

t Window Item Editors¹³⁹

There are two kinds of Window Item editors: a generic WYSIWYG Window Item editor that applies to all subclasses of Window Item, and one Specification editor for each of these classes.

One Window Item editor is invoked for each «Window Item» in the item list of a «Window» when doing WYSIWYG editing of the «Window». The editable Window Item images appear appropriately placed in the window image of the «Window».

A Specification editor for a «Window Item» can be invoked only by double-clicking a Window Item image, or by a double-click that creates a Window Item image.

NOTE: All editor dialogs are compatible with System 7 and TrueType fonts, allowing you to set the font and size separately. Any size compatible with the fonts in your particular system can be entered in the size field.

WYSIWYG Window Item Editor¹³⁹

The editing actions that can be performed on Window Item images are as follows:¹³⁹

o

A double-click opens the appropriate Specification editor for the «Window Item».¹⁴⁰

o

A Shift-Option-Command-click opens the Value window for the window item's instance (that is, you can directly examine its attributes).

o

An Option-double-click opens the Window Item Specification editor, which allows you to change the class of the «Window Item».

o

A click-drag on the bottom right corner resizes the Window Item image. The size is aligned to an 8x8 grid. If the item is created by a click-drag, the drag resizes the newly created item.¹⁴⁰

o

A click-drag elsewhere in the Window Item image repositions it in the window image. The location is aligned to an 8x8 grid.

o

An Option-click-drag repositions and resizes without grid restriction.

o

A Shift-click-drag restricts the vertical or horizontal repositioning or resizing according to the initial direction of the drag. Again, the result is aligned to an 8x8 grid unless the Option key is also pressed.¹⁴⁰

Resizing a Window Item image or repositioning it in the window image of a «Window» sets the values of the attributes size and location of «Window Item» so that, when the «Window» is activated and displayed

on the screen, the pictorial representation of the «Window Item» appears at the same location and is the same size as its image in the window image.

When the WYSIWYG Window Item editor is applied to a «Window Item» for which FALSE is the value of the attribute visible?, it appears in the window image of the «Window» in its unselected state as follows:

In the following description of the Specification editors for «Window Item»s, we describe the unselected appearance of Window Item images corresponding to visible «Window Item»s. We also describe the appearance of each «Window Item» when its «Window» is active. When a Window Item image is selected, it has a gray rectangle with a “grow spot” at the bottom right corner superimposed on it.

Window Item^{-140*}

Aside from the usual ways of invoking Specification editors on «Window Item»s, the Window Item Specification editor can also be invoked by an Option-double-click on a Window Item image, regardless of the class of the «Window Item». This allows the developer to change the class of the «Window Item».^{-141*}

The Window Item Specification editor has a scrolling list displaying the names of all subclasses of Window Item, only one of which can be selected. A selection is made by clicking on a name or typing its first few characters. The selected name specifies the class to which the «Window Item» is converted if the OK button is clicked.^{-141*}

The Window Item image appears as follows when editing an instance of this class:^{-141*}

When the «Window» is active, the instance of Window Item is invisible.

Click Item^{-141*}

The Click Item Specification editor has the following items:

Click Name, Click Method^{-141*}

Editable-text fields displaying the values of attributes name and click method respectively of «Click Item»

Active, Visible, Move w/Window, Grow w/Window ^{~141~}

These are checked if and only if the attributes active?, visible?, move?, and grow? respectively, of «Click Item» are TRUE.

Balloon... ^{~141~}

Button for opening Balloon Help dialog. ^{~142~}

When editing a visible instance of Click Item, its image appears as: ^{~142~}

When the «Window» is active the «Click Item» also has this appearance.

