

t Window Descriptions^{72*}

In this section the edit windows are described, and allowable user actions in these windows discussed. We use the word “window” here to mean “edit window.”

Every window has a close box, a zoom box, a grow box, and a banner containing a title consisting of an icon and a name, according to kind and identity of the structure displayed in the window. All windows except Value windows are nonmodal. Value windows are “movable modal”: they may be dragged, and certain menus (Info, Edit and Find) remain enabled.

In this section, we first discuss the effects of actions on single selected elements. We then note how these effects generalize to groups of selected elements only if there is something special about the generalization.

The Universal Methods Window^{72*}

This window displays all universal methods. In this section, by “method” we mean “universal method.”

pening^{73*}

This window appears when Universal Methods is selected from the Windows menu.

Creating^{73*}

Clicking in space in the window creates a method icon with an insertion point below the icon for typing its name.

Deleting^{73*}

All Case windows of a deleted method are automatically closed.

Naming^{73*}

If the name of a method is a nonempty string, it must be distinct from the name of other methods and Mac Methods. A method can, however, have the same name as a primitive. Note that the Prograph Classic compiler requires all methods to have nonempty names.

Editing a Method^{73*}

Double-clicking a method opens a window on its first case.

Combinations^{73*}

Double-clicking in space creates a new method and opens a window on its first case.

The Classes Window^{74*}

This window displays all classes. A parent class which contains hidden subclasses has an icon distinguished by an additional hexagonal border.

Opening^{74*}

The Classes window is opened when one of the following actions occurs:

- o when Classes is selected from the Windows menu;
- o when the left end of an Instance generator for a nonexistent class is double-clicked.

Creating^{74*}

Cmd-clicking in free space creates a class icon with an insertion point below its icon for typing its name.

Showing/Hiding Classes^{74*}

Selecting a subtree of classes, and then selecting Hide Classes from the Info menu, causes the subclass icons to disappear, and the parent class icon to assume a distinctive form, with an additional hexagonal border. Selecting a class which has hidden subclasses, and then selecting Show Classes from the Info menu, causes the “first generation” of subclasses to appear (that is, subclasses which themselves have hidden subclasses are shown as “encapsulating” class icons: their subclasses are not shown).

Editing^{75*}

Delete^{75*}

The Attributes and Methods windows of the deleted class and Case windows on methods of the class are closed.

Deletion of a class removes inheritance links to its subclasses and removes all inherited attributes from the subclasses.

A class cannot be deleted if:

- o an instance exists of the class or any of its successors, or
- o the class is a System class.

Copy Object^{75*}

This copies the selected class and all its ancestors into the Object clipboard. For example, if Copy Object is applied to class “e,” the classes “a,” “c,” and “e” are copied, together with their inheritance links.

Paste Object^{75*}

This copies the class structure from the Object clipboard to the Classes window. The pasted classes are added or absorbed as described above in the discussion of incremental loading.

Replicate Object^{75*}

Produces a new copy of the class. The copy is renamed and added as a subclass of the superclass of the original class.

Cut Object^{75*}

This is equivalent to the menu item Copy Object followed by Delete Object.

Naming^{76*}

Classes must have unique names. The names of System classes cannot be changed. Do not use the names of any Mac types (such as Ptr, and Handle) or any of the following names: array, attribute, class, method, number, natural, object, persistent, text.

Connecting / Disconnecting^{78*}

If class A is a selected class, Option-clicking on a class B creates an inheritance link from A to B. Class B and all its descendants inherit the attributes of A and their default values.

If B previously had a superclass C, then the link from C to B is deleted.

Connection is not permitted if B has an attribute with the same name as that of an attribute in A.

NOTE: Inheritance in System classes cannot be altered.

Opening Classes^{79*}

A class icon is divided into two halves, each of which can be opened. A double-click:

- o on the left side opens an Attributes window for the attributes of that class
- o on the right side opens a Methods window for the methods of that class.

Combinations⁷⁶

If A is the selected class, Option-clicking in space creates a class B and an inheritance link from A to B.

Group Selection⁷⁶

Copy Object⁷⁶

Copies into the Object clipboard the smallest subtrees of classes that includes all classes in the group and their ancestors.

