

Rational Environment
Basic Keymap

Rational Terminal

Copyright © 1985, 1986, 1987 by Rational

Document Control Number: 8001A-03 (803-002317)

Rev. 6.0, November 1985
Rev. 6.1, March 1986
Rev. 6.2, July 1986
Rev. 7.0, July 1987 (Delta)

This document subject to change without notice.

Note the Reader's Comments form on the last page of this book, which requests the user's evaluation to assist Rational in preparing future documentation.

Ada is a registered trademark of the U.S. Government (Ada Joint Program Office).

Rational and R1000 are registered trademarks and Rational Environment and Rational Subsystems are trademarks of Rational.

VT100 is a trademark of Digital Equipment Corporation.

Rational
1501 Salado Drive
Mountain View, California 94043

Contents

How to Use the Basic Keymap	1
Keymap Overview	1
Quick Reference to Key Bindings	1
Detailed Reference to Key Bindings	1
Master Reference to Key Bindings by Command	1
Environment Key Combinations	2
Item-Operation Key Combinations	2
Patterns among Item-Operation Combinations	2
Modified Key Combinations	3
Patterns among Modified Key Combinations	3
Basic and Accelerated Keystrokes	4
Keymap Notation	4
Symbols	4
Numeric Arguments	4
Case Sensitivity of Key Bindings	4
Quick Reference to Key Bindings	5
Getting Help	6
Traversing the Environment	6
Logging Off	6
Selecting Items	6
Executing Commands	6
Managing Windows	7
Moving within an Image	7
Writing Text Files	7
General Editing Operations	8
Writing Ada Programs	8
Debugging Ada Programs	9
Managing Libraries	9

Using CMVC	9
Managing Links	9
Using Environment I/O Resources	10
Managing Jobs	10
Detailed Reference to Key Bindings	11
Getting Help and Other Information	12
Traversing the Environment	12
Logging Off	12
Selecting Items	13
Executing Commands	13
Managing Windows	13
Moving between Windows	13
Resizing and Repositioning Windows	14
Redrawing the Screen	14
Retaining Windows	14
Removing Windows	14
Finding Windows	14
Moving within an Image	15
By Character	15
By Word	15
By Underline or Prompt	15
By Line	15
In a Region	16
By Tabs	16
By Scrolling	16
By Marking Your Place	16
General Editing Operations	17
Selecting an Arbitrary Region	17
Moving and Copying Text	17
Deleting Text	17
Searching and Replacing Text	17
Entering Text	18
Transposing Text	18
Controlling Case	18
Holding and Retrieving Text	19
Formatting Text	19
Writing Text Files	20

Accessing Text Files	20
Saving Changes	20
Terminating Edit	20
Selecting Substructures within Text	20
Writing Ada Programs	21
Creating Ada Programs	21
Accessing Ada Programs	21
Saving Changes and Terminating Edit	21
Checking for Errors	21
Changing the Compilation State	22
Changing to a Higher Compilation State	22
Changing to a Lower Compilation State	22
Selecting Structures within Ada Programs	22
Modifying Ada Programs	23
Entering Comments and Special Strings	23
Browsing Ada Programs	23
Checking Using Occurrences	23
Debugging Ada Programs	24
Stepping and Executing	24
Setting and Removing Breakpoints	24
Viewing Stacks	24
Displaying and Modifying Variables	24
Handling Exceptions	24
Managing Libraries	25
Creating Libraries	25
Manipulating Objects in Libraries	25
Controlling Library Display	25
Using CMVC	25
Managing Links	26
Accessing Links	26
Removing the Link Editor	26
Selecting Links	26
Modifying Links	26
Traversing Linked Ada Units	26
Controlling the Display	26
Managing Searchlists	27
Accessing the Searchlist	27
Removing the Searchlist Editor	27

Selecting Entries	27
Modifying the Searchlist	27
Using Keyboard Macros	28
Remembering and Redisplaying Screens	28
Using Environment I/O Resources	28
Managing Jobs	28
Master Reference to Key Bindings by Command	29

How to Use the Basic Keymap

The Rational Environment Basic Keymap is designed to acquaint new users with the keys that have been bound to Environment commands. Users have the option of modifying these key bindings for their own use, following procedures described in Rational Environment Basic Operations, also in this manual.

Note that there is a more complete reference to Environment key bindings in the Rational Environment Keymap, in Volume 1 of the *Rational Environment Reference Manual*. It is intended as the primary key reference for Environment users.

Keymap Overview

The Keymap has been divided into the following three sections. The first two sections apply to the Rational Terminal only. The last section includes key bindings for both the Rational Terminal and the VT100TM terminal.

Quick Reference to Key Bindings

The Quick Reference is a guide to the most commonly used key combinations, organized by topic. The Quick Reference entry for each key combination includes:

- A brief description of what the combination does
- The full name of the command that is bound to it

Detailed Reference to Key Bindings

The Detailed Reference provides a nearly complete list of key combinations, organized by topic and subtopic. The Detailed Reference entry for each key combination includes:

- A brief description of what the combination does
- The full name of the command that is bound to it
- Alternative key bindings, including accelerated key combinations (see "Basic and Accelerated Keystrokes," below)

Master Reference to Key Bindings by Command

This section provides a complete, alphabetic list of the commands that are bound to keys on both the Rational Terminal and the VT100 terminal. Each entry includes:

How to Use the Basic Keymap

- The full name of an Environment command
- The key combination(s) to which the command is bound on the Rational Terminal
- The key combination(s) to which the command is bound on the VT100 terminal

Environment Key Combinations

Environment commands are bound to two types of key combinations:

- Item-operation combinations
- Modified key combinations

These two types of key combinations differ in how they are executed.

Item-Operation Key Combinations

Each item-operation key combination contains an item key (**Object**, **Region**, **Window**, **Image**, **Line**, **Word**, or **Mark**) followed by an operation key (either alphabetic or nonalphabetic). The item key identifies the item affected by the operation; the operation key identifies the action that applies to the indicated item.

The keystrokes must be sequential in an item-operation key combination. To execute an item-operation key:

1. Press and release the item key.
2. Press and release the operation key.

The notation indicates sequential keystrokes by separating them with a hyphen:

item key - **operation key**.

Patterns among Item-Operation Combinations

In general, commands that execute similar operations are bound to combinations that contain a common operation key. Some examples include:

item - **C** Commands that copy items are bound to combinations such as **Line** - **C**, **Region** - **C**, and **Object** - **C**, which share the operation key **C**.

item - **D** Commands that delete items are bound to combinations such as **Line** - **D**, **Word** - **D**, and **Window** - **D**, which share the operation key **D**.

item - **T** Commands that transpose items are bound to combinations such as **Word** - **T**, **Line** - **T**, and **Window** - **T**, which share the operation key **T**.

Modified Key Combinations

Each modified key combination contains one or more modifier keys (**Shift**, **Control**, **Meta**), along with another key (either alphabetic or nonalphabetic). Modifier keys are never used with item keys.

The keystrokes must overlap in a modified key combination. To execute a modified combination:

1. Press and hold the modifier key(s).
2. While holding down the modifier key(s), press the key to be modified.

The notation indicates overlapping keystrokes by naming the keys adjacently:

modifier key **other key**.

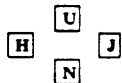
Patterns among Modified Key Combinations

Many commands that operate on characters are bound to modified combinations involving the **Control** key. In many cases, the modified key also serves as the operation key in one of the item-operation patterns (see above). For example, **Control****D** deletes a character and **Control****T** transposes two characters.

Several patterns hold among combinations of modifiers and arrow keys, as shown:

(no modifier)	Simple cursor movement within an image
Shift	Scrolling an image (up, down) or traversal in place (left, right)
Control	Selecting an object
Control Shift	Screen cursor movement
Meta	Word (left, right) or item movement (up, down)
Control Meta	Library (left, right) or window movement (up, down)
Meta Shift	Scrolling an image (left, right)
Control Meta Shift	Screen stack operations

Note that operations for simple cursor movement have been bound to combinations involving the **Control** key to form a convenient diamond pattern on the primary keyboard. This pattern matches the pattern of the arrow keys:



How to Use the Basic Keymap

Basic and Accelerated Keystrokes

Certain key combinations (namely, item-operation combinations and modified function keys) are considered *basic* combinations because they involve explicitly labeled keys, such as `Word` or `Definition`. Basic key bindings are recommended if you are new to the Environment, because they are easy to remember.

However, experienced users may find *accelerated* key bindings more convenient. Accelerated bindings generally involve the modifier keys in combination with keys on the main keyboard so that you can use them without moving your hands away from normal typing position.

