	Physical Key	Logical Key	Alternate Key
Window.Demote	[Page Up] - [Y]	[Window] - [Y]	[Window] - [F9]
Window.Directory	[Page Up] - [?]	[Window] - [?]	[Window] - [F5]
Window.End_Of	[Page Up] - [Control] [E]	[Window] - [Control] [E]	[Window] - [E]
Window.Expand	[Page Up] - [I]	[Window] - [I]	
Window.Expand (-4)	[Page Up] - [.]	[Window] - [.]	
Window.Focus	[Page Up] - [F]	[Window] - [F]	[Window] - [Format]
Window.Join (-1)	[Page Up] - [Backspace]	[Window] - [Backspace]	
Window.Join (1)	[Page Up] - [J]	[Window] - [J]	
Window.Next	[Page Up] - [↓]	[Window] - [↓]	[Meta] [V]
Window.Parent	[Page Up] - [←]	[Window] - [←]	
Window.Previous	[Page Up] - [↑]	[Window] - [↑]	[Meta] [Z]
Window.Promote	[Page Up] - [Delete]	[Window] - [Promote]	[Window] - [M]
Window.Transpose	[Page Up] - [T]	[Window] - [T]	
Word.Capitalize	[Numeric *] - [^]	[Word] - [^]	[Meta] [^]
Word.Delete	[Numeric *] - [D]	[Word] - [D]	[Meta] [D]
Word.Delete_Backward	[Numeric *] - [Backspace]	[Word] - [Backspace]	[Meta] [Backspace]
Word.Delete_Forward	[Numeric *] - [K]	[Word] - [K]	[Meta] [K]
Word.End_Of	[Numeric *] - [E]	[Word] - [E]	[Meta] [E]
Word.Lower_Case	[Numeric *] - [<]	[Word] - [<]	[Meta] [<]
Word.Next	[Numeric *] - [→]	[Word] - [→]	[Meta] [→]
Word.Previous	[Numeric *] - [←]	[Word] - [←]	[Meta] [←]
Word.Transpose	[Numeric *] - [T]	[Word] - [T]	[Meta] [T]
Word.Upper_Case	[Numeric *] - [>]	[Word] - [>]	[Meta] [>]

KEY MAPPINGS FOR COMMANDS FROM PACKAGE COMMON

Command Name	Physical Key	Logical Key	Alternate Key
Abandon	[Insert] - [G]	[Object] - [G]	
Commit	[Numeric Enter]		[Object] - [Promote]
Complete	[End]		
Create_Command	[Clear display]	[F10]	
Definition	[F5]		[Control] [Meta] [→]
Demote	[Shift] [F9]		
Edit	[F9]		
Elide	[Insert] - [.]	[Object] - [.]	
Enclosing	[Page Up] [H]	[Window] [H]	[Control] [Meta] [←]
Enclosing in place	[Alt] [Shift] [F5]	[Meta] [Shift] [F5]	
Expand	[Insert] - [I]	[Object] - [I]	[Control] [I]
Explain	[Insert] - [?]	[Object] - [?]	[Control] [?]
Format	[Page Down]		
Object.Child	[Insert] - [→]	[Object] - [→]	[Control] [→]
Object.Copy	[Insert] - [C]	[Object] - [C]	
Object.Delete	[Insert] - [D]	[Object] - [D]	

Command Name	Physical Key	Logical Key	Alternate Key
Object.Insert	[Insert] - [I]	[Object] - [I]	[Control] [F10]
Object.Move	[Insert] - [M]	[Object] - [M]	
Object.Next	[Insert] - [↓]	[Object] - [↓]	[Control] [↓]
Object.Parent	[Insert] - [←]	[Object] - [←]	[Control] [←]
Object.Previous	[Insert] - [↑]	[Object] - [↑]	[Control] [↑]
Promote	[Delete]		
Redo	[Insert] - [R]	[Object] - [R]	[Object] - [V]
Release	[Insert] - [X]	[Object] - [X]	
Revert	[Insert] - [L]	[Object] - [L]	[Object] - [1]
Semanticize	[F11]		
Undo	[Insert] - [U]	[Object] - [U]	

KEY MAPPINGS FOR MISCELLANEOUS COMMANDS

Command Name	Physical Key	Logical Key	Alternate Key
Ada.Code_Unit	[Control] [F8]		
Ada.Other_Part	[Control] [F5]		
Command.Debug	[Alt] [Promote]	[Meta] [Promote]	
Job.Interrupt	[Control] [G]		
Job.Kill [0]	[Alt] [G]	[Meta] [G]	
Text.End_Of_Input	[Numeric .]		
What.Home_Library	[Control] [Alt] [F5]	[Control] [Meta] [F5]	
What.Line	[Numeric /] - [?]	[Line] [?]	

