V2. Editing

All ViewIt windows can be edited from within a running program by pressing Option-âŒ˜-Shift to enter ViewIt's editing mode. If one or more floating windows are present when entering editing mode (i.e., more than one window is "active"), then either the window beneath the cursor, the window with the selected control, the active window, or the top floating window is the one that gets edited (in that order).

This topic describes how ViewIt windows are organized into "views" and "controls", the corresponding resource types, and ViewIt's support for on-line editing of these resources.

Windows/Views/Controls

A ViewIt window can contain any number of "views", and any number of "controls" within each of these views. This produces a hierarchy of window, view, and control objects within programs that open ViewIt windows:

```
 Window #1
  View #1
   Control #1
   Control #2
   ...
  View #2
   Control #1
   Control #2
   ...
  ...
 Window #2
  View #1
   Control #1
   ...
  ...
 ...
```

This hierarchy defines the order in which controls and views are drawn, and the numbering scheme that we make use of to identify controls. A control's view number is the number of the view containing the control, and its control number is the number of the control in that view. Thus the control containing this help text can be referred to as "control 14 in view 1" or simply "v1c14".

Technically, a ViewIt "control" is a typical Mac control item (i.e., check box, radio button, etc.) that belongs to a particular view, and a ViewIt "view" is a special type of rectangular control that contains other controls. Since a view is just a special type of control, we sometimes use the term "controls" when referring to both the controls and views in a window.

ViewIt controls can be as simple as standard buttons and check boxes, or as complex as fully-functional text editors which contain their own private controls and other parts (such as the help control displaying this text). FaceWare distributes several advanced controls that can be purchased separately (see our flier or the vDemoXY program for more info).

Editing Resources

Windows, views, and controls have a 1:1 correspondence with special resource types supported by ViewIt: FWND, FVEW, and FCTL resources. One FWND is used to open and save a ViewIt window, one FVEW is created when copying a view, and one FCTL is created when copying a control. Thus ViewIt's edit mode can be thought of as an FWND, FVEW, and FCTL resource editor.

Other resource editing not supported by ViewIt can be done with ResEdit or other resource editor. If FaceIt and System ≥ 7.0 are in use, then ResEdit or Resorcerer can even be accessed on-line via a menu within ViewIt's Control dialog (i.e., you can edit a resource without quitting the program).

All resources edited or added by ViewIt when in editing mode affect the "default" resource file, which is the file containing the LoadIt module (typically your ".Rsrc" file). If FaceIt and System ≥ 7.0 are in use, then this file can be quickly passed to ResEdit or Resorcerer for editing via the ResEdit menu icon at the bottom of the main ViewIt help window.

REMINDER: As discussed in the "Startup" notes, remember to use resource ID numbers < 1100 or > 7499 for new program resources since the range of ID numbers from 1100-7499 is reserved for FaceWare modules.

Selecting Controls

Most editing operations first require the selection of a control or view control in the window being edited. Typical operations such as clicking and shift-clicking to select, deselect, or extend selections work largely as expected. One important difference you'll find is that a view and a control cannot be selected at the same time, and that the movement, deletion, copying, or duplication of a view also affects all of the controls within it.

To help with the selection of multiple controls in the same view, ViewIt supports SHIFT-DRAGGING across controls. As the mouse is dragged, a lasso cursor appears and outlines the region in which controls are to be selected.

The TAB key can be used to select the next visible control in the control list, and SHIFT-TAB to select the previous visible control.

Finally, a quick way to deselect all controls is to click in the drag area of the bar at the bottom of the window. This is useful, for example, when a view fills the entire window and there is no place to click without selecting something.

The Controls Bar

A ViewIt window with many views and controls would be confusing to work with if there were no way to quickly find, select, and reorder controls. This functionality is provided by the "controls bar" at the right of a ViewIt window being edited. This bar shows a "pile" of controls. The controls shown are either the view controls (if a view or no control is selected), or the controls within the view which contains the currently selected control. The horizontal arrows at the bottom of the bar allow you to quickly flip through all of the views and controls within the window.

Controls can be selected in the bar using the same methods supported in the content area of the window (including shift-drag). The bar can also be used to reorder controls in the window by simply dragging one or more controls in the bar up or down to new locations.

Another important use for the bar is in finding and showing controls or views that are hidden. If working with views that are hidden and shown in succession to support "paging" of groups of controls, for example, then the "Hide" menu item (discussed below) can be used to hide one view and the controls bar to show the next view to be edited. Shortcut: If the OPTION key is pressed when clicking on a hidden control or view in the bar, then any selected controls or views are first hidden before showing the clicked control or view.

The Icon Menu Bar

The icon bar displayed at the bottom of a window being edited has the sort of functionality expected from a typical dialog or window resource editor. It also contains a drag region (to the right of the icons) which can be used to drag any window to a new location (even if it doesn't have a title bar). The drag region also serves as a status bar, showing information about either the control that the cursor is currently above, or the currently selected control if the cursor is located outside the content area of the window.

