

# t

## Menu Descriptions<sup>-31\*</sup>

### General Features<sup>-31\*</sup>

Frequently used menu items have key equivalents, which appear to the right of the item in the menu. Key equivalents can combine characters with the modifier keys Command, Option and Shift for which the following symbols are used:

### he Apple Menu<sup>-32\*</sup>

Aside from providing the usual access to Desk Accessories, this menu contains three items specific to the Prograph environment.

### bout Prograph...<sup>-32\*</sup>

Displays information about Prograph, such as version of release.

### Switch Context<sup>-32\*</sup>

Explicitly switches context between the editor/interpreter and the application. See chapter 3, “The Interpreter Environment.”

### Enter Editor<sup>-32\*</sup>

Opens the appropriate editor onto the current frontmost application window. See chapter 4, “Application Builder.”

### The File Menu<sup>-32\*</sup>

ew<sup>133</sup>

Creates a new Prograph file. If changes have been made to the current file since it was last saved, a dialog opens asking the user whether or not to save these changes. If, in the Options... dialog an Initial File name was given, and the Load File on New check box was checked, that file will be loaded in.

Open...<sup>133</sup>

Opens a Prograph file. A modal dialog is displayed with a scrolling list of names of all Prograph files from the current folder. This list always has one, and only one, selected name, referred to below as the disk file.

---

NOTE: The Prograph document you open must be compatible with the version of Prograph you are running. File translators are provided to translate Prograph files created with earlier versions of Prograph to the current version. Refer to the section on "Prograph File Translators" for details.

---

licking Open on the dialog has different effects depending on the values of the two check boxes, which have three possible combinations, as follows.<sup>133</sup>

Selective Load  
Incremental Load

If neither Selective Load or Incremental Load has been checked, then the disk file is loaded. If changes have been made to the current file since it was last saved, a dialog is opened asking the user whether or not to save these changes.<sup>133</sup>

Selective Load  
Incremental Load

If Incremental Load has been checked, each universal method and persistent from the disk file is added to

the current file. <sup>-34-</sup>

In order to specify how classes from the disk file are incorporated into the current file, we first define the notion of absorption. A class D from the disk file is absorbed by a class C in the current file if and only if:

o

D and C have the same name, and

o

the attributes of D have the same names and occur in the same order as the attributes of C, and

o

either D has no superclass and C has no superclass, or the superclass of D is absorbed by the superclass of C.

Each class from the disk file is added to the current file if and only if it is not absorbed by a class in the current file. If the superclass of an added class is absorbed by some class C, then C becomes the superclass of the added class. <sup>-34-</sup>

If a class C and a class D each contain a method M, method M from class D is loaded and renamed "M #1".

The name of a universal method, persistent, or class added to the current file is changed only if it is the same as that of an existing universal method, persistent, or class, respectively, in the current file. <sup>-34-</sup>

Selective Load

Incremental Load

If both Selective Load and Incremental Load have been checked, a dialog is displayed containing a scrolling list of names of classes, persistents, or methods in the disk file, depending on whether the front window is the Classes window, Persistents window, or any Methods window of the current file. The dialog is appropriately labeled to indicate the kind of elements displayed, as in the example below. <sup>-34-</sup>

Names in the scrolling list can be multiply selected. Clicking the Load button adds the selected elements from the disk file to the front window of the current file, according to the rules described for Incremental Load above.

**Save** <sup>-35-</sup>

Saves the current file, including the values of all attributes and persistents, all open windows, and the contents of the Object clipboard (see the description of Copy Object below). If this menu item has a file name appended to it, the current file is saved with that name. Otherwise, Save acts like Save As..., described below. If Make Backups in the Options... dialog is checked, the current disk version of the file will be renamed to .bak before the file is saved. <sup>-35-</sup>

---

TECHNICAL NOTE: When your Prograph document is saved, Macintosh entities which are referenced by handles (such as PICTs and ICONs) are saved. In Prograph 2.0 these were set to NULL. Note that resource information is not saved. That is, if you loaded a PICT from a resource, its resource ID and name are part of that resource, but not part of the PICT data structure itself. If you set an attribute value to be a handle to that PICT, the PICT structure will be saved when you save your Prograph document file, and the handle to that PICT will be restored when you re-load the file into the interpreter environment.

---

## Save As...<sup>35</sup>

Opens a dialog allowing the user to specify a name under which to save the current file.

## New Text...<sup>36</sup>

Creates a new Text window. A dialog is opened in which the user can specify a name for the new window, and whether or not text should be wrapped in this window.

licking the OK button creates a new Text window, which has the name in its banner. If Use Text Wrap has been checked, line wrapping takes place; otherwise the window appears with an active horizontal scroll bar.

## Open Text...<sup>36</sup>

Opens a text file, not necessarily one created by Prograph, placing its contents in a new Text window. A dialog is displayed with a scrolling list of the names of text files in the current folder. The user can select one of these names and specify whether or not text should be wrapped in the new window. Clicking the Open button creates a new Text window, which has the name of the selected file in its banner; line wrapping is in effect if Use Text Wrap has been checked.

## Save Text<sup>37</sup>

Available only if the front window is a Text window, this item saves the contents of a Text window as a file with the name in the banner of the Text window. If the banner displays the empty string, the file is named

Untitled.

## Save Text As...<sup>37</sup>

Available only if the front window is a Text window, this item opens a dialog allowing the user to specify a name under which to save the contents of the Text window.