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197

Rational X Interface User's Guide Supplement

Sun Workstation

X11/NeWS

Type 4 Keyboard

RATIONAL

Product Number: 800-002699

Rev. 1.0, March 1991 (Software Release Rev10_7_2)

This document is subject to change without notice.

Note the Reader's Comments forms at the end of this book, which request the user's evaluation to assist Rational in preparing future documentation.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Contents

RXI User's Guide Supplement: Sun with X11/NeWS and Type 4 Keyboard

This supplement augments the *Rational X Interface User's Guide* and should be placed in the same binder.

- The *Rational X Interface User's Guide* gives conceptual information that applies to all platforms on which RXI is installed. You should be familiar with these concepts before reading this supplement.
- This *Rational X Interface User's Guide Supplement* provides information specific to using RXI on a Sun Workstation® running X11/NeWS™ and equipped with a Type 4 keyboard. If your workstation runs MIT X11 and has a Type 3 or a Type 4 keyboard, see the appropriate supplement.

This supplement describes the following:

- The Rational Environment™ *terminal type* for your workstation
- The keyboard layout and its relationship to Environment operations
- The specific keystrokes for executing Environment commands

XNEWS4 TERMINAL TYPE

When you log into the Rational Environment from a workstation, the Environment must recognize the type of workstation you are using. To establish this recognition, each supported workstation has an Environment terminal type associated with it (see package Terminal in the System Management Utilities (SMU) book of the *Rational Environment Reference Manual* for a discussion of terminal types).

For a Sun Workstation with MIT X11 and a Type 4 keyboard, the terminal type is xnews4.

You use this terminal type when:

- Logging in without autorecognition
- Logging in with a specially sized RXI window
- Customizing RXI

Logging In without Autorecognition

The R1000® uses autorecognition to determine the terminal type. If the R1000 cannot automatically determine the terminal type, you are prompted for it when logging in; the terminal type last used appears in parentheses. The following example shows that a Facit terminal was last connected to the port:

```
Enter terminal type (FACIT):
```

To log in, enter xnews4 (case is insignificant):

```
Enter terminal type (FACIT): xnews4 [Return]
```

Logging In with a Specially Sized RXI Window

You can override autorecognition and force the terminal type prompt to appear by entering an equals sign (=) at the username prompt and pressing [Return]:

```
Enter user name: = [Return]
```

This is useful when you want to override the default window dimensions of 77 lines by 80 columns. You can enter your preferred dimensions after the terminal type. The following example specifies dimensions of 76 lines by 150 columns:

```
Enter terminal type (FACIT): xnews4 76 150 [Return]
```

The default window dimensions are specified in the Terminal_Types file entry for xnews4, as discussed in the next subsection.

After you enter the terminal type information, the username prompt returns, and you can continue logging in as usual.

Customizing RXI

Chapter 4, "Customizing RXI," in the *Rational X Interface User's Guide* explains how to change the key mappings and the default window size.

To change key mappings for your Sun Workstation, you must edit the Ada unit !Machine.Editor_Data.Xnews4_Commands. (In the *Rational X Interface User's Guide*, the reference to this Ada unit is !Machine.Editor_Data.*<XTerminalType>*-_Commands.)

To change the default window size, you must edit the xnews4 entry found in the !Machine.Editor_Data.Terminal_Types file. The entry looks like:

```
XNews4          XRTerm  77 80
```

The first value represents the number of lines, and the second value represents the number of columns.

KEYBOARD LAYOUT

Five areas of the keyboard contain groups of Environment key definitions: left keypad, function keys, alphanumeric keys, modifier keys, and right keypad. These areas are shown in Figure 1 and described below:

- Left keypad

  The left keypad is reserved for NeWS functions, except for the [Help] key, which is bound to various Environment help operations.

- Function keys, including *auxiliary keys*

  The function keys, [F1] through [F12], are located along the top of the alphanumeric keypad. These keys are bound to groups of Environment commands: Demote, Install, Explain, Analyze, Traverse, Create, Promote, Complete, and Format. These groups are listed on the keyboard overlay (see Figure 2).