Replicate Object⁷⁶

The classes that are replicated are the classes in the group, together with those classes that lie on a path of inheritance links between two classes of the group. A new copy of each subtree consisting of these classes is added; the added classes are renamed; and the root class of each added subtree is made a subclass of the superclass of the class of which it is a copy.

The Class Methods Window⁷⁷

This window displays the methods of the class whose name appears in the window banner.

pening⁷⁷

The window is opened in response to one of the following actions:

- o double-clicking on the right side of a class icon
- o double-clicking on the right side of an Instance generator for an existing class⁷⁷
- o selecting Methods... from the Windows menu , and then selecting a class from the dialog displaying a scrolling list of names of classes.

Selecting a member of the list and clicking the Select button opens the Methods window for the class and

dismisses the dialog. This is equivalent to double-clicking a member of the list.

Creating^{77*}

Cmd-clicking in space creates a method icon with an insertion point below the icon for typing its name.

Deleting^{78*}

All opened Case windows of a deleted method are automatically closed.

Naming^{78*}

Four kinds of methods can be created in this window: Simple, Get, Set, and Initialization. Methods within each of these categories must have unique names.

Methods of two different categories can have the same name, however, and the name of a method in a class can be the same as that of a primitive or Mac Method.

The name «» of an Initialization method cannot be edited.

Transforming a Class Method^{78*}

A selected class method icon can be transformed into a Get, Set, or Initialization method by annotating it with the appropriate item from the Oper menu, namely Get, Set or Instance. Annotating a method using Instance changes its name to «».

Editing a Method^{78*}

Double-clicking a method opens a window on its first case.

Combinations^{78*}

Cmd-double-clicking in space creates a new method and opens a window on its first case.

The Attributes Window^{79*}

This window displays the class and instance attributes of the class whose name appears in the window banner.

pening^{79*}

An Attributes window is opened in response to one of the following actions:

- o double-clicking on the left side of a class icon
- o double-clicking on the left side of an Instance generator for an existing class
- o selecting Attributes... from the Windows menu opens a dialog displaying a scrolling list of names of classes, in exactly the same way as when the Methods... menu item is selected (see above).

Attribute List Areas⁷⁹

Class attributes are located above the fuzzy gray line (green on a color monitor) and instance attributes below. Class attributes have hexagonal icons, while instance attributes have triangular icons.

Creating⁷⁹

Cmd-clicking in space above the fuzzy gray line creates an unnamed class attribute with a default value of NULL. Cmd-clicking in space below the gray line creates an unnamed instance attribute with a default value of NULL. If other attributes exist, the relative position of the new attribute to the others depends on the position of the click. Inherited attributes, however, always precede the locally defined attributes.

Deleting⁸⁰

Neither system attributes nor inherited attributes can be deleted.

Editing⁸⁰

Attributes can be cut, copied, pasted, and replicated.

Naming⁸⁰

The name of an attribute, if it is a nonempty string, must be unique. The names of system attributes and inherited attributes cannot be edited. The word “default” is reserved by Prograph and should not be used as an attribute name.

Commenting⁸⁰

Comments on attributes cannot be dragged.

Changing Default Values⁸⁰

Double-clicking on an attribute opens a Value window displaying the default value of the attribute; this can be edited (see the “Value Window” section below in this chapter).

Some system attribute values cannot be edited (see chapter 5, “System Classes”).

Dragging⁸⁰

The restrictions on dragging are:

- o Dragging occurs in the vertical direction only.
- o A class attribute cannot be dragged below the gray line.

- o An instance attribute cannot be dragged above the gray line.
- o Inherited attributes cannot be dragged.
- o Locally defined attributes cannot be dragged into inherited attributes.

Combinations⁸¹

Cmd-double-clicking in space creates an attribute and opens a Value window onto its default value.

Group selection⁸¹

Dragging⁸¹

Though a group of attributes can be selected, only one can be dragged.

The Persistents Window⁸¹

This window displays all persistents.

opening⁸¹

This window is opened when Persistents is selected from the Windows menu.

Creating⁸¹

Cmd-clicking in space creates a new persistent icon with a default value of NULL and an insertion point below its icon for typing its name.

Naming⁸¹

Persistents must have unique names.

Changing Values⁸¹

Double-clicking on a persistent opens a Value window for displaying and editing the value of the persistent. Persistents are saved along with their Prograph document.