Many commands are bound to both basic and accelerated key combinations. As an example, you can delete a word using either `Word - D` or the corresponding accelerated key combination, `Meta D`.

Keymap Notation

The following notations apply to all sections of the Keymap except the "Master Reference to Key Bindings by Command."

Symbols

`key1 - key2` Press and release `key1`; then press `key2`.

`key1 key2` Press and hold `key1` while pressing `key2`.

`numeric 1` Press `1` on the numeric keypad.

Numeric Arguments

You can give a numeric argument to many of the commands that are bound to keys. Indicate the desired number using the numeric keypad, and then press the key combination bound to the command. For example, `Word - D` deletes one word; the following combination deletes four words: `numeric 4 - Word - D`.

Indicate negative numbers by pressing `numeric -` first. For example, the following combination shrinks a window by seven lines ("expands" it by `-7` lines):

`numeric - - numeric 7 - Window - 1`

Case Sensitivity of Key Bindings

Although keys are shown as uppercase, the unshifted equivalent also works. This is true for the nonalphabetic characters as well. For example, `Object - d` is equivalent to `Object - D` and `Object - 1` is equivalent to `Object - 1`.

Quick Reference to Key Bindings

Getting Help

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Determine what help is available	Help on Help	What.Does
Get help on item	Help	What.Does
Get help on key	Help on Key	Editor.Key.Name
Display Help window	Help Window	Editor.Image.Find

Traversing the Environment

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
View object cursor is on	Definition	Common.Definition
Get to parent object	Enclosing	Common.Enclosing
Get to your home library	Home Library	What.Home-Library

Logging Off

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Log off, unless changes aren't saved	-	Editor.Quit
Log off, ignoring unsaved changes	-	Editor.Quit(True)

Selecting Items

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Select successively larger structures	Object - --	Common.Object.Parent
Select successively smaller structures	Object - --	Common.Object.Child
Select previous structure, same level	Object - ↑	Common.Object.Previous
Select next structure, same level	Object - ↓	Common.Object.Next
Turn off selection cursor is in	Item Off	Editor.Set.Designation-Off

Executing Commands

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Create a Command window	Create Command	Common.Create-Command
Complete command name and parameters	Complete	Common.Complete
Execute a command	Promote	Common.Promote
Move to the next parameter	Next Item	Editor.Cursor.Next
Move to the previous parameter	Previous Item	Editor.Cursor.Previous
Turn a prompt into text	Item Off	Editor.Set.Designation-Off
Redisplay the previous command (undo)	Object - U	Common.Undo
Redisplay the next command (redo)	Object - R	Common.Redo

Managing Windows

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Move to the next window	Window - I	Editor.Window.Next
Move to the previous window	Window - ↑	Editor.Window.Previous
Join with the next window	Window - J	Editor.Window.Join (1)
Transpose current window with previous	Window - T	Editor.Window.Transpose
Realign windows	Window - Format	Editor.Window.Focus
Redraw the screen	Control L	Editor.Screen.Redraw
Lock a window on the screen	Window - Promote	Editor.Window.Promote
Release a locked window	Window - Demote	Editor.Window.Demote
Remove a window temporarily	Window - D	Editor.Window.Delete
Release image permanently, saving changes	Object - X	Common.Release
Display the Window Directory	Window - Definition	Editor.Window.Directory
View Window Directory entry cursor is on	Definition	Common.Definition
Delete selected Window Directory entry	Object - D	Common.Object.Delete

Moving within an Image

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Move to beginning of line	Begin Of	Editor.Line.Beginning_Of
Move to end of line	End Of	Editor.Line.End_Of
Scroll up	Image - ↑	Editor.Image.Up
Scroll down	Image - ↓	Editor.Image.Down
Scroll to top of image	Image - Begin Of	Editor.Image.Beginning_Of
Scroll to end of image	Image - End Of	Editor.Image.End_Of

Writing Text Files

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Create a new text file	Create Text	Text.Create
View existing text file	Definition	Common.Definition
Edit existing text file	Edit	Common.Edit
Revert to last saved version	Object - L	Common.Revert
Save, leaving open for editing	Enter	Common.Commit
Save, making read only	Promote	Common.Promote

General Editing Operations

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Select start of region	Region - I	Editor.Region.Start
Select end of region	Region - J	Editor.Region.Finish
Copy a selected item	Region - C	Editor.Region.Copy
Move a selected item	Region - M	Editor.Region.Move
Delete character — forward	Control D	Editor.Char.Delete_Forward
Delete character — backward	Delete	Editor.Char.Delete_Backward
Delete word	Word - D	Editor.Word.Delete
Delete line	Line - D	Editor.Line.Delete
Delete selected item	Region - D	Editor.Region.Delete
Search for next occurrence	Control S	Editor.Search.Next
Replace next occurrence	Meta S	Editor.Search.Replace_Next

Writing Ada Programs

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Create an Ada unit in library	Create Ada	Common.Object.Insert
Build a body	Create Body	Ada.Create_Body
Build a private part	Create Private	Ada.Create_Private
Demote to source, open for editing	Edit	Common.Edit
Revert to last saved version	Object - L	Common.Revert
Save, leaving open for editing	Enter	Common.Commit
Save, regardless of errors	Enter	Common.Commit
Complete and check syntax	Format	Common.Format
Check for semantic errors	Semanticize	Common.Semanticize
Explain underlined error	Explain	Common.Explain
Move to next underlined error	Next Item	Editor.Cursor.Next
Move to previous underlined error	Previous Item	Editor.Cursor.Previous
Promote to next higher state	Promote	Common.Promote
Change to source state	Source Unit	Ada.Source_Unit
Change to installed state	Install Unit	Ada.Install_Unit
Change to coded state	Code Unit	Ada.Code_Unit
Demote to next lower state	Demote	Common.Demote
Compile unit and those it depends on	Code (This World)	Compilation.Make
Demote units and its dependents	Source (This World)	Compilation.Demote
Get to other part of Ada unit	Other Part	Ada.Other_Part

Debugging Ada Programs

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Execute program with Debugger on	Meta Promote	Command.Debug
Get to Debugger window	Debugger Window	Debug.Current-Debugger
Continue program execution	Execute	Debug.Execute
Step one statement	Run	Debug.Run
Step one statement at same level	Run Local	Debug.Run (Local)
Display values of variables	Put	Debug.Put
Set breakpoints	Break	Debug.Break
Display breakpoints	Show Breaks	Debug.Show

Managing Libraries

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Create a world	Create World	Library.Create-World
Create a directory	Create Directory	Library.Create-Directory
Delete selected object from library	Object - D	Common.Object.Delete
Print image or selected object	Print	Queue.Print
Toggle information in library display	Explain	Common.Explain
Show access list for designated object	Show Access List	Access-List.Display

Using CMVC

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Check out designated object	Check Out	Cmvc.Check-Out
Check in designated object	Check In	Cmvc.Check-In
Accept changes for designated object	Accept Changes	Cmvc.Accept-Changes
Show objects that are checked out		
In this view	Checked Out In View	Cmvc.Show-Checked-Out-In-View
By you, any view	Checked Out By User	Cmvc.Show-Checked-Out-By-User
Show info about designated object	Show Info	Cmvc.Show
Show out-of-date objects in this view	Show Out Of Date	Cmvc.Show-Out-Of-Date-Objects

Managing Links

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
List links	-	Links.Display
Add a new link	-	Links.Add

Using Environment I/O Resources

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Indicate end of input to program	End Of Input	Text.End_Of_Input
Commit interactive input	Promote	Common.Promote

Managing Jobs

<i>Description</i>	<i>Basic Keys</i>	<i>Command</i>
Disconnect job from terminal	Control G	Job.Interrupt
Kill job	Job Kill	Job.Kill(0)

Detailed Reference to Key Bindings

Getting Help and Other Information

<i>Description</i>	<i>Basic Keys</i>	<i>Accelerated Keys</i>	<i>Command</i>
Determine what help is available	Help on Help		What.Does
Get help on item	Help		What.Does
Get help on key	Help on Key	Control Q	Editor.Key.Name
Display Help window	Help Window		Editor.Image.Find
Explain underlined error	Explain	Control ?, Object - ?	Common.Explain
Show time and date	What Time		What.Time
Show system load	What Load		What.Load (True)
Show current users	What Users		What.Users (True)
Show lock information for object in window	What Locks		What.Locks
Show full name of object in window	What Object		What.Object
Show access list for designated object	Show Access List		Access-List.Display

