

The Tools Menu¹⁴²

This menu is used for locating operations by name in Case windows, for locating strings in Text windows, and for replacing the names of operations and replacing strings. It is also used for making methods compact or execute-only.

ind...¹⁴³

Opens the following dialog, in which various search criteria can be set:

ind, Replace¹⁴⁵

These editable text fields display the search and replacement strings respectively.

Cancel and Retain Buttons¹⁴⁵

Clicking either of these buttons dismisses the dialog without performing a search. However, Retain saves changes made in the search criteria, while Cancel does not.

Find Button¹⁴⁵

Clicking this button dismisses the dialog and starts a search for an item that satisfies the search criteria. If such an item is found, the window containing it is opened with the item selected. Otherwise a warning beep sounds and an appropriate message is generated.

By checking boxes in the dialog the user can:¹⁴⁶

- ☐ define the scope of search
- ☐ specify criteria for string matching

0

modify the display of windows during search

The check boxes that determine the scope of search are as follows:⁻⁴⁴⁻

Simple/Primitive, Local/Evaluate, Constant/Match, Persistent, Instance, Get/Set, Macintosh⁻⁴⁴⁻

These specify the kinds of operations that are acceptable targets for the search.

Text Attribute⁻⁴⁴⁻

If this box is checked, all attributes of type String in all existing instances are investigated for matching strings. Also, when this box is checked, the following boxes are gray and unchecked: Simple/Primitive, Local/Evaluate, Constant/Match, Persistent, Instance, Get/Set, Macintosh, Match Arity, Current Method Only, Current Class Only, and Close Windows.

Text Window⁻⁴⁴⁻

If this box is checked, Text windows are investigated for matching strings. Also, when this box is checked, all boxes that relate to searching for operations are gray and unchecked, namely Simple/Primitive, Local/Evaluate, Constant/Match, Persistent, Instance, Get/Set, Macintosh, Match Arity, Current Method Only, and Current Class Only.

Ignore Upper/Lower⁻⁴⁴⁻

If this box is checked, matching does not distinguish between lower and upper case. Otherwise, matching is case sensitive.

Match Pattern⁻⁴⁴⁻

If this box is checked, the search and Set strings are treated as patterns, in which occurrences of the asterisk (*) and question mark (?) characters have special status. A string matches the match pattern if the pattern can be made identical to it by substituting a string for each asterisk and a single character for each question mark.

The string substituted for a string or question mark is called its match value. If there is more than one set of substitute strings that satisfy these conditions, the set chosen is the one that minimizes the lengths of substitute strings for occurrences of the asterisk, starting from the left of the match pattern.

In searching for operations, a match succeeds only if the search pattern matches the entire name of an operation. In searching for text attributes, a match succeeds only if the search pattern matches the entire value of an attribute. In searching Text windows, however, matching succeeds if any substring of the window contents matches the search pattern.⁻⁴⁵⁻

Match Arity⁻⁴⁵⁻

This box is available (that is, is not gray) only if the Find field has not been edited since the last time that the Set Selection menu item was applied to a selected operation or method.

When this box is checked, the search for operations is restricted to those with arity that matches that of the operation or method to which Set Selection was applied.

Current Method Only⁻⁴⁵⁻

If the front window is a Methods window with a single selected method or a Case window of a method,

then search is restricted to operations occurring in cases of this method.

Current Class Only ^{~45~}

If the front window is a Classes window with a single selected class or a Methods window of a class, then search is restricted to operations occurring in all methods of this class.

Close Windows ^{~45~}

If this box is checked, then selecting the menu item Find Again closes the last window opened by searching, provided that it is still the top window and does not contain the next item found.

Set Selection ^{~45~}

This item is available if the top window is a Text window or an edit window with a single selected element, and it sets the search criteria accordingly.

If the top window is a Text window, the search string is set to the selected string in the window, the Text Window box is checked, and the Match Pattern box is unchecked.