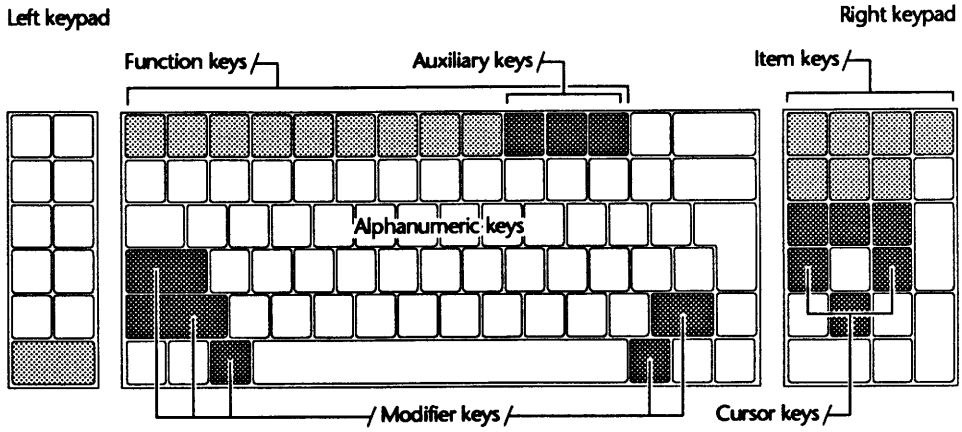

Figure 1 Key Groups on the Sun Type 4 Keyboard

Environment documentation refers to the [Promote], [Complete], and [Format] keys as *auxiliary* keys. The auxiliary keys are bound to [F10], [F11], and [F12].

- Alphanumeric keys

  The alphanumeric keys are arranged in the QWERTY layout. When used in combination with modifier or item keys, some alphanumeric keys execute Environment commands.

- Modifier keys

  The modifier keys include [Shift], [Control], and [◊] (note that [◊] corresponds to the logical key [Meta]). Modifier keys are located in the two lower corners of the alphanumeric keypad. They are used in combination with other keys or with mouse buttons to execute Environment commands.

- Right keypad: *Cursor keys*

  The four cursor keys move the Environment cursor. The direction of movement is indicated by arrows on the keycaps ([←], [→], [↑], [↓]). These keys can be used in combination with modifier or item keys to execute Environment commands.

- Right keypad: *Item keys*

  The item keys are located at the top of the right keypad. The physical key names are [Pause], [PrSc], [Scroll Lock], [Num Lock], [Right =], [Right /], and [Right *]. Environment documentation uses logical names for the item keys: [Object], [Region], [Window], [Image], [Line], [Word], and [Mark]. (See the "Logical and Physical Keys" subsection on page 4 for an explanation of logical and physical keys.) Item keys are used in combination with alphanumeric, cursor, or auxiliary keys to form [*item*] - [*operation*] key sequences.

  The numeric keys in the right keypad do not enter numeric arguments. Instead, numeric arguments must be entered with the [Control] [*alphanumeric number*] key sequence, as explained in Chapter 1 of the *Rational X Interface User's Guide.*

You can change the mapping for any key or key combination. See Chapter 4, "Customizing RXI," in the *Rational X Interface User's Guide* for details.

BASIC KEY MAPPINGS

This section lists some of the basic key mappings for your Sun Workstation, grouped by Environment function and name. Note that key mappings can be customized for individual Environment users and for all users on a system. If modified key mappings are installed on your system, the lists provided in this section may no longer apply.

Logical and Physical Keys

A physical key is the actual keyboard key. A logical key represents a key's function in the Environment. Key mappings bind the physical keys of the keyboard to the logical keys of the Environment. The logical key names are used by Environment documentation when discussing key operations. The correspondence between logical and physical keys is shown in Table 1.

Table 1 Logical and Physical Keys

Logical Key	Physical Key
Item	
[Object]	[Pause]
[Region]	[PrSc]
[Window]	[Scroll Lock]
[Image]	[Num Lock]
[Line]	[Right =]
[Word]	[Right /]
[Mark]	[Right *]
Auxiliary	
[Promote]	[F10]
[Complete]	[F11]
[Format]	[F12]
Modifier	
[Meta]	[◊]
[Shift]	[Shift]
[Control]	[Control]
Others	
[Begin Of]	[Home]
[End Of]	[PgUp]

Help on Key

You can use the Environment procedure Editor.Key.Name to see the command bound to a key combination:

1. Press [Control] [Help] to execute Editor.Key.Name. The resulting prompt is:

```
Press key to be described:
```

2. Enter the key combination. A help window appears, containing a description of the corresponding command and any alternative key combination assigned to that command.