Button ^{~142~}

The Button Specification editor is exactly analogous to the Click Item Specification editor. ^{~142~}

When editing a visible instance of Button, its image appears as follows: ^{~142~}

or

depending on whether or not the «Button» is the first element of the item list of «Window» (that is, the

default button). The text displayed is the value of the attribute name of the «Button». If the «Button» is inactive, the image is grayed. ¹⁴²

When the «Window» is active, the «Button» has the same appearance. ¹⁴³

Radio Set

The Radio Set Specification editor is analogous to the Click Item Specification editor, but has an extra editable-text field, Value List, which displays the list of names for the radio buttons in the set. ¹⁴³

When editing a visible instance of Radio Set, its image appears as: ¹⁴³

The strings displayed to the right of the buttons are the strings in the value list of the «Radio Set». The selected button is the one whose name is in the value attribute of the «Radio Set». The radio buttons are displayed in columns within the bounding rectangle of the «Radio Set». If the «Radio Set» is inactive, the image is grayed.

When the «Window» is active, the «Radio Set» has the same appearance.

Check Box ¹⁴³

The Check Box Specification editor is analogous to the Click Item Specification editor, but has an extra check box, Checked, which is checked if and only if the value of attribute checked? of «Check Box» is TRUE. ¹⁴³

When editing a visible instance of Check Box, its image appears as: ¹⁴⁴

here the text to the right of the box is the value of attribute name of «Check Box», and the box contains a check mark if and only if the attribute checked? of «Check Box» has value TRUE. If the «Check Box» is inactive, the image is grayed.

When the «Window» is active, the «Check Box» has the same appearance.

Graphic⁻¹⁴⁴⁻

This class has no Specification editor. The WYSIWYG Window Item editor applied to an instance of Graphic has the same appearance as when applied to an instance of Window Item.

Pict⁻¹⁴⁴⁻

The Pict Specification editor is analogous to the Click Item Specification editor, but has an extra editable-text field, Picture ID #, which displays the value of the attribute rsrc number of the «Pict». ⁻¹⁴⁴⁻

hen editing a visible instance of Pict, its image displays the PICT whose resource number is the value of attribute rsrc number of «Pict». If this number does not correspond with a PICT in the resource fork of the file, the Pict item image displays nothing. ⁻¹⁴⁵⁻

When the «Window» is active, the «Pict» also has this appearance.

PICTs can be pasted from the Macintosh Clipboard into a selected Pict image. The PICT is saved to the resource fork of the application, with the resource number specified in the Pict specification editor. If the resource number of a pasted PICT is the same as an existing PICT resource in the application's resource fork, the new PICT replaces the existing one. ⁻¹⁴⁵⁻

When a PICT is pasted into a Pict image, the Pict image is resized to the size of the PICT in the Clipboard.

When a PICT is cut or cleared from a Pict image, the PICT is not removed from the application's resource fork; rather, a new, unique resource number is assigned to the «Pict».

Icon⁻¹⁴⁵⁻

The Icon Specification editor is analogous to the Click Item Specification editor, but has an extra editable-text field, Icon ID #, which displays the value of the attribute rsrc number of the «Icon». ⁻¹⁴⁵⁻

¹⁴⁵When editing a visible instance of Icon, its image displays the ICON whose resource number is the value of attribute rsrc number of «Icon». If this number does not correspond with an ICON in the resource fork of the file, the Icon item image shows nothing.

When the «Window» is active, the «Icon» also has this appearance.¹⁴⁵

¹⁴⁶ICONS can be pasted from the Macintosh Clipboard into a selected Icon image. The ICON is saved to the resource fork of the application, with the resource number specified in the Icon specification editor. If the resource number of a pasted ICON is the same as that of an existing ICON resource in the application's resource fork, the new ICON replaces the existing one.

When an ICON is pasted into an Icon image, the «Icon» is resized to a 32x32 square, the standard size of a Macintosh ICON.

When an ICON is cut or cleared from an Icon image, the ICON is not removed from the application's resource fork; rather, a new, unique resource number is assigned to the «Icon».¹⁴⁶

NOTE: The Scrapbook can be used for intermediate storage of ICONs. (ICONs are not visible when stored in the Scrapbook.)