If the top window is an edit window, the search string is set to a pattern matching the name of the selected element, and other search criteria are set depending on the selected element. For example, if a Set Selection is chosen when the front window is Persistents and has a single selected persistent called "sales," then Find... displays the following: ^{~45~}

Find Again ^{~45~}

Finds the next operation or string in the scope of the search.

Replace ^{~45~}

If the top window contains a single selected operation, text attribute, or a selected string satisfying the search criteria, it is replaced according to the Set pattern.

If the box Match Pattern is not checked, the name of the selected operation or the selected string is replaced by the Set string. Otherwise, it is replaced by a string constructed by substituting every question mark and asterisk in the Set pattern, starting from the left, with the match value of the corresponding question mark or asterisk in the match pattern. Leftover occurrences of asterisk or question mark at the right of the Set pattern are replaced by the empty string.

Replace All⁻⁴⁶⁻

Replaces the names of all operations or strings matching the search criteria.

Compact⁻⁴⁶⁻

Prograph Classic lets you compact methods: this removes all visual information from them, with the benefit that they take up much less RAM space. However, the Prograph document file on disk still retains that visual information, making it possible to uncompact methods. A third option, to make methods execute only, permanently removes the visual information. This may be desirable if you want to ship code with some methods protected from modification by the user. ⁻⁴⁷⁻

TECHNICAL NOTE: When a Prograph method is compacted, all its visual information is stripped out and placed in a temporary storage area on disk, freeing up that portion of RAM. When you save your program, that information is saved in the resource fork of your document file. This allows loading in of methods from the file, with the visual information being loaded in only for uncompact methods. It also permits uncompacting a method, since its visual information is still out there on disk.

When one or more methods or classes have been selected, the Uncompact, Compact and Make Execute Only menu items in the Tools menu become active. ⁻⁴⁷⁻

NOTE: To Compact, Uncompact or Make Execute Only methods in classes, the classes must be visible. Methods in hidden classes will not be affected by these menu items.

Choosing Compact causes the selected methods, or all the methods in the selected classes, to be compacted as described above. The icon for a Compact method has a reduced version of the little dataflow diagram that is on a regular method. ⁻⁴⁷⁻

When executed, compacted methods behave exactly the same as uncompact methods. They can call, and be called by, uncompact methods. However, when executing in Step or Trace mode, the code inside compact methods is not made visible. Compact methods can be made the main program method (using Set Program under the Exec menu), and all the normal editing operations (Cut Object, etc.) apply to them. ⁻⁴⁷⁻

Uncompact⁻⁴⁸⁻

Selecting Uncompact from the Tools menu restores the selected methods or the methods in the selected classes to their uncompact state.

Make Execute Only⁻⁴⁸⁻

Selecting this menu item permanently removes all visual information from the selected Compact methods or from the Compact methods in the selected classes. The icon for an execute-only method is the same as that for a Compact method, with the addition of a little lock in the bottom left-hand corner.

WARNING: Before you make a method execute-only, be sure you have a backup of it in uncompact form, as you will not be able to edit it once it's been made execute-only.

The Opers Menu¹⁴⁸

Selected operations and methods are transformed by the items of this menu. The effects of annotating an operation are explained in “The Case Window” section below in this chapter.

In the following, unless otherwise noted, a control associated with an operation is preserved in any transformation by items from this menu.¹⁴⁸

imple¹⁴⁹

The operation becomes a Simple operation.

Constant¹⁴⁹

The operation becomes a Constant operation. Any associated control is removed.

Match¹⁴⁹

The operation becomes a Match. If the operation has no control, the “Next Case on Failure” control is appended.

Persistent¹⁴⁹

The operation becomes a Persistent operation.

Instance¹⁴⁹

The operation becomes an Instance generator.

The Method in a class Methods window becomes an Initialization.

Get¹⁵⁰

The operation becomes a Get attribute.

The Method in a class Methods window becomes a Get method.

Set⁵⁰

The operation becomes a Set attribute.

The Method in a class Methods window becomes a Set method.

Local⁵⁰

The operation becomes a Local operation.

Evaluate⁵⁰

The operation becomes an Evaluation.

Mac Constant⁵⁰

The operation becomes a Mac Constant operation. Any associated control is removed.