• File Menu (file icon)
Glossary Glossary of commonly used terms.
ViewIt Help Opens main ViewIt on-line help window.
Driver Help Opens the control driver's help window.
Save All to disk Saves updated FWND to disk. All changes made to a window are temporarily stored in the FWND in memory until this item is chosen to force the FWND to disk.
Revert from disk Replaces the contents of the window and the FWND in memory with the FWND last saved to disk.
Edit Another Edits another FWND resource. This makes it possible for you to edit any other window, or even create completely new windows, from within ViewIt's edit mode. The new windows do not interact with program code, but can be tested as modal windows with the "Test as Modal" option.
Edit Options Various editing options saved in FWND.
Test as Modal Leaves edit mode but keeps window open as a modal window for testing. This option is available for windows opened from "Edit Another" or from the window icon in the main ViewIt Help window. To close a window being tested, reenter edit mode and choose "Quit Edit Mode".
Test w/o Colors Attempts to display the window as it will look on a 1-bit deep, black-and-white screen (or

when the screen depth is less than the "Bits/Pixel Minimum" value set in the Window dialog).
Quit Edit Mode   Shuts down ViewIt's editing mode.   You can also hit the ENTER or RETURN keys to do this.   The FWND in memory associated with the window is updated to reflect the current state of the window so that your changes will not be lost if the window is closed and later reopened.   The FWND on disk, however, is not updated unless "Save All to disk" is chosen (or the "Always Save All..." option is checked).

• Edit Menu   (scissors icon)
Undo Changes   Reverts window to state it was in when edit mode was entered.   This is accomplished by using the FWND in memory (rather than the disk copy used by "Revert").
Cut   Copies and clears the current selection.
Copy   Copies the current selection as an FCTL, FVEW, or FWND resource.   Note that multiple controls or views can be copied, respectively, to a single FCTL or FVEW resource.
Copy Colors   Copies the color table associated with the current selection.   The copied colors can then be applied to any other view or control (see "Apply" below).
Paste   Pastes a resource into the window.   FWNDs replace the entire window contents.   FVEWs and DITLs add a new view to the window (ViewIt also converts the DITL's items to controls and adds them to the view).   FCTLs and CNTLs add a new control to the top or currently selected view.   Views and controls are inserted after the current selection, giving you control over where the pasted items get placed in the control list.
Apply   If the current clipboard contains an FVEW or FCTL resource, then size or style settings are extracted from that resource and used to reset the size or style of the currently selected view(s) or control(s).   If the current clipboard does not contain an FVEW or FCTL resource, then the colors last copied with "Copy Colors" are used to reset the colors of the currently selected view(s) or control(s).
Clear   Deletes the selected control(s) from the window.   The Delete key can also be used to clear items.
Select All   If a view control or no control is selected, then all views get selected.   If a control within a view is selected, then all controls in that view get selected.
Duplicate   Creates a copy of the selected control(s) and inserts the new controls just after their parents in the control list.   COMMAND-DRAGGING can also be used to clone controls. The latter has the advantage of giving you control over where the new controls get placed in the window.

• Import Menu   ("+" icon)
Auto-Add Res   If not checked, then any resources linked to an imported control (PICT, ICON, etc.) will not be copied, and the control will continue to use the existing example resources in the FaceWare file.   This may lead to missing resources when creating stand-alone applications since the non-editing version of the FaceWare file does not contain these example resources.   If checked, then ViewIt makes a copy of any linked resources and adds these to the program's resource file when importing a control.   The added resources are given an ID 20,000 greater than that found in the FaceWare file (i.e., "1311" in FaceWare file becomes "21311" in program file).   This option is the same as the "Auto-Add" option in the "Edit Options" dialog.
View   List of named FVEWs available in all open resource files which can be imported as views.   Note that the FVEWs appearing in this menu are just examples of typical view types.   See the "Views" topic for a complete discussion of view options.
Dial   List of available dial-type FCTLs.
List   List of available list-type FCTLs.
Menu   List of available menu-type FCTLs.
Rad Btn   List of available radio button-type FCTLs.
Chk Box   List of available check box-type FCTLs.
Button   List of available button-type FCTLs.
Editable   List of available editable-type FCTLs.
Static   List of available static-type FCTLs.
Other   All other FCTLs available in all open resource files.
Note that the FCTLs appearing in these menus are just examples of typical control configurations.   See the "Controls" topic for a discussion of control options.

Control drivers are typically shipped with default FCTLs that will automatically appear in the Import menu after installing the driver.   To add additional view or control items to the Import menus, simply copy

a view or control as an FVEW or FCTL resource when in edit mode, use a resource editor to paste this resource into a resource file used by your program, and then give the new resource a name so that it will appear in the corresponding Import menu. FCTL resources will appear in the "Other" menu unless their names are preceded by one of the following lower-case letters:
"d" for dial-type controls
"l" for list-type controls
"m" for menu-type controls
"r" for radio button-type controls
"c" for check box-type controls
"b" for button-type controls
"e" for editable-type controls
"s" for static-type controls
where this leading character is removed before adding the name to the corresponding menu.