Keyboard Overlay

A keyboard overlay is distributed with the software; it is discussed in Chapter 1 of the *Rational X Interface User's Guide*. The overlay serves as a reminder of the key mappings for the workstation. If you modify the standard key mappings or install optional layered products that redefine key mappings, you may want to update the overlay so that it reflects your configuration.

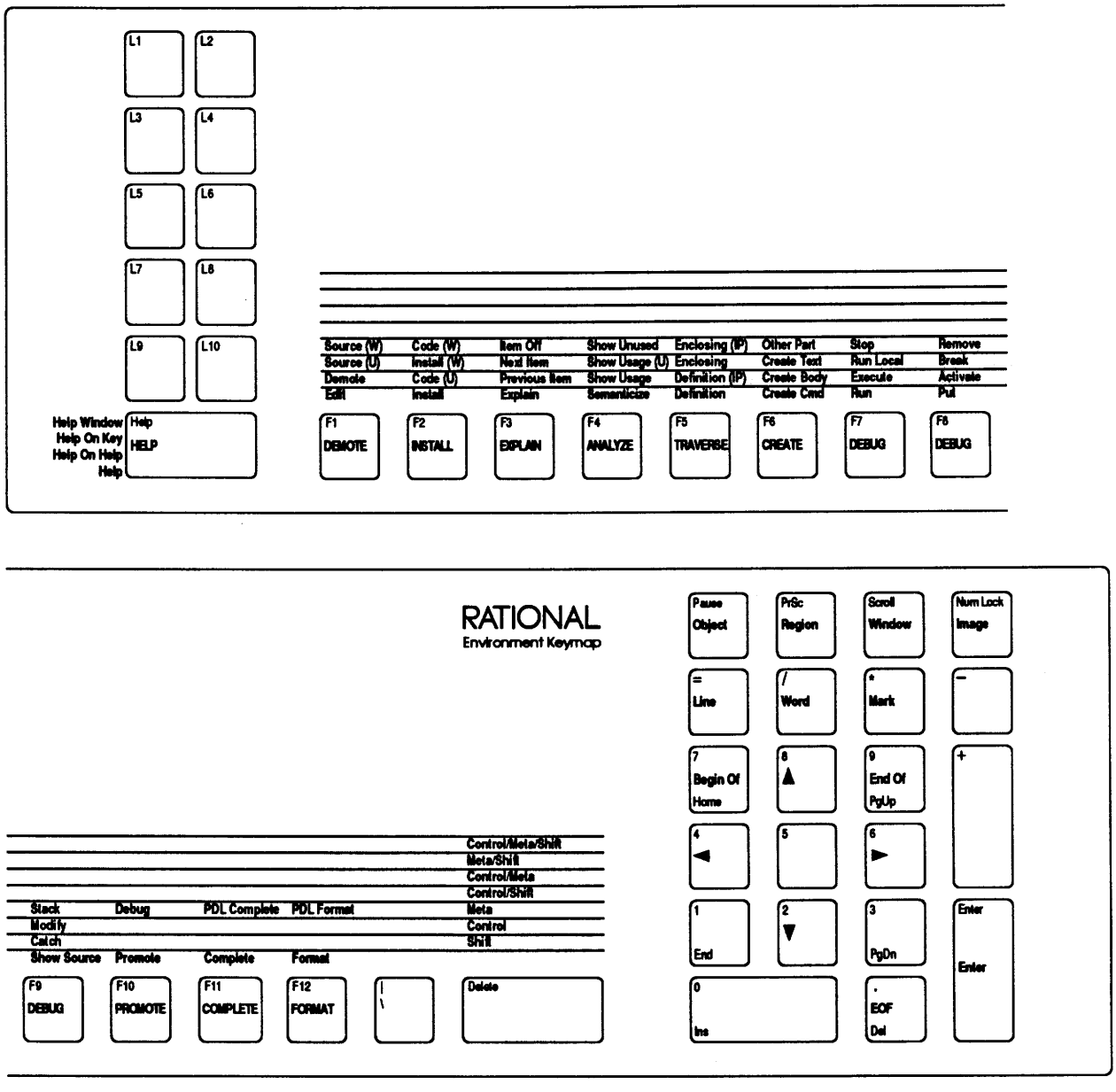

Figure 2 Type 4 Keyboard Overlay for the Sun Workstation with X11/NeWS

KEY MAPPINGS FOR ENVIRONMENT PACKAGES

This section lists the key mappings for commands from package Common, package Editor, and miscellaneous Environment packages. Locate a command under "Command Name"; the command is executed by entering the key combination listed under "Physical Key." The "Logical Key" column lists the logical key name used by the Environment and referenced in Environment documentation. Many of the commands are associated with accelerated key combinations, listed under "Alternate Key." You can use these combinations instead of those listed under "Physical Key."

Table 2 Key Notation

Notation	Description
[Key1] [Key2]	Press [Key1] and hold it down while pressing [Key2]
[Key1] - [Key2]	Press [Key1], release it, and then press [Key2]

Key Mappings for Commands from Package Common

Command Name	Physical Key	Logical Key	Alternate Key
Abandon	[Pause] - [G]	[Object] - [G]	
Commit	[Numeric Enter]	[Enter]	[Control] [Return]
Complete	[F11]	[Complete]	
Create_Command	[F6]	[Create Command]	
Definition	[F5]	[Definition]	
Definition (In_Place)	[Shift] [F5]	[Definition in Place]	
Demote	[Shift] [F1]	[Demote]	
Edit	[F1]	[Edit]	
Elide	[Pause] - [.]	[Object] - [.]	
Enclosing	[Control] [F5]	[Enclosing]	[Control] [◊] [←]
Enclosing (In_Place)	[◊] [F5]	[Enclosing in Place]	[Shift] [←]
Expand	[Pause] - [1]	[Object] - [!]	[Control] [!]
Explain	[Pause] - [/]	[Object] - [?]	[F3]
Format	[F12]	[Format]	
Object.Child	[Pause] - [→]	[Object] - [→]	[Control] [→]
Object.Copy	[Pause] - [C]	[Object] - [C]	
Object.Delete	[Pause] - [D]	[Object] - [D]	
Object.Insert	[Pause] - [I]	[Object] - [I]	
Object.Move	[Pause] - [M]	[Object] - [M]	
Object.Next	[Pause] - [↓]	[Object] - [↓]	[Control] [↓]
Object.Parent	[Pause] - [←]	[Object] - [←]	[Control] [←]
Object.Previous	[Pause] - [↑]	[Object] - [↑]	[Control] [↑]
Promote	[F10]	[Promote]	
Redo	[Pause] - [R]	[Object] - [R]	
Release	[Pause] - [X]	[Object] - [X]	
Revert	[Pause] - [L]	[Object] - [L]	
Semanticize	[F4]	[Semanticize]	
Undo	[Pause] - [U]	[Object] - [U]	