Mac Match⁵⁰

The operation becomes a Mac Match. If the operation has no control, “Next Case on Failure” is appended.

Mac Global⁵⁰

The operation becomes a Mac Global.

Mac Address⁵⁰

The operation becomes a Mac Address.

Mac Get Field⁵¹

The operation becomes a Mac Get Field.

Mac Set Field⁵¹

The operation becomes a Mac Set Field.

Opers to Local⁵¹

Transforms the group of selected operations into a Local (see “The Case Window” section below in this chapter for more details).

Local to Method...⁵¹

Transforms a single selected Local operation to a simple operation. The following

dialog is opened displaying a suggested name for the simple operation. If the Local is in a universal method, this suggested name is simply the name of the Local; otherwise, it is the name of the Local preceded by //.

ame⁵¹

This is an editable-text field for the name of the operation.

Cancel⁵¹

If clicked, this closes the dialog without performing the conversion.

OK⁵¹

When this button is clicked, if the name is not a data-determined reference, the Local is transformed into a simple operation and the corresponding new method is created in the appropriate place. If the name is a data-determined reference, a second dialog is displayed over the first, with a scrolling list of names of classes. At most one class can be selected in this list.

elect⁵²

If a class is selected, clicking this button closes both dialogs, transforms the Local into a simple operation with the specified name, and creates a corresponding method in the selected class with cases identical to those of the original Local.

Double-clicking an item in the scroll list is equivalent to first selecting it and then clicking the Select button.

Cancel⁵²

Clicking this button dismisses the topmost dialog without transforming the Local or creating a method.

In each of these situations, if the required new method cannot be created because its name would conflict with that of an existing method, the Local is not transformed, the first dialog remains open, a warning beep is sounded, and an appropriate message is generated.⁵²

The Controls Menu⁵³

Prograph operations, roots, terminals, and the output bar are annotated by items on this menu.

Operations ⁵³

Simple ⁵³

Transforms a root (or terminal) into a simple root (or terminal), and an operation into a simple operation.

Simple removes all controls and control annotations from an operation. As a consequence, all its roots (or terminals) are transformed to simple roots (or terminals).

Super ⁵³

Transforms an operation into a Super operation, changing its name into the form //<name>.

Repeat ⁵³

Transforms an operation into a Repeat multiplex.

Terminals and Roots ⁵⁴

Inject ⁵⁴

Transforms a terminal into an Inject terminal. Note that Mac Methods cannot have Inject terminals.

EDITING TIP: To add a terminal to a Persistent with an existing Inject terminal:

- u Create the first terminal.
 - u Annotate it as Inject.
 - u Drag the terminal to make the persistent operation wider.
 - u Click (or command-click) to create the second (data) terminal.
-

Partition ⁵⁴

Applied to a selected terminal of an operation with no roots, this item transforms the associated operation into a Partition multiplex, automatically adding True and False roots.

List^{-54*}

Transforms a terminal (or root) into a List terminal (or root). As a consequence, the associated operation becomes a Repeat multiplex.

Loop^{-54*}

Transforms a terminal (or root) into a Loop terminal (or root). Either a simple root (or terminal) on the associated operation is converted into the corresponding Loop root (or terminal), or a new corresponding Loop root (or terminal) is added to the operation. The associated operation becomes a Repeat multiplex.

Controls on Operations and on the Output Bar^{-54*}

The Controls—Next Case, Continue, Terminate, Finish and Fail—can be applied to selected operations and to the output bar. The general rules for these controls are:

o

If an operation already has a control, and the control is changed by selecting the appropriate menu item, the activation condition “on Success” or “on Failure” of the original control is retained in the new one.

o

If the operation has no control, and a menu item is applied to add a control to it, the new control has the activation condition “on Failure”^{-54*}

Invert^{-55*}

oggles the activation condition of the control on the operation. Note that applying this item to an operation with no control produces the control “Continue on Failure” (the default control for an operation, “Continue on Success”, does not have a control icon).

Next Case, Continue, Terminate, Finish, Fail^{-55*}

Appends a control to an operation.