Traversing the Environment

<i>Description</i>	<i>Basic Keys</i>	<i>Accelerated Keys</i>	<i>Command</i>
Display the Window Directory	Window - Definition	Window - ?	Editor.Window.Directory
Display object cursor is on	Definition	Control Meta -	Common.Definition
Display object, same window	Definition In Place	Shift -	Common.Definition
Display parent object	Enclosing	Control Meta -	Common.Enclosing
Display parent object, same window	Enclosing In Place	Shift -	Common.Enclosing
Display parent library, same window	Enclosing Library		Common.Enclosing
Display your home library	Home Library		What.Home-Library
Set mark at current location	Mark -	Control M	Editor.Mark.Push
Cycle through marks in stack	Mark - -	Meta M	Editor.Mark.Next
Cycle back through marks in stack	Mark - -		Editor.Mark.Previous
Return to most recent mark	Mark - ↑		Editor.Mark.Top

Logging Off

<i>Description</i>	<i>Basic Keys</i>	<i>Accelerated Keys</i>	<i>Command</i>
Log off, unless changes aren't saved	-	-	Editor.Quit
Log off, ignoring unsaved changes	-	-	Editor.Quit(True)

Selecting Items

Description	Basic Keys	Accelerated Keys	Command
Select successively larger structures	Object . --	Control --	Common.Object.Parent
Select successively smaller structures	Object . --	Control --	Common.Object.Child
Select previous structure, same level	Object . ↑	Control ↑	Common.Object.Previous
Select next structure, same level	Object . ↓	Control ↓	Common.Object.Next
Select first structure	Object . Begin Of	Control Begin Of	Common.Object.First_Child
Select last structure	Object . End Of	Control End Of	Common.Object.Last_Child
Turn off selection cursor is in	Item Off	Control X	Editor.Set.Designation_Off

Executing Commands

Description	Basic Keys	Accelerated Keys	Command
Create a Command window	Create Command		Common.Create_Command
Complete command name and parameters	Complete		Common.Complete
Execute a command	Promote	Control Return	Common.Promote
Execute command in background	Control Promote		Command.Spawn
Move to the next parameter prompt	Next Item	Meta N, Meta ↓	Editor.Cursor.Next
Move to the previous parameter prompt	Previous Item	Meta U, Meta ↑	Editor.Cursor.Previous
Turn a prompt into text	Item Off	Control X	Editor.Set.Designation_Off
Redisplay the previous command (undo)	Object . U		Common.Undo
Redisplay the next command (redo)	Object . R		Common.Redo
Provide prompts for the next key pressed	Prompt For	Meta Q	Editor.Key.Prompt

Managing Windows

Description	Basic Keys	Accelerated Keys	Command
Moving between Windows			
Move to the next window	Window . ↓	Meta V, Control Meta ↓	Editor.Window.Next
Move to the previous window	Window . ↑	Meta Z, Control Meta ↑	Editor.Window.Previous
Move to next attached window	Window . --		Editor.Window.Child
Move to previous attached window	Window . --		Editor.Window.Parent
Move up, ignoring window edge	Control Shift ↑		Editor.Screen.Up
Move down, ignoring window edge	Control Shift ↓		Editor.Screen.Down
Move right, ignoring window edge	Control Shift --		Editor.Screen.Right
Move left, ignoring window edge	Control Shift --		Editor.Screen.Left

Managing Windows (Continued)

Description	Basic Keys	Accelerated Keys	Command
Resizing and Repositioning Windows			
Join with the next window Join with the previous window Expand a window 4 lines Shrink a window 4 lines Transpose current window with previous Realign windows Copy a window	Window - J Window - Delete Window - I Window - . Window - T Window - Format Window - C		Editor.Window.Join (1) Editor.Window.Join (-1) Editor.Window.Expand Editor.Window.Expand (-4) Editor.Window.Transpose Editor.Window.Focus Editor.Window.Copy
Redrawing the Screen			
Redraw the screen Erase the screen, resetting to ANSI mode	Control L Meta L		Editor.Screen.Redraw Editor.Screen.Clear
Retaining Windows			
Lock a window on the screen Release a locked window	Window - Promote Window - Demote	Window - Edit	Editor.Window.Promote Editor.Window.Demote
Removing Windows			
Remove a window temporarily Release image, discarding changes Release image, saving changes Delete selected Window Directory entry	Window - D, Window - K, Window - X Object - G Object - X Object - D		Editor.Window.Delete Common.Abandon Common.Release Common.Object.Delete
Finding Windows			
Display Window Directory Display Window Directory entry	Window - Definition Definition	Window - ? Control Meta -	Editor.Window.Directory Common.Definition

Moving within an Image

Description	Basic Keys	Accelerated Keys	Command
By Character			
Move right 1 character Move right 8 characters Move right 1 character, wrapping lines Move left 1 character Move left 8 characters Move left 1 character, wrapping lines	-- Numeric 8 - -- -- Numeric 8 - --	Control J Control Meta J Control F Control H Control Meta H Control B	Editor.Cursor.Right Editor.Cursor.Right(8) Editor.Cursor.Forward Editor.Cursor.Left Editor.Cursor.Left(8) Editor.Cursor.Backward
By Word			
Move to next word Move to previous word Move to beginning of word Move to end of word	Word - -- Word - -- Word - Begin Of Word - End Of	Meta J, Meta -- Meta H, Meta -- Meta A, Meta B, Meta Begin Of Meta E, Meta End Of	Editor.Word.Next Editor.Word.Previous Editor.Word.Beginning-Of Editor.Word.End-Of
By Underline or Prompt			
Move to next underline or prompt Move to previous underline or prompt Move to next underline Move to previous underline Move to next prompt Move to previous prompt	Next Item Previous Item Next Underline Previous Underline Next Prompt Previous Prompt	Meta N, Meta ↓ Meta U, Meta ↑	Editor.Cursor.Next Editor.Cursor.Previous Editor.Cursor.Next Editor.Cursor.Previous Editor.Cursor.Next Editor.Cursor.Previous
By Line			
Move up 1 line Move up 8 lines Move down 1 line Move down 8 lines Move to beginning of line Move to end of line	↑ Numeric 8 - ↑ ↓ Numeric 8 - ↓ Begin Of End Of	Control U Control Meta U Control N Control Meta N Control A Control E	Editor.Cursor.Up Editor.Cursor.Up(8) Editor.Cursor.Down Editor.Cursor.Down(8) Editor.Line.Beginning-Of Editor.Line.End-Of

Moving within an Image (Continued)

Description	Basic Keys	Accelerated Keys	Command
In a Region			
Move to beginning of region Move to end of region	Region - Begin Of Region - End Of		Editor.Region.Beginning_Of Editor.Region.End_Of
By Tabs			
Tab forward Tab backward Display tab settings	Tab Control Tab Control Meta Tab		Editor.Char.Tab_Forward Editor.Char.Tab_Backward What.Tabs
By Scrolling			
Scroll up Scroll down Scroll right Scroll left Scroll to top of image Scroll to end of image Scroll current line to top Scroll current line to bottom	Image - ↑ Image - ↓ Image - → Image - ← Image - Begin Of Image - End Of Window - Begin Of Window - End Of	Control Z, Shift ↑ Control V, Shift ↓ Shift Begin Of Shift End Of Control Meta Begin Of Control Meta End Of	Editor.Image.Up Editor.Image.Down Editor.Image.Right Editor.Image.Left Editor.Image.Beginning_Of Editor.Image.End_Of Editor.Window.Beginning_Of Editor.Window.End_Of
By Marking Your Place			
Set mark at cursor position Cycle through marks in stack Cycle back through marks in stack Return to most recent mark Edit the mark stack Move top mark to bottom of stack Exchange top 2 marks on stack Copy top mark on stack Delete top mark from stack	Mark - ↓ Mark - → Mark - ← Mark - ↑ Mark - R Mark - T Mark - P Mark - Delete	Control M Meta M	Editor.Mark.Push Editor.Mark.Next Editor.Mark.Previous Editor.Mark.Top Editor.Mark.Rotate Editor.Mark.Swap Editor.Mark.Copy_Top Editor.Mark.Delete_Top