Key Mappings for Commands from Package Editor

Command Name	Physical Key	Logical Key	Alternate Key
Char.Delete_Backward	[Backspace]		[Delete]
Char.Delete_Forward	[Control] [D]		
Char.Delete_Spaces	[Control] [Backspace]		[Control] [Delete]
Char.Quote	[Control] [']		[Control] [\]
Char.Tab_Backward	[Control] [Tab]		
Char.Tab_Forward	[Tab]		
Char.Transpose	[Control] [T]		
Cursor.Down	[↓]		[Control] [N]
Cursor.Down (8)	[Control] [◊] [N]	[Control] [Meta] [N]	
Cursor.Left	[←]		[Control] [H]
Cursor.Left (8)	[Control] [◊] [H]	[Control] [Meta] [H]	
Cursor.Next	[◊] [↓]	[Meta] [↓]	[◊] [N]
Cursor.Previous	[◊] [↑]	[Meta] [↑]	[◊] [U]
Cursor.Right	[→]		[Control] [J]
Cursor.Right (8)	[Control] [◊] [J]	[Control] [Meta] [J]	[Control] [◊] [F]
Cursor.Up	[↑]		[Control] [U]
Cursor.Up (8)	[Control] [◊] [U]	[Control] [Meta] [U]	
Hold_Stack.Next	[PrSc] - [→]	[Region] - [→]	[◊] [C]
Hold_Stack.Previous	[PrSc] - [←]	[Region] - [←]	
Hold_Stack.Push	[PrSc] - [↓]	[Region] - [↓]	[Control] [C]
Hold_Stack.Top	[PrSc] - [↑]	[Region] - [↑]	[Control] [Y]
Image.Beginning_Of	[Num Lock] - [Home]	[Image] - [Begin Of]	[Shift] [Home]
Image.Down	[Num Lock] - [↓]	[Image] - [↓]	[Control] [V]
Image.End_Of	[Num Lock] - [PgUp]	[Image] - [End Of]	[Shift] [PgUp]
Image.Find ("")	[Num Lock] - [/]	[Image] - [?]	
Image.Left	[Num Lock] - [←]	[Image] - [←]	[Shift] [◊] [←]
Image.Right	[Num Lock] - [→]	[Image] - [→]	[Shift] [◊] [→]
Image.Up	[Num Lock] - [↑]	[Image] - [↑]	[Control] [Z]
Key.Name	[Control] [Help]	[Help on Key]	[Control] [Q]
Line.Beginning_Of	[Right =] - [Home]	[Line] - [Begin Of]	[Control] [B]
Line.Copy	[Right =] - [C]	[Line] - [C]	[Control] [◊] [C]
Line.Delete	[Right =] - [D]	[Line] - [D]	[Control] [◊] [D]
Line.Delete_Backward	[Right =] - [Backspace]	[Line] - [Backspace]	[Right =] - [Delete]
Line.Delete_Forward	[Right =] - [K]	[Line] - [K]	[Control] [K]
Line.End_Of	[Right =] - [PgUp]	[Line] - [End Of]	[Control] [E]
Line.Insert	[Right =] - [I]	[Line] - [I]	[Control] [I]
Line.Join	[Right =] - [J]	[Line] - [J]	[◊] [O]
Line.Open	[Right =] - [O]	[Line] - [O]	[Control] [O]
Line.Transpose	[Right =] - [T]	[Line] - [T]	[Control] [◊] [T]
Macro.Bind	[Right *] - [F]	[Mark] - [F]	
Macro.Execute	[Right *] - [F10]	[Mark] - [Promote]	[◊] [X]
Macro.Finish	[Right *] - [)]	[Mark] - [End Of]	[◊] [)]
Macro.Start	[Right *] - [(]	[Mark] - [Begin Of]	[◊] [(]

Command Name	Physical Key	Logical Key	Alternate Key
Mark.Next	[Right *] - [→]	[Mark] - [→]	[◊] [M]