Text¹⁴⁶

The Text, Edit Text, and Scroll Text editors all use the same editor dialog. However, the attributes you can set change depending on the type of item you're creating. For example, Edit Text has Wrap and Active options; these are not available in the Text editor. The Text editor allows you to create "Read-only" text fields.¹⁴⁶

he following table defines the fields and various attributes associated with the Text, Edit Text, and Scroll Text editor dialogs.¹⁴⁷

Name¹⁴⁷

This is the name that is used to identify the text item. It can be any length and usually indicates something about the contents of the text item. The name you enter is used by the primitive find-item to locate the text item in your program. Therefore, avoid using mathematical symbols or quotes in the name as these characters can cause problems if and when you attempt to use the text item name in your code.

Text¹⁴⁷

This is where you enter “default” text (text that is automatically displayed in your text item). The only limitation is that the amount of memory used to store the text cannot exceed 32K. The text you enter will automatically scroll if you exceed the limits of this field’s length. If you want to add a carriage return in the default text, you must hold down the Option key and press Return when you come to the end of a text string. To move the cursor within this field, use the arrow keys on your keyboard.

Font¹⁴⁷

This is a scrolling list of all available fonts in your system.

Size¹⁴⁷

Enter the font size. The default value of 0 means that no size has been entered, and Prograph will use the system default font size for the field’s font size.

Bold, Italic, Underline, Outline, Shadow¹⁴⁷

Click the associated check box to assign one or more of the available styles.

Left, Center, Right¹⁴⁷

Click one of the associated radio buttons to set the field’s alignment. These attributes are grayed out and are not available if you are creating a Scroll Text field.

Active¹⁴⁷

If a text item is active, its text can be edited at runtime, including the default text previously entered in the Text field. If the text item is inactive, text cannot be entered and the default text can not be edited unless you return to this Text editor dialog.

Visible¹⁴⁷

If you uncheck this attribute, the text item and any default text will not be visible. The default is visible (checked).

Border¹⁴⁸

Check if you want a 1 point border drawn around the text item. Uncheck if the text item is to be used as a title or other static text.

Move w/Window¹⁴⁸

When you check this attribute, the text item remains the same size but the top/left corner of the text item moves in relation to the size of the window. You would check this attribute when building a program that will run on several different size monitors and you have set the window to be the same size as the monitor.

Grow w/Window¹⁴⁸

Check this attribute when you want the text item size to change when you change the window size. If both Move w/Window and Grow w/Window are checked, the text item will change proportionately to the size of the window.

Wrap¹⁴⁸

Check this attribute when you want the text to wrap within the confines of the defined text item. Checking this option in the Scroll Text editor window removes the horizontal scroll bar.

When editing a visible instance of Text, the string displayed is the value of the Text attribute of the «text» using the font, point size, and style specified by the attributes font, font size, and font style. The string is justified in the bounding rectangle of «text» according to the value of the attribute justification, and is wrapped to fit into the bounding rectangle. The bounding rectangle is drawn in gray.¹⁴⁸

When the «Window» is active, the «Text» also has this appearance, except that the bounding rectangle is not drawn.

Edit Text¹⁴⁸

The Edit Text Specification editor is identical to the Text Specification editor, except that the check boxes Active and Wrap are available. These boxes are checked if and only if the attributes active? and wrap? respectively of «Edit Text» are TRUE.

When editing a visible instance of Edit Text, the string displayed is the value of the text attribute of the «Edit Text» in the same way as for an instance of Text, except that the bounding rectangle is drawn black and the string is wrapped to fit the bounding rectangle only if the wrap? attribute has value TRUE. For example:

When the «Window» is active, the «Edit Text» also has this appearance, but if the «Edit Text» is active, it can also be edited.¹⁴⁹

Scroll Text¹⁴⁹

The Scroll Text Specification editor is identical to the Edit Text Specification editor.

When editing a visible instance of Scroll Text, the string displayed is the value of the text attribute of the «Edit Text» in the same way as for an instance of Edit Text, except that the bounding rectangle has both

vertical and horizontal scroll bars. For example:

When the «Window» is active, the «Scroll Text» also has this appearance, but if the «Scroll Text» is active, it can be edited.

Styled Text in Prograph Classic

All Edit Text and Scroll Text window items are created with “style-aware” edit records. This means your applications can include the ability to change the font, style or size of individual pieces of text within an item. Mac toolbox calls in your code must be used to accomplish this functionality, but there are “canned” Font, Style and Size menus in an example file on the distribution disk, so you can just “plug them in and go.”

