

## Chapter 2

# The Editor Environment

t General Rules in the Editor

t Menu Descriptions

t Window Descriptions

This chapter provides a broad and detailed overview of the features and behavior of the Prograph editor environment.

t

## General Rules in the Editor

This section describes the general rules of the Prograph editor, which are not repeated in other sections of this Reference manual. However, if one of these rules does not apply in certain circumstances, it is discussed in detail in the appropriate section. In this chapter, “window” means “Prograph edit window.”

## Launching Prograph Documents

Double-clicking a Prograph document on the Macintosh desktop will either launch Prograph with the user's application running, or with the Prograph editor active, depending on the current setting of the Run On Launch checkbox in the Options... dialog. The current setting can be defeated by holding down either the mouse button or the command key while Prograph is launching. See the description of Options... in the section below on “Menu Descriptions.”

## Prograph Extensions Folder

The Prograph Extensions folder contains three types of files, each with its own icon (as shown below), that add functionality to Prograph: help files, primitive files and external definition files.

The help files contain the contents of the Balloon Help messages for the Interpreter and Compiler running under System 7. The primitive files contain the code for Prograph primitives as described throughout the documentation. The external definition files contain the code and other information needed to access C or Pascal code from Prograph. For instance, the Macintosh Toolbox interfaces for Prograph are defined in this way.

---

NOTE: The Prograph Extensions folder or an alias to it must reside in the same folder as the Prograph and Compiler applications or aliases.

---

## When Prograph Is Launched<sup>26</sup>

The very first time Prograph is launched it builds a table of references to those files.

This table is kept in the Prograph Preferences file in the Preferences folder in the System folder. Any time that Prograph is launched, if something has changed in the Prograph Extensions folder (files have been added or removed, or a file's modification date has changed), Prograph will rebuild that table. Messages showing which files are being used to construct this table will be displayed, if Startup Messages has been checked in the Prograph Options dialog.<sup>26</sup>

Only files at the top level of the Prograph Extensions folder will be used to build the table in the Prograph Preferences file; files in folders within the Prograph Extensions folder will be ignored. Under System 7, aliases within the folder will be recognized.

Once the Prograph Preferences file is built, all the mac methods, mac globals, mac constants and mac fields defined in the table are available for use in your program. The Info window in Prograph contains a complete list of what is currently defined in the Prograph Preferences file.<sup>26</sup>

A Prograph file may contain references to primitives or mac interfaces that are no longer present in your Prograph Extensions folder. In this case, when the file is loaded into Prograph, a message will be displayed and a text window will open with a list of the undefined elements. This window may be saved as a Prograph text file or printed.<sup>27</sup>

If a mac structure definition cannot be found, the file cannot even be loaded, and the message will indicate this.

If primitives, mac methods or other mac definitions cannot be found, the operations that reference them are de-annotated—changed to simpler operations with the same names. For instance, a primitive operation is changed to a Universal operation, a Mac Global operation is changed to a Persistent operation, a Mac Constant operation is changed to a Constant operation, and so on.

---

WARNING: It is not advisable to save the file with these undefined operations, because they will be saved in their de-annotated forms. <sup>27</sup>

---

## Creating Prograph Elements<sup>27</sup>

Prograph Classic gives you a choice of how to create new Prograph elements (such as class or operation icons). Depending on the setting of the Cmd-click to create Prograph objects checkbox in the Prograph Options dialog (accessed from the Info menu), you can either :

u  
click in unoccupied space in a window to create a new element, or

u  
Command-click in unoccupied space (keeping the Command key held down as you click).

This second option is the default. In either case clicking on an existing element selects it. However, with the simple click option, clicking in space does not de-select the currently selected object(s), but rather creates a new object. To be consistent with the behavior of the Macintosh Finder, where clicking in space de-selects, you may prefer to keep the Cmd-click option set.<sup>27</sup>

To set this option, if it's not already set:

u

Double-click the Prograph Classic application icon to load Prograph.

u

Select Options... from the Info menu to display the Options dialog.

u

Click the check box to activate the Cmd-click option.

u

Click the Save Settings button in the Options dialog.

Note that with the Cmd-click option on, the process of creating a datalink from an existing root to a new terminal is as follows:

u

Click on the root to select it. Note that the pointer turns into an arrow when directly over terminals or roots.

u

Keeping the Command key as well as the Option key down, move the mouse (without pressing the mouse button) to where you want to create the new terminal.

u

Click in the appropriate spot to create the new terminal with connected datalink.