Mark.Previous	[Right *] - [←]	[Mark] - [←]	
Mark.Push	[Right *] - [↓]	[Mark] - [↓]	[Control] [M]
Mark.Top	[Right *] - [↑]	[Mark] - [↑]	
Region.Beginning_Of	[PrSc] - [Home]	[Region] - [Begin Of]	
Region.Capitalize	[PrSc] - [^]	[Region] - [^]	[PrSc] - [']
Region.Copy	[PrSc] - [C]	[Region] - [C]	
Region.Delete	[PrSc] - [D]	[Region] - [D]	[Control] [W]
Region.End_Of	[PrSc] - [PgUp]	[Region] - [End Of]	
Region.Finish	[PrSc] - []]	[Region] - []]	[Control] []]
Region.Lower_Case	[PrSc] - [<]	[Region] - [<]	[PrSc] - [,]
Region.Move	[PrSc] - [M]	[Region] - [M]	
Region.Off	[PrSc] - [X]	[Region] - [X]	
Region.Start	[PrSc] - [[]	[Region] - [[]	[Control] [[]
Region.Upper_Case	[PrSc] - [>]	[Region] - [>]	[PrSc] - [.]
Screen.Clear	[◊] [L]	[Meta] [L]	
Screen.Down	[Control] [Shift] [↓]		
Screen.Left	[Control] [Shift] [←]		
Screen.Redraw	[Control] [L]		
Screen.Right	[Control] [Shift] [→]		
Screen.Up	[Control] [Shift] [↑]		
Search.Next	[Control] [S]		
Search.Previous	[Control] [R]		
Search.Replace_Next	[◊] [S]	[Meta] [S]	
Search.Replace_Previous	[◊] [R]	[Meta] [R]	
Set.Designation_Off	[◊] [F3]		[Control] [X]
Set.Fill_Mode (False)	[Num Lock] - [X]	[Image] - [X]	
Set.Fill_Mode (True)	[Num Lock] - [F]	[Image] - [F]	
Set.Insert_Mode (False)	[Num Lock] - [O]	[Image] - [O]	
Set.Insert_Mode (True)	[Num Lock] - [I]	[Image] - [I]	
Window.Beginning_Of	[Scroll Lock] - [Home]	[Window] - [Begin Of]	[Control] [◊] [Home]
Window.Child	[Scroll Lock] - [→]	[Window] - [→]	
Window.Copy	[Scroll Lock] - [C]	[Window] - [C]	
Window.Delete	[Scroll Lock] - [D]	[Window] - [D]	
Window.Demote	[Scroll Lock] - [F1]	[Window] - [Demote]	[Scroll Lock] - [Y]
Window.Directory	[Scroll Lock] - [F5]	[Window] - [Definition]	[Scroll Lock] - [/]
Window.End_Of	[Scroll Lock] - [PgUp]	[Window] - [End Of]	[Control] [◊] [PgUp]
Window.Expand	[Scroll Lock] - [!]	[Window] - [!]	[Scroll Lock] - [1]
Window.Expand (-4)	[Scroll Lock] - [.]	[Window] - [.]	
Window.Focus	[Scroll Lock] - [F12]	[Window] - [Format]	[Scroll Lock] - [F]
Window.Join (-1)	[Scroll Lock] - [Backspace]	[Window] - [Backspace]	
Window.Join (1)	[Scroll Lock] - [J]	[Window] - [J]	
Window.Next	[Scroll Lock] - [↓]	[Window] - [↓]	[◊] [V]
Window.Parent	[Scroll Lock] - [←]	[Window] - [←]	
Window.Previous	[Scroll Lock] - [↑]	[Window] - [↑]	[◊] [Z]
Window.Promote	[Scroll Lock] - [F10]	[Window] - [Promote]	[Scroll Lock] - [Enter]
Window.Transpose	[Scroll Lock] - [T]	[Window] - [T]	