You set the default font, size and style with the dialog in the Application Builder. If you don’t put any style changing code in your application, Edit Text and Scroll Text items will have text with font, size and style set to those defaults. In this case, setting the font, font size, or font style attributes at runtime will change settings for all the text in the item.

However, if multiple fonts, styles or sizes exist in the text, setting those attributes will have the following effect: only runs of text that have the old value of the attribute will be changed to use the new value. For example, if an item has a font size attribute with value 10, and there is 10, 12 and 14 point text in the item, setting the font size attribute to 18 will affect only the 10 point text—the 12 and 14 point text will be unchanged.

A new attribute, style record, has been added to the Edit Text class and is inherited by Scroll Text. Prograph uses this attribute to automatically save information about style changes within the item’s text. The attribute is set to NULL if one font, size and style is used throughout.

Setting the text attribute will cause the saved style record information to be set to NULL, and the default settings from the font, font size, and font style attributes will be applied to all of the new text.

Scroll List

The Scroll List Specification editor is analogous to the Click Item Specification editor, but has an extra editable-text field, List of Values, which displays the list of strings that is the value of the attribute value list. It also supports your choice of font, size, and style for the scroll list items, and lets you specify whether the user is restricted to selecting just one item from the scroll list, or whether multiple selection is allowed.

¹⁵⁰ When editing a visible instance of Scroll List, the Scroll List image displays the list of strings that is the value of the value list attribute of the «Scroll List». The strings are displayed in a rectangle from top to bottom in the order in which they occur in value list. If the «Scroll List» is active, the rectangle has a vertical scroll bar. For example: ¹⁵¹

¹⁵¹ When the «Window» is active, the «Scroll List» also has this appearance, but some of the displayed strings can also be selected.

Canvas ¹⁵¹

The Canvas Specification editor displays the following items.

Item Name, Click Method, Draw Method^{151*}

Editable-text fields that display the values of the attributes name, click method, and draw method, respectively of «Canvas».

Item Limits^{151*}

Four editable-text fields, Top, Left, Bottom, and Right, which together display the value of the attribute limits of the «Canvas». The limits value is a rectangle defining the area over which the top left corner of the canvas can range, and thus, together with the size attribute of the «Canvas», defines the size of the drawing area onto which the «Canvas» provides a view.

Vertical Scroll Bar, Horizontal Scroll Bar^{151*}

These boxes are checked if the attributes vScroll? and hScroll?, respectively, of «Canvas» are TRUE.

Border^{152*}

If checked, a 1 point border is drawn around the edges of the canvas.

Active, Visible, Move w/Window, Grow w/Window^{152*}

These boxes are checked if and only if the attributes active?, visible?, move?, and grow?, respectively, of «Canvas» are set to TRUE.

Balloon...^{152*}

Button for opening Balloon Help dialog.

When editing a visible instance of Canvas, the Canvas image appears as follows, although the scroll bars are drawn as active if the «Canvas» drawing rectangle is smaller than the drawing area as defined in the description of Item Limits above.^{152*}

When the «Window» is active, the «Canvas» also has this appearance, except that the «Canvas» rectangle can also contain graphics generated by the method specified in the attribute draw method.^{152*}

Pop-up Menu^{153*}

Value List is a list of strings for the pop-up's menu item names. If, for example, you enter a list of numbers, Prograph will beep, and you will not be able to hit OK until you have changed the list to a list of strings.

If you want a title for your pop-up menu, leave the default setting checked. However, if you want just the pop-up menu, without a separate title, displayed, then uncheck the Title? option.¹⁵³

If the Fixed Size option is checked, the pop-up menu always extends to the right edge of the pop-up menu item. If the contents of the menu do not fit in the provided space, the text is truncated to fit. If the Fixed Size option is unchecked, the pop-up menu will automatically adjust its size to fit the longest item in the menu. The default setting is "unchecked" for this option.

The font attributes only apply to the menu's title.

The Click Method is the method that is executed when selecting a menu item.¹⁵³