General Editing Operations

Description	Basic Keys	Accelerated Keys	Command
Selecting an Arbitrary Region			
Select start of region	Region - [Control [Editor.Region.Start
Select end of region	Region -]	Control]	Editor.Region.Finish
Unselect a region	Region - X		Editor.Region.Off
Moving and Copying Text			
Copy a selected item	Region - C		Editor.Region.Copy
Move a selected item	Region - M		Editor.Region.Move
Duplicate a single line	Line - C	Control Meta C	Editor.Line.Copy
Deleting Text			
Delete character — forward	Control D		Editor.Char.Delete_Forward
Delete character — backward	Delete		Editor.Char.Delete_Backward
Reduce multiple blanks to one	Control Delete		Editor.Char.Delete_Spaces
Delete word	Word - D	Meta D	Editor.Word.Delete
Delete to end of word	Word - K	Meta K	Editor.Word.Delete_Forward
Delete to beginning of word	Word - Delete	Meta Delete	Editor.Word.Delete_Backward
Delete line	Line - D	Control Meta D	Editor.Line.Delete
Delete to end of line	Line - K	Control K	Editor.Line.Delete_Forward
Delete to beginning of line	Line - Delete	Meta Delete	Editor.Line.Delete_Backward
Delete selected item	Region - D , Region - K		Editor.Region.Delete
Searching and Replacing Text			
Search for next occurrence	Control S		Editor.Search.Next
Search for previous occurrence	Control R		Editor.Search.Previous
Replace next occurrence	Meta S		Editor.Search.Replace_Next
Replace previous occurrence	Meta R		Editor.Search.Replace_Previous

General Editing Operations (Continued)

Description	Basic Keys	Accelerated Keys	Command
Entering Text			
<p>Quote a special character</p> <p>Split line, cursor on new line</p> <p>Open new line above current line</p> <p>Split line, cursor on old line</p> <p>Join 2 lines</p> <p>Enter text in insert mode</p> <p>Enter text in overwrite mode</p> <p>Show current line number</p>	<p>Control </p> <p>Line - </p> <p>Control </p> <p>Line - O</p> <p>Line - J</p> <p>Image - </p> <p>Image - O</p> <p>Line - ?</p>	<p>Control O</p> <p>Meta O,</p> <p>Control Meta O</p>	<p>Editor.Char.Quote</p> <p>Editor.Line.Insert</p> <p>Editor.Line.Insert(-1)</p> <p>Editor.Line.Open</p> <p>Editor.Line.Join</p> <p>Editor.Set.Insert_Mode(True)</p> <p>Editor.Set.Insert_Mode(False)</p> <p>What.Line</p>
Transposing Text			
<p>Transpose with previous character</p> <p>Transpose with previous word</p> <p>Transpose with previous line</p>	<p>Control T</p> <p>Word - T</p> <p>Line - T</p>	<p>Meta T</p> <p>Control Meta T</p>	<p>Editor.Char.Transpose</p> <p>Editor.Word.Transpose</p> <p>Editor.Line.Transpose</p>
Controlling Case			
<p>Capitalize character</p> <p>Capitalize to end of word</p> <p>Capitalize words to end of line</p> <p>Capitalize every word in region</p> <p>Convert character to lowercase</p> <p>Make lowercase to end of word</p> <p>Make lowercase to end of line</p> <p>Convert entire region to lowercase</p> <p>Convert character to uppercase</p> <p>Make uppercase to end of word</p> <p>Make uppercase to end of line</p> <p>Convert entire region to uppercase</p>	<p>Control ^</p> <p>Word - ^</p> <p>Line - ^</p> <p>Region - ^</p> <p>Control <</p> <p>Word - <</p> <p>Line - <</p> <p>Region - <</p> <p>Control ></p> <p>Word - ></p> <p>Line - ></p> <p>Region - ></p>	<p>Meta ^</p> <p>Control Meta ^</p> <p>Meta <</p> <p>Control Meta <</p> <p>Meta ></p> <p>Control Meta ></p>	<p>Editor.Char.Capitalize</p> <p>Editor.Word.Capitalize</p> <p>Editor.Line.Capitalize</p> <p>Editor.Region.Capitalize</p> <p>Editor.Char.Lower_Case</p> <p>Editor.Word.Lower_Case</p> <p>Editor.Line.Lower_Case</p> <p>Editor.Region.Lower_Case</p> <p>Editor.Char.Upper_Case</p> <p>Editor.Word.Upper_Case</p> <p>Editor.Line.Upper_Case</p> <p>Editor.Region.Upper_Case</p>

General Editing Operations (Continued)

Description	Basic Keys	Accelerated Keys	Command
Holding and Retrieving Text			
<p>Hold selected text</p> <p>Retrieve most recently held text</p> <p>Retrieve previous held text</p> <p>Retrieve next held text</p> <p>Edit the hold stack</p> <p style="padding-left: 20px;">Move top element to bottom of stack</p> <p style="padding-left: 20px;">Exchange top 2 elements on stack</p> <p style="padding-left: 20px;">Copy top element on stack</p> <p style="padding-left: 20px;">Delete top element from stack</p>	<p>Region - I</p> <p>Region - ↑</p> <p>Region - ←</p> <p>Region - →</p> <p>Region - R</p> <p>Region - T</p> <p>Region - P</p> <p>Region - Delete</p>	<p>Control C</p> <p>Control Y</p> <p>Meta C, Meta Y</p>	<p>Editor.Hold_Stack.Push</p> <p>Editor.Hold_Stack.Top</p> <p>Editor.Hold_Stack.Previous</p> <p>Editor.Hold_Stack.Next</p> <p>Editor.Hold_Stack.Rotate</p> <p>Editor.Hold_Stack.Swap</p> <p>Editor.Hold_Stack.Copy_Top</p> <p>Editor.Hold_Stack.Delete_Top</p>
Formatting Text			
<p>Center the line cursor is on</p> <p>Fill text in selected region</p> <p>Justify text in selected region</p> <p>Automatically wrap lines</p> <p>Do not wrap lines</p>	<p>Line - §</p> <p>Region - Format</p> <p>Region - Complete</p> <p>Image - F</p> <p>Image - X</p>		<p>Editor.Line.Center</p> <p>Editor.Region.Fill</p> <p>Editor.Region.Justify</p> <p>Editor.Set.Fill_Mode(True)</p> <p>Editor.Set.Fill_Mode(False)</p>

Writing Text Files

Description	Basic Keys	Accelerated Keys	Command
Accessing Text Files			
Create a new text file Display existing text file Open text file for editing Revert to last saved version	Create Text Definition Edit Object - L	Control Meta -	Text.Create Common.Definition Common.Edit Common.Revert
Saving Changes			
Save, leaving open for editing Save, making read only	Enter Promote	Control Return	Common.Commit Common.Promote
Terminating Edit			
Remove image, discarding changes Remove image, saving changes	Object - G Object - X		Common.Abandon Common.Release
Selecting Substructures within Text			
Select current word Select current sentence Select current paragraph Select smaller structure Select previous structure, same level Select next structure, same level Turn off selection	Object - - numeric 2 - Object - - numeric 3 - Object - - Object - - Object - ↑ Object - ↓ Item Off	Control - Control - Control - Control - Control ↑ Control ↓ Control X	Common.Object.Parent Common.Object.Parent Common.Object.Parent Common.Object.Child Common.Object.Previous Common.Object.Next Editor.Set.Designation..Off

Writing Ada Programs

Description	Basic Keys	Accelerated Keys	Command
Creating Ada Programs			
Create an Ada unit in library Build a body Build a private part Put temporary name in library	Create Ada Create Body Create Private Install Stub		Common.Object.Insert Ada.Create-Body Ada.Create-Private Ada.Install-Stub
Accessing Ada Programs			
Display Ada unit, read only Demote to source, open for editing	Definition Edit	Control Meta --	Common.Definition Common.Edit
Saving Changes and Terminating Edit			
Save, leaving open for editing Release image, discarding changes Release image, saving changes Revert to last version	Enter Object . G Object . X Object . L	Control Return	Common.Commit Common.Abandon Common.Release Common.Revert
Checking for Errors			
Complete and check syntax Check for semantic errors Explain underlined error Move to next underlined error Move to previous underlined error Remove underline from error Clear all underlined errors Redisplay cleared errors	Format Semanticize Explain Next Underline Next Item Previous Underline Previous Item Item Off Underlines Off Show Errors	Control ? Meta N, Meta I Meta U, Meta ? Control X	Common.Format Common.Semanticize Common.Explain Editor.Cursor.Next Editor.Cursor.Next Editor.Cursor.Previous Editor.Cursor.Previous Editor.Set.Designation-Off Common.Clear-Underlining Ada.Get-Errors

Writing Ada Programs (Continued)