Command Name	Physical Key	Logical Key	Alternate Key
Word.Beginning_Of	[Right /] - [Home]	[Word] - [Begin Of]	[◊] [B]
Word.Capitalize	[Right /] - [^]	[Word] - [^]	[◊] [6]
Word.Delete	[Right /] - [D]	[Word] - [D]	[◊] [D]
Word.Delete_Backward	[Right /] - [Backspace]	[Word] - [Backspace]	[◊] [Backspace]
Word.Delete_Forward	[Right /] - [K]	[Word] - [K]	[◊] [K]
Word.End_Of	[Right /] - [PgUp]	[Word] - [End Of]	[◊] [E]
Word.Lower_Case	[Right /] - [<]	[Word] - [<]	[◊] [,]
Word.Next	[Right /] - [→]	[Word] - [→]	[◊] [J]
Word.Previous	[Right /] - [←]	[Word] - [←]	[◊] [H]
Word.Transpose	[Right /] - [T]	[Word] - [T]	[◊] [T]
Word.Upper_Case	[Right /] - [>]	[Word] - [>]	[◊] [.]

Key Mappings for Commands from Miscellaneous Packages

Command Name	Physical Key	Logical Key	Alternate Key
Ada.Code_Unit	[Shift] [F2]		
Ada.Other_Part	[◊] [F6]		
Command.Debug	[◊] [F10]	[Meta] [Promote]	
Job.Interrupt	[Control] [G]		
Job.Kill [0]	[◊] [G]	[Meta] [G]	[Right =] - [Numeric Del]
Text.End_Of_Input	[Right .]		
What.Line	[Right =] - [?]	[Line] - [?]	[Right =] - [/]

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197

RATIONAL

READER'S COMMENTS

Note: This form is for documentation comments only. You can also submit problem reports and comments electronically by using the SIMS problem-reporting system. If you use SIMS to submit documentation comments, please indicate the manual name, book name, and page number.

Did you find this book understandable, usable, and well organized? Please comment and list any suggestions for improvement.

_____

_____

_____

_____

_____

_____

If you found errors in this book, please specify the error and the page number. If you prefer, attach a photocopy with the error marked.

_____

_____

_____

_____

Indicate any additions or changes you would like to see in the index.

_____

_____

_____

_____

How much experience have you had with the Rational Environment?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

How much experience have you had with the Ada programming language?

❑ 6 mo. or less ❑ 6 mo.–1 year ❑ 1–3 years ❑ 3 years or more

Name (optional) _____ Date _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Please return this form to:

Publications Department
RATIONAL
3320 Scott Boulevard
Santa Clara, CA 95054-3197