The newly created element is appropriate to the window and to the location of the click in relation to other elements. It also becomes the selected element. An exception to this rule is discussed in “The Case Window” section of this chapter.

## Automatic Scrolling

If the cursor is moved outside the window boundary during dragging, the window is scrolled, if possible, to keep the cursor within the visible region.

## Selection

When referring to “selected elements” in the following, we mean “elements selected in the front window.”

## Appearance

Selected Prograph elements are highlighted (red on color monitors).

## Click

Clicking on an element selects it and deselects previously selected elements.

## Shift-Click

Clicking on an element while pressing the Shift key toggles selection of the element without affecting other selected elements. Thus elements can be multiply selected.

## Marquee<sup>29</sup>

0

Dragging in space creates a marquee for multiply selecting enclosed elements.

0

Shift-dragging in space creates a marquee that adds enclosed elements to the selected set.

## Dragging<sup>29</sup>

Dragging an unselected element moves it in the window in the usual way. Dragging a selected element moves all selected elements in the window so that they maintain the same positions relative to each other.

When elements are dragged, all associated links (datalinks or inheritance links) are adjusted so that the topology of the structure depicted in the window is preserved.

Comments associated with dragged elements are moved to remain in the same positions relative to the elements.

If the final cursor position is outside the window boundary while dragging elements and automatic scrolling is not possible, the drag has no effect; that is, the elements remain in their original positions.

## Nudging<sup>29</sup>

Option-arrow key nudges a selected Prograph element(s) in an editor window, or a window-item in the Window editor, one pixel at a time.

## Zoom and Shrink-Wrap<sup>29</sup>

Zoom expands the window only far enough so it doesn't cover the trash can icon. Option-zoom (shrink-wrap) makes the window large enough to show all of its contents, or the size of the screen, whichever is smaller. Command-Option-zoom moves the window contents to its upper left corner.

## Editing Names<sup>30</sup>

When an element that has a name field is the only selected element, the text editor is active on its name. When such an element is created, its name is the empty string.

## Linking/Unlinking<sup>30</sup>

In a Case or Classes window, if a single element is selected, pressing the Option key produces a rubber band of an appropriate kind from the selected element to the cursor position. Clicking on another element either creates a link to it from the selected element or removes an existing link.

## Opening<sup>30</sup>

Double-clicking on an element opens an appropriate window. The type of window depends on the type of the element and on the position of the click on it.

## Commenting<sup>30\*</sup>

Command-clicking on an element toggles the display of its associated comment. If no comment exists, Command-click creates one, the empty string, and activates the text editor on it. Clicking on a displayed comment activates the text editor on it. If the text editor is active on a comment, then no other comments have an active text editor, and no elements are selected. Dragging a comment moves it in relation to its associated element, except when the final cursor position is outside the window boundary and automatic scrolling is not possible.

## Manipulation of Groups of Selected Elements<sup>30\*</sup>

Groups of elements can be dragged.

Selecting a menu item applies the associated action to every selected element and is equivalent to separately applying the action to each element of the group. There are, however, exceptions to this rule, which are discussed in “The Case Window” section and “The Classes Window” section, both below in this chapter.

If an action cannot be applied to some element of the group, then applying it to the group is disallowed. For example, since an input bar cannot be deleted from a case, the menu item Delete Object is unavailable if a group of operations including the input bar is selected in a case window.<sup>30\*</sup>

## Combinations<sup>31\*</sup>

Actions described above can be combined where appropriate, producing combined effects.

When a click in space occurs while the Command key is depressed, an element is created together with an associated comment, with the text editor active on the comment.

Combining certain actions, however, does not produce combined effects. For example, if a click in space is accompanied by depressed Option and Command keys, a new element is created without a comment, no link is constructed, and any element that was selected before the click remains selected.

## Error Prevention<sup>31\*</sup>

If the normal effect of a cursor click would result in a syntactically incorrect program, the Editor does not perform the expected action, but issues either one or two beeps. If one beep is sounded, no error message is given. If two beeps are sounded, an error message can be viewed by selecting Last Error... from the Info menu.

Syntactic errors that could result from menu selections are prevented by making inappropriate menu items unavailable.