• Info Menu (window icon)
Window Opens the "Window" dialog for resetting basic parameters that define the window's behavior and type.
Shortcut: Double-click in drag area at bottom of window.
Control Opens the "Control" dialog for resetting basic parameters that define the control's behavior and type.
Shortcuts: Triple-click the body or frame of the control, or double-click on a rounded rectangle in the controls bar.
Bounds Opens the "Bounds" dialog for resetting variables used to define the selected control's bounds and frame.
Shortcut: Double-click on control frame when selected.
Title Opens the "Title" dialog for resetting the title, style, color, item ID, and command key of the selected control.
Shortcut: Double-click on control body when selected.
Links Opens the "Data Links & Item IDs" dialog which displays a complete list of all data links and item IDs.
Lock Window Type Prevents opening of the "Window" dialog for resetting basic window parameters.
Lock Control Order Prevents the insertion, deletion, and reordering of controls in the window.
Lock Control Type Prevents opening of the "Control" dialog for resetting basic control parameters.
Lock Control Position Prevents dragging of the control. You might use this, for example, to lock the position of a view so that it is not accidentally dragged during editing.
Hide Hides the selected control(s). The controls bar at the right can be used to find and show any hidden controls. The combination of the "Hide" item and the controls bar to hide and show controls, for example, is often used when editing multiple, overlapping views in the same window.
Activate/Inactivate Activates or inactivates the selected control(s). Inactive controls are greyed and do not respond to events in the window.

• Style Menu ("A" icon)
Font, Size, Style Changes the control's text font, size, or style. The exact effect depends on what the control driver does with this text style information.
Justify Some controls support setting the justification of their contents. The "System" option refers to the System's default justification which depends on the System script.
Colors With System 7.0, Apple introduced 15 standard control part colors seen in this menu. Many ViewIt controls, however, only support coloring the "Frame", "Body" (background), and "Content/Text" parts. Choosing one of the parts in this menu will add a color to the control's color table, as indicated by the part name becoming underlined. Choosing the "System" option resets the control's colors to Apple's default colors by removing all entries from the control's private color table. For more info about colors, see "Colors" under the "Controls" topic.
NOTE: The standard CDEF controls from Apple have a "feature" of reusing the last defined part color for all other parts that are not assigned colors. Thus setting just the "Frame" to red will cause the entire control to be drawn in red unless you take the trouble to also define other part colors.
Solid Body If checked, enables drawing of the control's body (background), otherwise the control will be

"transparent". Some controls ignore this flag and always draw solid bodies.

- Align Menu

Center Horz...Switch Position Realigns a view within the window or a control within its view.
Bring To Front...Send Behind Reorders control(s) in the window's control list. The affected controls are redrawn and the controls bar is updated to show the new order.

- Grid Icon

The last icon in the icon bar shows the current "grid" size (1, 2, 4, or 8 pixels). Clicking this icon flips the grid size to its next value. Dragging of controls is then restricted to window locations which are multiples of the grid size.

Outline Mode

Pressing CAPS LOCK while editing a ViewIt window puts the window into "outline mode" . This mode displays all controls as simple, black-and-white rectangles with basic control information shown in each rectangle: a small icon indicating the control type, the control number, and the control title. This mode provides a quick way to view the position, order, and type of all controls in a window.

Saving Changes

If the "Always Save All..." option in "Edit Options" dialog is not checked, then changes made to ViewIt windows are not saved to disk until the File item "Save All to disk" is chosen. Thus, if the "Always Save All..." option is not checked, you must be careful to use "Save All to disk" when you wish to make changes permanent.

Also note that changes made to resources loaded from the FaceWare file cannot be saved since the FaceWare file is always opened as "read only". To make changes to FaceWare resources, simply move a copy of the resources to be edited from the FaceWare file to your program file or other res file used by your program. The resources can be moved using a resource editor or our MoveIt FCMD mover.

Getting Into Trouble

Code that is used to manage a ViewIt window often makes assumptions about the position and type of controls in the window. When editing such windows be careful to leave the window in a state that doesn't conflict with the code. This usually means that you must be careful not to remove or reorder certain controls in the window, but can also apply to simply hiding and showing controls or views if the code assumes that certain controls or views are visible.

One tricky aspect of ViewIt editing to be conscious of is the fact that ViewIt always updates the FWND in memory when leaving editing mode. This FWND in memory will then be used to open the next instance of that window, which might conflict with code if you happen to have left editing mode when the window contents were not arranged correctly.

Suppose, for example, that you wished to copy the help view from one of ViewIt's built-in dialogs. You would open the dialog, switch to the help view, enter edit mode, copy the help view, leave edit mode (which updates the FWND!), switch out of the help view, close the dialog, and paste the view into your window. Everything appears fine until the dialog is opened again and you find that its help is displayed first! The FWND in memory is now "out of synch" with the code in ViewIt.

One way to prevent problems like that just described is to write defensive code that always checks the state of controls and views when a window is opened. This is usually a waste of time, however, since such problems do not affect users of your programs, and are easily fixed with a little window editing (hide the view that should be hidden, show the view that should be shown, etc.). In the case of the Control dialog, for example, we always close the help view after editing it, and then enter and leave editing mode one more time so that the FWND in memory isn't left "out of synch".