Description	Basic Keys	Accelerated Keys	Command
Changing the Compilation State			
Change unit to source state from any state Change unit to installed state from any state Change unit to coded state from any state	<input type="text" value="Source Unit"/> <input type="text" value="Install Unit"/> <input type="text" value="Code Unit"/>		Ada.Source_Unit Ada.Install_Unit Ada.Code_Unit
Changing to a Higher Compilation State			
Promote unit to next higher state Code unit and those it depends on In this world only Across worlds Install unit and those it depends on In this world only Across worlds	<input type="text" value="Promote"/> <input type="text" value="Code (This World)"/> <input type="text" value="Code (All Worlds)"/> <input type="text" value="Install (This World)"/> <input type="text" value="Install (All Worlds)"/>		Common.Promote Compilation.Make Compilation.Make Compilation.Promote Compilation.Promote
Changing to a Lower Compilation State			
Demote unit to next lower state Demote unit and dependents to source In this world only Across worlds Demote unit and dependents to installed In this world only Across worlds	<input type="text" value="Demote"/> <input type="text" value="Source (This World)"/> <input type="text" value="Source (All Worlds)"/> <input type="text" value="Uncode (This World)"/> <input type="text" value="Uncode (All Worlds)"/>		Common.Demote Compilation.Demote Compilation.Demote Compilation.Demote Compilation.Demote
Selecting Structures within Ada Programs			
Select successively larger structures Select successively smaller structures Select previous structure, same level Select next structure, same level Select first structure Select last structure Turn off selection cursor is in	<input type="text" value="Object"/> - <input type="text" value="←"/> <input type="text" value="Object"/> - <input type="text" value="→"/> <input type="text" value="Object"/> - <input type="text" value="↑"/> <input type="text" value="Object"/> - <input type="text" value="↓"/> <input type="text" value="Object"/> - <input type="text" value="Begin Of"/> <input type="text" value="Object"/> - <input type="text" value="End Of"/> <input type="text" value="Item Off"/>	<input type="text" value="Control"/> <input type="text" value="←"/> <input type="text" value="Control"/> <input type="text" value="→"/> <input type="text" value="Control"/> <input type="text" value="↑"/> <input type="text" value="Control"/> <input type="text" value="↓"/> <input type="text" value="Control"/> <input type="text" value="Begin Of"/> <input type="text" value="Control"/> <input type="text" value="End Of"/> <input type="text" value="Control"/> <input type="text" value="X"/>	Common.Object.Parent Common.Object.Child Common.Object.Previous Common.Object.Next Common.Object.First_Child Common.Object.Last_Child Editor.Set.Designation_Off

Writing Ada Programs (Continued)

Description	Basic Keys	Accelerated Keys	Command
Modifying Ada Programs			
Edit selected Ada structure Insert Ada structures(s) in program Delete selected Ada structure Copy selected Ada structure Move selected Ada structure Withdraw Ada unit stub	Edit Object - I Object - D, Object - K Object - C Object - M Withdraw Unit		Common.Edit Common.Object.Insert Common.Object.Delete Common.Object.Copy Common.Object.Move Ada.Withdraw
Entering Comments and Special Strings			
Comment selected item or region Uncomment selected item or region Tab forward to comment Insert string (" Insert string ") Insert string := Insert string =>	Region - - Region - = Object - Tab Control { Control } Control : Control +	Meta Tab	Region.Comment Region.Uncomment Editor.Char.Tab.To.Comment Editor.Char.Insert.String Editor.Char.Insert.String Editor.Char.Insert.String Editor.Char.Insert.String
Browsing Ada Programs			
Display other part of Ada unit Display other part, same window Display Ada unit cursor is on Display parent object Set mark at current location Cycle through marks in stack Cycle back through marks in stack Return to most recent mark	Other Part Other Part In Place Definition Enclosing Mark - I Mark - - Mark - - Mark - ↑	Control Meta - Control Meta - Control M Meta M	Ada.Other-Part Ada.Other-Part Common.Definition Common.Enclosing Editor.Mark.Push Editor.Mark.Next Editor.Mark.Previous Editor.Mark.Top
Checking Using Occurrences			
Show uses of selected identifier In this unit only In any unit Including indirect references Show unused declarations In this unit only Check other units	Show Usage (Unit) Show Usage Show Usage (Indirect) Show Unused (Unit) Show Unused		Ada.Show-Usage Ada.Show-Usage Ada.Show-Usage Ada.Show-Unused Ada.Show-Unused

Debugging Ada Programs

Description	Basic Keys	Accelerated Keys	Command
Execute program with Debugger on Display Debugger window Show current statement in source	Meta Promote Debugger Window Show Source	Meta Return	Command.Debug Debug.Current_Debugger Debug.Source
Stepping and Executing			
Continue program execution Step one statement Step one statement at same level Run until subprogram has returned Stop task execution Display information about tasks Display task rendezvous info	Execute Run Run Local Run Returned Stop Task Display Rendezvous Info		Debug.Execute Debug.Run Debug.Run (Local) Debug.Run(Returned) Debug.Stop Debug.Task_Display Debug.Information
Setting and Removing Breakpoints			
Set breakpoints with default lifetime Set breakpoints with nondefault lifetime Display breakpoints Reactivate existing breakpoints Remove breakpoints	Break Break ~Default Show Breaks Activate Remove Breaks		Debug.Break Debug.Ereak(False) Debug.Show Debug.Activate Debug.Remove
Viewing Stacks			
Display calling stack	Stack		Debug.Stack
Displaying and Modifying Variables			
Display values of selected variables Modify value of selected variable Specify how much Put displays Request <i>n</i> pointer levels Request <i>n</i> array elements Start with <i>m</i> th array element	Put Modify numeric <i>n</i> - - Set Pointer Level - Set Element Count - Set First Element		Debug.Put Debug.Modify Debug.Set_Value Debug.Set_Value Debug.Set_Value
Handling Exceptions			
Stop execution when exception raised Do not stop when exception raised Summarize how exceptions are handled Remove handling for this exception	Catch Propagate Show Exceptions Forget		Debug.Catch Debug.Propagate Debug.Show(Exceptions) Debug.Forget

Managing Libraries

Description	Basic Keys	Accelerated Keys	Command
Creating Libraries			
Create a directory Create a world	Create Directory Create World		Library.Create_Directory Library.Create_World
Manipulating Objects in Libraries			
Create an Ada unit in library Create a text file in library Delete selected object from library Undelete selected object from library Print selected object Show access list for designated object	Create Ada Create Text Object - D, Object - K Object - U Print Show Access List		Common.Object.Insert Text.Create Common.Object.Delete Common.Object.Undelete Queue.Print Access-List.Display
Controlling Library Display			
Toggle information on library objects Show more detail Show less detail	Explain Object - I Object - O	Control ?, Object - ?	Common.Explain Common.Expand Common.Elide

Using CMVC

Description	Basic Keys	Accelerated Keys	Command
Check out designated object Check in designated object Accept changes for designated object Show objects that are checked out In this view By you, any view Show info about designated object Show out-of-date objects in this view	Check Out Check In Accept Changes Checked Out In View Checked Out By User Show Info Show Out Of Date		Cmvc.Check_Out Cmvc.Check_In Cmvc.Accept_Changes Cmvc.Show_Checked_Out_In_View Cmvc.Show_Checked_Out_By_User Cmvc.Show Cmvc.Show_Out_Of_Date_Objects

Managing Links

Description	Basic Keys	Accelerated Keys	Command
Accessing Links			
List links Edit links display Refresh link image	- - Object . L		Links.Display Links.Edit Common.Revert
Removing the Link Editor			
Remove window temporarily Release image permanently	Window . D Object . X		Editor.Window.Delete Common.Release
Selecting Links			
Select link cursor is on Select all links Select previous link Select next link Select first link in image Select last link in image	Object . -- Object . -- Object . Object . Object . Begin Of Object . End Of	Control -- Control -- Control Control Control Begin Of Control End Of	Common.Object.Child Common.Object.Parent Common.Object.Previous Common.Object.Next Common.Object.First_Child Common.Object.Last_Child
Modifying Links			
Add a new link—simple method Add a new link Give selected link another source Delete selected link	- Object . I Edit Object . D, Object . K		Links.Add Common.Object.Insert Common.Edit Common.Object.Delete
Traversing Linked Ada Units			
Go to source unit of current link Go to world associated with links List Ada units that use current link	Definition Enclosing Explain	Control Meta -- Control Meta -- Control ? , Object . ?	Common.Definition Common.Enclosing Common.Explain
Controlling the Display			
Toggle order of kind of link Toggle classes of source of link	Object . Object .	Control	Common.Expand Common.Elide

