

t Menus^{-113*}

The menu items that are relevant to the interpreter are Switch Context in the Apple menu and all the menu items in the Exec menu, except Edit Application.

Apple Menu^{-115*}

Switch Context^{-115*}

Choosing this item explicitly switches the context from the editor/interpreter to the application and vice versa. Any execution in progress will be interrupted.

Enter Editor^{-115*}

This opens the appropriate editor onto the current frontmost window. See chapter 4, "Application Builder".

Exec Menu^{-115*}

For convenience in explaining these menu items, we say that a window designates a particular method either if the window is an editing case window onto the method, or if the method icon is selected in the window. We also say that a window designates a particular stack element either if the window is an execution case window onto the element, or if the element is selected in the Stack window.

un^{-116*}

If System classes are not present, and if no method has been specified with Set Program (see the “Set Program” section below in this chapter), the Run menu item is disabled.

If System classes are present, and no method has been specified with Set Program, Run changes to Run Initial and is enabled. Selecting this item switches to application context, executes the universal method Initial (referred to as the run method), and activates the current application.

If another method has been specified using Set Program, that method is executed.

The Run menu item changes to Stop Running during execution.

Execute Method^{-116*}

This item is applicable when exactly one method is designated by the active window. Exec is started on the designated method.

Abort^{-116*}

Choosing this item aborts all current executions and switches to the editor/interpreter context.

Set Program^{-117*}

This item is available when the active window is the Universal Methods window and one icon in it is selected. The selected method becomes the run method, and its name is appended to the menu item Run. If the name of this method is subsequently edited, it is also changed in the Run menu item.

If a Prograph program with a run method is saved, its behavior when double-clicked from the desktop will depend on the current setting of the Run on Launch item in the Options... dialog.

If Run on Launch is checked, the run method is automatically executed when the Prograph document is double-clicked. This automatic execution can be avoided by keeping the mouse button or command key pressed while Prograph loads the saved file: in this case Prograph will open with the editor, rather than your application, active.^{-117*}

If Run on Launch is not checked the behavior is exactly the opposite. Double-clicking a Prograph document will launch Prograph with the editor active, while keeping the mouse button or command key pressed down during launch will cause Prograph to launch with your application active, and its run method executing.

Clear Program^{-117*}

Choosing this item removes the run method name from the Run item. The application does not commence its execution from any specific method. The same occurs if the run method is deleted.

Step/Show Level...^{-117*}

This item is available when the active window designates at least one method or stack element. A dialog opens showing the current execution level of the designated methods or stack elements. The user can change the level of debugging of the designated methods or stack elements by checking boxes in this dialog and dismissing it with the Set Level button. The Cancel button dismisses the dialog without changing the debugging level.

¹¹⁸If Show Cases is checked, execution case windows open during execution of the designated methods or stack elements. Checking Show Cases also enables the Single Step check box in the dialog.

If Single Step is checked, then during execution of the designated methods or stack elements, the interpreter executes one operation each time the Return key is pressed, provided that either an execution window onto the operation's case is open or the Stack window is open.

If neither box is checked, then execution occurs without opening execution case windows, and without user intervention. ¹¹⁸

When this dialog applies to several methods or stack elements having different Step/Show levels, one or both check boxes are grayed, for example: ¹¹⁸

When the level of debugging is changed for a method, the debugging level of existing copies of the method on the stack is not affected. However, subsequent invocations execute according to the new setting. Thus, a change in the debug level of a designated method made from an execution window is temporary, effective for that execution only, while a change in the debug level of the designated method made from an edit window is permanent.

Step/Show On, Step/Show Off ¹¹⁸

These items are applicable in the same circumstances as the menu item Step/Show Level.... Their effects are equivalent to selecting Step/Show Level..., checking (or unchecking) both boxes on the ensuing dialog, and dismissing the dialog by clicking the Set Level button. ¹¹⁸

Breakpoint On ¹¹⁹

Available when an editing case window with at least one selected operation is the active window, this item sets breakpoints on the selected operations. When execution reaches an operation with a Breakpoint, the execution case window opens with the operation selected, indicating that it is next to be executed, and Single Step is turned on for the window.

Breakpoint Off¹¹⁹

This item is available when an editing case window with at least one selected operation is the active window. It removes breakpoints from the selected operations.

Clear Steps & Breaks¹¹⁹

This menu item is applicable either when the active window is the Classes window with at least one selected class, or when the active window designates at least one method or stack element. In the first situation, Step/Show and Breakpoint are turned off for all methods in all classes selected in the Classes window. In the second situation, Step/Show and Breakpoint are turned off for all designated methods or stack elements. This process is propagated through all levels of locals within the affected methods or stack elements.

Trace¹¹⁹

When Trace is checked, the debug level setting of all methods is ignored, and Single Step and Show Cases are set “On” for every stack element as it is added to the stack. When Trace is not checked, the debug level of each element as it is added to the stack is determined by the debug level of the corresponding method.

The effect of Trace, however, is overridden if Debug is not checked.

Debug¹¹⁹

When Debug is checked, the debug level of each element as it is added to the stack is determined by the setting of Trace and the debug level of the corresponding method. When Debug is not checked, the setting of Trace and the debug level settings and breakpoints of all methods are ignored, and Single Step and Show Cases are set “Off” for every stack element as it is added to the stack.