## Page Setup...<sup>37</sup>

Sets printer parameters in the usual way.

## Print...<sup>37</sup>

Available only if the front window is a Prograph edit window or a Text window, this item initiates printing, with printing options depending on the front window, as described below.

During printing of Constant and Match values in a case, if the length of a value is longer than that set in the Options... dialog, it is replaced by a footnote number referring to a footnote in which the value is printed in its entirety.

## Classes Window<sup>37</sup>

If this window is frontmost, the following dialog appears.

Clicking the OK button causes program elements to be printed depending on which boxes are checked, as follows:<sup>38</sup>

### classes

Prints icons for all classes in the top window.

### methods of selected classes

Prints icons for all methods in the selected classes.

### cases<sup>38</sup>

Prints all cases of all methods from the selected classes.

#### locals

Prints all cases of all locals in all methods of selected classes.

#### attributes of selected classes<sup>-38\*</sup>

Prints icons for attributes of selected classes.

#### Separate Pages

Prints at most one method, local or other Prograph element per page. If not checked, will try to conserve paper, printing as many different classes, methods, etc. on a page as will fit.

#### Any Methods Window<sup>-38\*</sup>

If such a window is frontmost, the following dialog appears:

licking the OK button causes program elements to be printed depending on which boxes are checked, as follows:<sup>-39\*</sup>

#### methods

Prints icons for all methods in the window.

#### cases of selected methods

Prints all cases of all selected methods.

#### locals of selected methods<sup>-39\*</sup>

Prints all cases of all locals in all selected methods.

#### Separate Pages

Prints at most one method or local per page. If not checked, will try to conserve paper, printing as many different methods and locals on a page as will fit.

#### Case Window<sup>-39\*</sup>

The Case window may belong to a method or to a local. The contents of the window will be printed as specified by the following dialog. Checking cases of selected locals causes the printout to include the case windows of the selected locals. The current, frontmost case window will also be printed if the this case checkbox is checked.

Attribute, Persistents, or Text Window<sup>35</sup>

Prints the content of the window.

Launch...<sup>36</sup>

Launches another application. A modal dialog is displayed with a scrolling list of names of all applications in the current folder. In Finder, Prograph will first quit.

Quit<sup>40</sup>

Quits from Prograph. If changes have been made to the current file since it was last saved, a dialog is opened asking the user whether or not to save these changes.

The Edit Menu<sup>40</sup>

ndo<sup>40</sup>

The Undo menu item is available only after an object has been deleted. At all other times, this menu item is disabled.

Cut, Copy, Paste, Clear<sup>40</sup>

These perform the standard editing operations on text. When in the Window Editor (compare chapter 4, “Application Builder”), these menu items are available for moving PICTs and ICONs between the

clipboard and PICT and ICON window items.

## Cut Object<sup>41</sup>

Equivalent to Copy Object followed by Delete Object.

## Copy Object<sup>41</sup>

Copies the selected elements and all their dependent elements into the Object clipboard, a storage mechanism analogous to the Macintosh clipboard but internal to Prograph. The content of the Object clipboard is retained until the next Copy Object or Cut Object. When a file is saved, the content of the Object clipboard is saved in the file. Later sections in this manual explain in more detail the effect of applying this menu item to different kinds of elements.

## Paste Object<sup>41</sup>

Copies the contents of the Object clipboard into the top window. If there is a name conflict between an element in the window and one from the Object clipboard, the pasted element is renamed. Later sections in this manual explain in more detail the effect of applying this menu item in different windows.

## Delete Object<sup>41</sup>

Deletes the selected elements and all their dependent elements. All windows displaying the structures of deleted elements are closed.

## Replicate Object<sup>41</sup>

Equivalent to Copy Object followed by Paste Object, except that the structure is not copied into the Object clipboard. The previous content of the Object clipboard is therefore retained.

The copied elements are placed slightly offset from the originals.

## Select All Objects<sup>41</sup>

Selects all elements in the front window.

## Copy Value<sup>41</sup>

If a Value window is the front window, Copy Value copies the content of this window to the Value clipboard, a storage mechanism analogous to the Macintosh clipboard but internal to Prograph. The content of the Value clipboard is retained until the next Copy Value but is not saved with the file.

If the Attributes window of a class is the front window, then Copy Value makes an instance of this class in the Value clipboard.

## Replace Value<sup>42</sup>

If a Value window is the front window, Replace Value replaces the content of this window with a copy of the content of the Value clipboard. The default value in the Value clipboard is NULL.

If an Attributes window of a class is the front window, then Replace Value is available only if the content of the Value clipboard is an instance of this class, its ancestors, or any of its descendants. The values of attributes in the window are replaced by values of corresponding attributes of the instance in the Value



clipboard.

## Propagate Attribute<sup>42</sup>

Copies the values of the selected attributes of a class to the corresponding attributes of all its descendants. Propagating instance attributes will thus change the default values of instance attributes in descendant classes.

## Bring Closer<sup>42</sup>

This menu item, and Send Further, are active only when a Window Item is selected in the Window Editor. Bring Closer brings the selected Window Item closer to the foreground. The Window Item that is closest to the foreground (that is, the first item in the window's item list) will, if it is a button, be highlighted with a double-heavy line to indicate that it is the default button.

## Send Further<sup>42</sup>

Sends the selected Window Item further into the background.