Managing Searchlists

Description	Basic Keys	Accelerated Keys	Command
Accessing the Searchlist			
Edit or view searchlist Refresh searchlist image	- Object - L		Search.List.Edit Common.Revert
Removing the Searchlist Editor			
Remove window temporarily Release image permanently	Window - D Object - X		Editor.Window.Delete Common.Release
Selecting Entries			
Select entry cursor is on Select all entries Select next entry Select previous entry Select first entry on list Select last entry on list Go to world named by current entry	Object - ← Object - → Object - ↓ Object - ↑ Object - Begin Of Object - End Of Definition	Control - ← Control - → Control - ↓ Control - ↑ Control - Begin Of Control - End Of Control - Meta - ←	Common.Object.Child Common.Object.Parent Common.Object.Next Common.Object.Previous Common.Object.First_Child Common.Object.Last_Child Common.Definition
Modifying the Searchlist			
Add a new entry Delete selected entry Move selected entry	Object - I Object - D Object - K Object - M		Common.Object.Insert Common.Object.Delete Common.Object.Move

Using Keyboard Macros

Description	Basic Keys	Accelerated Keys	Command
Start macro definition	Mark . Begin Of	Meta	Editor.Macro.Start
End macro definition	Mark . End Of	Meta]	Editor.Macro.Finish
Execute macro	Mark . Promote	Meta X	Editor.Macro.Execute
Bind macro to key	Mark . Definition		Editor.Macro.Bind

Remembering and Redisplaying Screens

Description	Basic Keys	Accelerated Keys	Command
Remember this screen	Control Meta Shift ↓		Editor.Screen.Push
Redisplay most recently remembered screen	Control Meta Shift ↑		Editor.Screen.Top
Cycle through screens in stack	Control Meta Shift --		Editor.Screen.Previous
Cycle back through screens in stack	Control Meta Shift ==		Editor.Screen.Next

Using Environment I/O Resources

Description	Basic Keys	Accelerated Keys	Command
Indicate end of input to program	End Of Input	numeric .	Text.End.Of.Input
Commit interactive input	Promote	Control Return, Enter	Common.Commit

Managing Jobs

Description	Basic Keys	Accelerated Keys	Command
Disconnect job from terminal	Control G		Job.Interrupt
Kill job	Job Kill	Meta G	Job.Kill(0)
Stop running jobs	Job Disable		Job.Disable(0)
Resume stopped jobs	Job Enable		Job.Enable(0)
Reconnect job	Job Connect		Job.Connect(0)

Master Reference to Key Bindings by Command

Legend	Command		Key Bindings	
====	C = Control	[] = Meta	CN = Control	MS = Meta
	CH = Control	[] = Meta	MS = Meta	MS = Meta
	S = Shift	[] = Meta	MS = Meta	MS = Meta
====				
Command	====	Rational		
Access_List.Display (For_Object...)	CMS_F20			
Ada.Code_Unit	S_F13			
Ada.Create_Body	CMS_F15			
Ada.Create_Private	CH_K			
Ada.Delete_Blank_Line	CH_K			
Ada.Get_Errors	S_F16			
Ada.Insert_Blank_Line	CMS_I			
Ada.Install_Stub	CH_L			
Ada.Install_Unit	F13			
Ada.Make_Inline	CMS_F17			
Ada.Make_Separate	CMS_F18			
Ada.Other_Part (Name => "<Image>...)	C_F10			
Ada.Other_Part (Name => "<Image>...)	CS_F10			
Ada.Show_Deused (In_Unit => "I...)	M_F17			
Ada.Show_Deused (In_Unit => "I...)	CH_F17			
Ada.Show_Usage (Name => "<Cursor>...)	M_F16			
Ada.Show_Usage (Name => "<Cursor>...)	CH_F16			
Ada.Show_Usage (Name => "<Cursor>...)	CMS_F16			
Ada.Source_Unit	C_F14			
Ada.Withdraw	M_F14			
Conv.Accept_Changes (Destination...)	CH_F12			
Conv.Check_In (What_Object => " ...)	M_F12			
Conv.Check_Out (What_Object => " ...)	C_F12			
Conv.Show (Objects => "<Cursor>...)	CS_F12			
Conv.Show_Checked_Out_By_User (...)	MS_F12			
Conv.Show_Checked_Out_In_View (...)	CMS_F12			
Conv.Show_Out_Of_Data_Objects (...)	S_F12			
Command.Debug	M_PROMOT			
MS_CARRIAGE_RETURN	MS_CARRIAGE_RETURN			
M_CARRIAGE_RETURN	M_CARRIAGE_RETURN			
MS_ENTER	MS_ENTER			
M_ENTER	M_ENTER			
C_PROMOT	C_PROMOT			
OBJECT.'G'	OBJECT.'G'			
OBJECT.'g'	OBJECT.'g'			
S_ENTER	S_ENTER			
C_CARRIAGE_RETURN	C_CARRIAGE_RETURN			
CS_CARRIAGE_RETURN	CS_CARRIAGE_RETURN			
COMPLT	COMPLT			
Command.Spawn				
Common.Abandon				
Common.Clear_Underline				
Common.Commit				
Common.Complete				
Command.Create_Command	F15			
Common.Definition (In_Placement => " ...)	CH_RIGHT			
Common.Definition (In_Placement => " ...)	S_RIGHT			
Common.Definition (In_Placement => " ...)	F10			
Common.Definition (In_Placement => " ...)	S_F10			
Common.Definition (In_Placement => " ...)	S_F14			
Common.Definition (In_Placement => " ...)	F14			
Common.Definition (In_Placement => " ...)	OBJECT.'>			
Common.Definition (In_Placement => " ...)	M_F10			
Common.Definition (In_Placement => " ...)	CH_LEFT			
Common.Definition (In_Placement => " ...)	S_LEFT			
Common.Definition (In_Placement => " ...)	MS_F10			
Common.Definition (In_Placement => " ...)	CMS_F10			
Common.Definition (In_Placement => " ...)	C_EXCLAM			
Common.Definition (In_Placement => " ...)	OBJECT.'!'			
Common.Definition (In_Placement => " ...)	OBJECT.'!'			
Common.Definition (In_Placement => " ...)	OBJECT.'?'			
Common.Definition (In_Placement => " ...)	F17			
Common.Definition (In_Placement => " ...)	C_QUERY			
Common.Definition (In_Placement => " ...)	OBJECT.'/'			
Common.Definition (In_Placement => " ...)	C_SLASH			
Common.Definition (In_Placement => " ...)	FORMAT			
Common.Definition (In_Placement => " ...)	OBJECT.RIGHT			
Common.Definition (In_Placement => " ...)	C_RIGHT			
Common.Definition (In_Placement => " ...)	OBJECT.'c'			
Common.Definition (In_Placement => " ...)	OBJECT.'d'			
Common.Definition (In_Placement => " ...)	OBJECT.'D'			
Common.Definition (In_Placement => " ...)	OBJECT.'k'			
Common.Definition (In_Placement => " ...)	OBJECT.'K'			
Common.Definition (In_Placement => " ...)	OBJECT.BEGIN_OF			
Common.Definition (In_Placement => " ...)	C_BEGIN_OF			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	C_F15			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	C_END_OF			
Common.Definition (In_Placement => " ...)	OBJECT.END_OF			
Common.Definition (In_Placement => " ...)	OBJECT.'M'			
Common.Definition (In_Placement => " ...)	OBJECT.'m'			
Common.Definition (In_Placement => " ...)	C_DOWN			
Common.Definition (In_Placement => " ...)	OBJECT.DOWN			
Common.Definition (In_Placement => " ...)	OBJECT.LEFT			
Common.Definition (In_Placement => " ...)	C_LEFT			
Common.Definition (In_Placement => " ...)	C_UP			
Common.Definition (In_Placement => " ...)	PROMOT			
Common.Definition (In_Placement => " ...)	OBJECT.'r'			
Common.Definition (In_Placement => " ...)	OBJECT.'R'			
Common.Definition (In_Placement => " ...)	OBJECT.'x'			
Common.Definition (In_Placement => " ...)	OBJECT.'X'			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	OBJECT.'L'			
Common.Definition (In_Placement => " ...)	F16			
Common.Definition (In_Placement => " ...)	OBJECT.'s'			
Common.Definition (In_Placement => " ...)	OBJECT.'u'			
Common.Definition (In_Placement => " ...)	OBJECT.'U'			
Command.Definition (In_Placement => " ...)	OBJECT.RIGHT			
Common.Definition (In_Placement => " ...)	OBJECT.'c'			
Common.Definition (In_Placement => " ...)	OBJECT.'d'			
Common.Definition (In_Placement => " ...)	OBJECT.'D'			
Common.Definition (In_Placement => " ...)	OBJECT.'k'			
Common.Definition (In_Placement => " ...)	OBJECT.'K'			
Common.Definition (In_Placement => " ...)	OBJECT.BEGIN_OF			
Common.Definition (In_Placement => " ...)	C_BEGIN_OF			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	C_F15			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	C_END_OF			
Common.Definition (In_Placement => " ...)	OBJECT.END_OF			
Common.Definition (In_Placement => " ...)	OBJECT.'M'			
Common.Definition (In_Placement => " ...)	OBJECT.'m'			
Common.Definition (In_Placement => " ...)	C_DOWN			
Common.Definition (In_Placement => " ...)	OBJECT.DOWN			
Common.Definition (In_Placement => " ...)	OBJECT.LEFT			
Common.Definition (In_Placement => " ...)	C_LEFT			
Common.Definition (In_Placement => " ...)	C_UP			
Common.Definition (In_Placement => " ...)	PROMOT			
Common.Definition (In_Placement => " ...)	OBJECT.'r'			
Common.Definition (In_Placement => " ...)	OBJECT.'R'			
Common.Definition (In_Placement => " ...)	OBJECT.'x'			
Common.Definition (In_Placement => " ...)	OBJECT.'X'			
Common.Definition (In_Placement => " ...)	OBJECT.'l'			
Common.Definition (In_Placement => " ...)	OBJECT.'L'			
Common.Definition (In_Placement => " ...)	F16			
Common.Definition (In_Placement => " ...)	OBJECT.'s'			
Common.Definition (In_Placement => " ...)	OBJECT.'u'			
Common.Definition (In_Placement => " ...)	OBJECT.'U'			

NUMERIC_0..'U'

OBJECT..'U'

```

Compilation.Demote ( Unit => "<S..." MS_F14
Compilation.Demote ( Unit => "<S..." CS_F14
Compilation.Demote ( Unit => "<S..." MS_F14
Compilation.Demote ( Unit => "<S..." CS_F14
Compilation.Make ( Unit => "<Ima..." CS_F13
Compilation.Make ( Unit => "<Ima..." MS_F13
Compilation.Promote ( Unit => "<..." MS_F13
Compilation.Promote ( Unit => "<..." MS_F13
Debug.Activate ( Breakpoint => 0) M_F7
Debug.Break ( Default_Lifetime =>... S_F7
Debug.Catch C_F8
Debug.Current_Debgger ( "" ) MS_F9
Debug.Execute S_F6
Debug.Forget S_F8
Debug.Information ( Debug_Refresh... MS_F8
Debug.Modify ( New_Value => "" )... MS_F9
Debug.Propagate M_F9
Debug.Put E9
Debug.Remove ( Breakpoint => 0) MS_F7
Debug.Run F6
Debug.Run ( Debug_Returned ) M_F6
Debug.Run ( Stop_At => Debug_Loc... ) C_F6
Debug.Set_Value ( Variable => De... ) C_F9
Debug.Set_Value ( Variable => De... ) S_F9
Debug.Set_Value ( Variable => De... ) M_F9
Debug.Show MS_F7
Debug.Show ( Debug_Exceptions ) MS_F8
Debug.Source ( Location => "" , S... ) F7
Debug.Stack F8
Debug.Stop ( Name => "" ) MS_F6
Debug.Task_Display CS_F6
Editor.Char.Capitalize C_6
Editor.Char.Delete_Backward DELETE
Editor.Char.Delete_Backward C_DELETE
Editor.Char.Delete_Backward CS_DELETE
Editor.Char.Delete_Spaces CS_DELETE
Editor.Char.Insert_Character ( I... ) MS_SPACE
Editor.Char.Insert_Character ( I... ) CS_SPACE
Editor.Char.Insert_Character ( I... ) MS_SPACE
Editor.Char.Insert_Character ( I... ) CS_SPACE
Editor.Char.Insert_String ( "" ) CS_SPACE
Editor.Char.Insert_String ( "" ) S_SPACE
Editor.Char.Insert_String ( "" ) C_RIGHT_PAREN
Editor.Char.Insert_String ( "" ) C_9
Editor.Char.Insert_String ( "" ) C_LEFT_PAREN
Editor.Char.Insert_String ( "" ) C_COLON
Editor.Char.Insert_String ( "" ) C_SEMICOLON
Editor.Char.Insert_String ( "" ) C_EQUAL
Editor.Char.Insert_String ( "" ) C_LESS_THAN
Editor.Char.Lower_Case C_COMMA
Editor.Char.Quote C_QUOTE
Editor.Char.Quote CS_TICK

```

```

Editor.Char.Tab_Backward CS_TAB
Editor.Char.Tab_Forward C_TAB
Editor.Char.Tab_Forward S_TAB
Editor.Char.Tab_Forward MS_TAB
Editor.Char.Tab_Forward CS_TAB
Editor.Char.Tab_Forward OBJECT_TAB
Editor.Char.Tab_Forward M_TAB
Editor.Char.Transpose C_J
Editor.Char.Upper_Case C_PERIOD
Editor.Cursor.Backward C_DELETE
Editor.Cursor.Backward CS_DELETE
Editor.Cursor.Down CS_DELETE
Editor.Cursor.Down MS_DELETE
Editor.Cursor.Down ESC_DELETE
Editor.Cursor.Forward C_DELETE
Editor.Cursor.Forward CS_DELETE
Editor.Cursor.Left C_DELETE
Editor.Cursor.Left CS_DELETE
Editor.Cursor.Next ( Prompt => F... ) S_F10
Editor.Cursor.Next ( Prompt => F... ) M_F10
Editor.Cursor.Next ( Prompt => I... ) F10
Editor.Cursor.Next ( Prompt => I... ) M_F10
Editor.Cursor.Previous MS_DELETE
Editor.Cursor.Previous M_DELETE
Editor.Cursor.Previous ( Prompt... ) CS_F10
Editor.Cursor.Previous ( Prompt... ) M_F10
Editor.Cursor.Previous ( Prompt... ) C_F10
Editor.Cursor.Right CS_DELETE
Editor.Cursor.Right CS_DELETE
Editor.Cursor.Up CS_DELETE
Editor.Cursor.Up ( 7 ) CS_DELETE
Editor.Hold_Stack.Copy_Top REGION_P
Editor.Hold_Stack.Delete_Top REGION_P
Editor.Hold_Stack.Next M_DELETE
Editor.Hold_Stack.Previous MS_DELETE

```

Editor.Hold_Stack.Push	C.C	REGION.DOWN
Editor.Hold_Stack.Rotate	PF1.DOWN	CS.C
Editor.Hold_Stack.Swap		REGION.'r'
Editor.Hold_Stack.Top		REGION.'R'
Editor.Line.Beginning_0f	PF1.UP	REGION.'t'
Editor.Image.Down	C.Y	REGION.'c'
Editor.Image.End_Of		REGION.UP
Editor.Image.Find (")	PF3.NUMERIC_7	CS.Y
Editor.Image.Find ("Help Window")	C.V	CS.X
Editor.Image.Left	PF3.DOWN	CS.Y
Editor.Image.Right	PF3.DOWN	CS.Y
Editor.Image.Up	PF3.DOWN	CS.Y
Editor.Key.Name	C.Q	CS.Q
Editor.Key.Prompt	ESC.Q	M.A
Editor.Key.Prompt (*Key_Code => *)	ESC.Q	M.L
Editor.Line.Beginning_Delete	ESC.C.B	M.O
Editor.Line.Capitalize	C.B	CS.B
Editor.Line.Center	NUMERIC_7	CS.B
Editor.Line.Copy	ESC.C.A	CS.A
Editor.Line.Delete	C.A	C.A
Editor.Line.Delete (Backward)		C.A
Editor.Line.Delete (Forward)		CS.A
Editor.Line.Delete (Next Line)		LINE.BEGIN_OF
Editor.Line.Delete (Previous Line)		CS.A
Editor.Line.Delete (Whole Line)		CS.A
Editor.Line.Delete (Whole Paragraph)		REGION_OF
Editor.Line.Delete (Whole Section)		CS.A
Editor.Line.Delete (Whole Text)		CS.A
Editor.Line.Delete (Whole Window)		CS.A
Editor.Line.Delete (Whole Document)		CS.A
Editor.Line.Delete (Whole File)		CS.A
Editor.Line.Delete (Whole System)		CS.A

Editor.Line.Delete_Forward	CS.K	ESC.C.K
Editor.Line.End_Of	LINE.'k'	C.K
Editor.Line.Indent ()	CM.E	NUMERIC_9
Editor.Line.Insert (- (1))	CS.E	C.E
Editor.Line.Join	END_OF	ESC.C.E
Editor.Line.Open	LINE.END_OF	
Editor.Line.Transpose	CS.L	C.L
Editor.Line.Upper_Case	CS.L	
Editor.Macro.Bind	MARK.ENTER	PF4.NUMERIC_9
Editor.Macro.Execute	MARK.ENTER	PF4.ENTER
Editor.Macro.Finish	MARK.ENTER	PF4.C.M
Editor.Mark.Copy_Top	MARK.ENTER	PF4.NUMERIC_7
Editor.Mark.Delete_Top	MARK.DELETE	
Editor.Mark.Delete_Top (Backward)	MARK.DELETE	
Editor.Mark.Delete_Top (Forward)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Line)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Paragraph)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Section)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Text)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Window)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole Document)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole File)	MARK.DELETE	
Editor.Mark.Delete_Top (Whole System)	MARK.DELETE	

Editor.Mark.Previous	MARK.LEFT	PF4.LEFT
Editor.Mark.Push	MARK.DOWN	MUL
Editor.Mark.Toggle	CS_M	PF4.DOWN
Editor.Mark.Swap	MARK.'R'	
Editor.Mark.Top	MARK.LEFT	
Editor.Region.Beginning_Of	MARK.'T'	PF4.UP
Editor.Region.Capitalize	REGION.BEGIN_OF	PEL.NUMERIC_7
Editor.Region.Comment	REGION.'6'	PEL.'6'
Editor.Region.Copy	REGION.'5'	PEL.'5'
Editor.Region.Delete	REGION.'4'	PEL.'4'
Editor.Region.End_Of	REGION.'3'	PEL.'3'
Editor.Region.Fill	REGION.'2'	PEL.'2'
Editor.Region.Finish	REGION.'1'	PEL.'1'
Editor.Region.Format	REGION.END_OF	PEL.NUMERIC_9
Editor.Region.Toggle	REGION.FORMAT	'ESC.RIGHT_BRACKET
Editor.Region.Toggle	REGION.'0'	PEL.'0'
Editor.Region.Toggle	C.RIGHT_BRACKET	
Editor.Region.Toggle	C.LEFT_BRACKET	
Editor.Region.Toggle	REGION.'	
Editor.Region.Toggle	REGION.COMPLI	
Editor.Region.Toggle	REGION.'<'	
Editor.Region.Toggle	REGION.'	
Editor.Region.Toggle	REGION.'M'	
Editor.Region.Toggle	REGION.'m'	
Editor.Region.Toggle	REGION.'x'	
Editor.Region.Toggle	REGION.'X'	
Editor.Region.Toggle	C.LEFT_BRACKET	
Editor.Region.Toggle	REGION.'	
Editor.Region.Toggle	C.LEFT_BRACKET	
Editor.Region.Toggle	REGION.'	
Editor.Region.Toggle	REGION.'>'	
Editor.Region.Toggle	REGION.'	
Editor.Region.Toggle	REGION.'S'	
Editor.Region.Toggle	REGION.'s'	
Editor.Region.Toggle	MS_DOWN	
Editor.Region.Toggle	MS_UP	
Editor.Region.Toggle	CS_DOWN	
Editor.Region.Toggle	CS_LEFT	
Editor.Region.Toggle	MS_RIGHT	
Editor.Region.Toggle	MS_LEFT	
Editor.Region.Toggle	CS_DOWN	
Editor.Region.Toggle	CS_UP	
Editor.Region.Toggle	CS_RIGHT	
Editor.Region.Toggle	MS_UP	
Editor.Region.Toggle	CS_UP	
Editor.Region.Toggle	C.S	
Editor.Region.Toggle	CS.F	
Editor.Region.Toggle	CS.M	
Editor.Region.Toggle	C.R	
Editor.Region.Toggle	M.S	
Editor.Region.Toggle	MS.S	

Editor.Search.Replace.Previous	M.R	ESC.R
Editor.Set.Argument.Digit (0)	MS.R	
Editor.Set.Argument.Digit (1)	M.NUMERIC_0	
Editor.Set.Argument.Digit (2)	C.NUMERIC_0	
Editor.Set.Argument.Digit (3)	S.NUMERIC_0	
Editor.Set.Argument.Digit (4)	M.NUMERIC_1	
Editor.Set.Argument.Digit (5)	C.NUMERIC_1	
Editor.Set.Argument.Digit (6)	S.NUMERIC_1	
Editor.Set.Argument.Digit (7)	M.NUMERIC_2	
Editor.Set.Argument.Digit (8)	C.NUMERIC_2	
Editor.Set.Argument.Digit (9)	S.NUMERIC_2	
Editor.Set.Argument.Minus	M.NUMERIC_3	
Editor.Set.Designation.Off	C.NUMERIC_3	
Editor.Set.Fill.Mode (False)	NUMERIC_4	
Editor.Set.Fill.Mode (True)	C.NUMERIC_4	
Editor.Set.Insert.Mode (False)	S.NUMERIC_4	
Editor.Set.Insert.Mode (True)	M.NUMERIC_5	
Editor.Set.Insert.Mode (True)	C.NUMERIC_5	
Editor.Set.Insert.Mode (True)	S.NUMERIC_5	
Editor.Set.Insert.Mode (True)	M.NUMERIC_6	
Editor.Set.Insert.Mode (True)	C.NUMERIC_6	
Editor.Set.Insert.Mode (True)	S.NUMERIC_6	
Editor.Set.Insert.Mode (True)	M.NUMERIC_7	
Editor.Set.Insert.Mode (True)	C.NUMERIC_7	
Editor.Set.Insert.Mode (True)	S.NUMERIC_7	
Editor.Set.Insert.Mode (True)	M.NUMERIC_8	
Editor.Set.Insert.Mode (True)	C.NUMERIC_8	
Editor.Set.Insert.Mode (True)	S.NUMERIC_8	
Editor.Set.Insert.Mode (True)	M.NUMERIC_9	
Editor.Set.Insert.Mode (True)	C.NUMERIC_9	
Editor.Set.Insert.Mode (True)	S.NUMERIC_9	
Editor.Set.Insert.Mode (True)	C.DASH	
Editor.Set.Insert.Mode (True)	DASH	
Editor.Set.Insert.Mode (True)	S.DASH	
Editor.Set.Insert.Mode (True)	M.DASH	
Editor.Set.Insert.Mode (True)	NUMERIC_CONNA	
Editor.Set.Insert.Mode (True)	C.NUMERIC_CONNA	
Editor.Set.Insert.Mode (True)	S.NUMERIC_CONNA	
Editor.Set.Insert.Mode (True)	C.SX	
Editor.Set.Insert.Mode (True)	C.FY	
Editor.Set.Insert.Mode (True)	C.J	
Editor.Set.Insert.Mode (True)	C.X	
Editor.Set.Insert.Mode (True)	IMAGE.'X'	PF3.'X'
Editor.Set.Insert.Mode (True)	IMAGE.'x'	PF3.'x'
Editor.Set.Insert.Mode (True)	IMAGE.'f'	PF3.'f'
Editor.Set.Insert.Mode (True)	IMAGE.'F'	PF3.'F'
Editor.Set.Insert.Mode (True)	IMAGE.'o'	PF3.'o'
Editor.Set.Insert.Mode (True)	IMAGE.'O'	PF3.'O'
Editor.Set.Insert.Mode (True)	IMAGE.'i'	PF3.'i'
Editor.Set.Insert.Mode (True)	IMAGE.'I'	PF3.'I'

14 C

20 11 1978

also see the ... of ...
... of ...
... of ...

... of ...
... of ...
... of ...

... of ...
... of ...
... of ...

... of ...
... of ...
... of ...

... of ...
... of ...
... of ...

... of ...
... of ...
... of ...

...
...
...
...
...

RATIONAL

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 months or less _____ 1 year _____ 3 years or more _____

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

6 months or less _____ 1 year _____ 3 years or more _____

Name (optional) _____ Date _____
Company _____
Address _____
City _____ State _____ ZIP Code _____

Please return this form to: **Publications Department**
Rational
1501 Salado Drive
Mountain View, CA 94043

1A 10

2 11 10 2

of the ... also ... only ... the ... the ... the ...

of the ... the ... the ... the ...

of the ... the ... the ... the ...

of the ... the ... the ...

of the ... the ... the ...

of the ... the ... the ...

of the ... the ... the ...

of the ... the ... the ...

of the ... the ... the ...

Political ...
Political ...
is ...
Municipal ...

Political ...

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 months or less _____ 1 year _____ 3 years or more _____

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

6 months or less _____ 1 year _____ 3 years or more _____

Name (optional) _____ Date _____
Company _____
Address _____
City _____ State _____ ZIP Code _____

Please return this form to:

**Publications Department
Rational
1501 Salado Drive
Mountain View, CA 94043**

