

## Control Settings

This dialog resets control parameters that affect control type, behavior, and data linking. Changes made here are saved in the FWND in memory. Use "Save All to disk" to make changes permanent.

## Page 1 - Type & Behavior

**Driver** - Each control in a ViewIt window is supported by a code resource that is either a CDEF or an FCMD "control driver". CDEFs are denoted by BaseID = CDEF ID and VersID = 0. Any CDEF based control can be used in a ViewIt window, but the most common ones used are Apple's standard button, check box, radio button, and scroll bar.

FCMD control drivers are denoted by BaseID = FCMD ID and VersID = 10 x the driver's version number. Drivers shipped with ViewIt include:

Name	BaseID	VersID	Use
BaseCt	1310	22	basic controls
PeekCt	1390	22	debugging
ScrapCt	1800	22	scrap display
HelpCt	1900	22	on-line help

All drivers are shipped with example controls that appear in ViewIt's import menu, so it is easy to experiment by adding these to any window.

Control drivers support more powerful controls than CDEFs by taking advantage of the extended control record and message scheme implemented by ViewIt. These extensions are documented in the "Inside Drivers" document within the "Program Controls" folder included with the ViewIt product.

To learn more about the operation of a driver, enter edit mode, select a control, open the Control dialog (the one you are now in), and press the "Driver Help" button below VarCode to open the driver's help. For BaseCt controls, the window opened is titled "About BaseCt" and describes all of the control types supported by the BaseCt driver.

**Behavior** - One of the nifty features of ViewIt controls is that control behavior can often be set independently of the control's appearance. This means, for example, that an ICON-based control supported by BaseCt can act as either a button, check box, radio button, or static item, depending upon which "class" of behavior is chosen. Thus you should always examine the driver's on-line help (via the "Driver Help" button) to determine the full capabilities of controls.

A short description of each class is presented to help clarify its use:

- Static - displays text, icon, picture, etc., but does not become hilited or checked when clicked
- Button - becomes hilited when clicked, but is never checked. Most button-type controls also have the "Return On Hit" option checked so that the main program is notified when they are hit.
- Check Box - becomes hilited when clicked, and toggles between a checked and unchecked state. The checked state often corresponds to the control value = max value, and unchecked = min value.
- Radio Button - similar to Check Box, except that adjacent radio buttons are mutually exclusive with respect to their being checked. This feature is actually supported by the ViewBV view driver which manages the checking/unchecking of radio buttons and check boxes after clicks in controls.
- Dial Control - a control which contains an "indicator" that can be dragged to change the value of the control.
- List or Menu - a control which displays a list of items, one or more of which can be selected. The item(s) selected usually have a relationship to the control's current value.
- Editable - manages multiple, discrete items within the content area of the control which can usually be edited (such as the characters in a text editor). Only one editable control in a window can be "selected" at a time (just as only one window can be active). The currently selected, editable control receives all key and menu events.

**Messages** - Optional messages that ViewIt can send to control drivers. In most cases the example controls included with drivers will have these options preset correctly. See the driver's on-line help for more info. Briefly,

- Idle - gets idle time when selected
- Hook - called whenever no events pending
- Scrap - notified if clipboard (scrap) changes

- Heap - called when heap space needed
- Save - given chance to save contents
- Quit - disposed of before program quits
- Active - notified if window is act/deactivated

Keys - Determines whether the keys listed are sent to editable controls. In most cases the example controls included with drivers have these options preset correctly. See the driver's on-line help for more info.

Returns On Hit - If the control is not editable, then this option causes an "item hit" message to be returned to the main program after a click occurs in the control. If the control is editable, then this option causes a message to be returned whenever the control becomes selected or deselected. See the "Item Events" and "Editable Items" subtopics in the "Windows" topic of the ViewIt Guide for more information about the messages returned.

Is Default Item - Makes the control the default item in the window (the item "hit" when the Enter or Return key is pressed). The "Clear Others" button clears this flag for all other controls in the window. Note that the drawing of a ring around a default button is a separate option (set in Bounds dialog), that default items need not be standard buttons (the box at the top, left of this window is a default item), and that it sometimes makes sense to have more than one default item in a window (as long as only one of these is active & visible).

NOTE: Don't confuse the default item hit when Return or Enter is pressed with the default item checked in a set of radio buttons. The latter is the responsibility of the program and is not related to the "Is Default Item" option.

Supports FSSC - Informs ViewIt that it should enable its Font, Size, Style, and Color (FSSC) menus when an editable control is selected and the user attempts to choose a menu item. In most cases the example controls included with drivers have this option preset correctly. See the driver's on-line help for more info. Note that this option is not related to choosing a control's overall Font, Size, Style, or Color when in edit mode, but rather to choosing styles within a control's content when not in edit mode (like the text you are reading).

No Autscroll (views only) - Informs ViewIt that it should not do any autoscrolling of editable controls within this view. If this option is not set, then ViewIt always attempts to bring an editable control into view when it becomes selected (i.e., when tabbing from one editable control to another within a scrollable view). This option is useful if you don't like the way that ViewIt autoscrolls and wish to handle such scrolling yourself.

Global Hilite - Requests that the control make use of the System's hilite color instead of doing a simple inversion. Many control types ignore this option.

Res Link - Optional link between the control and a resource. The basic control driver shipped with ViewIt, BaseCt, makes extensive use of this option to support many different types of controls. Some control drivers, and all standard CDEFs, however, ignore the resource Type and ID fields. Always examine the driver's on-line help to determine whether and how the Type and ID fields are used.

The ResEdit icon menu can be used to add/edit the resource specified by the Type and ID fields:

- Copy - Copies linked resource to clipboard.
- Paste - Replaces the linked resource with the resource on the clipboard (types must match).
- Add - If the resource does not exist, adds a new resource of the specified Type and ID to the program's default resource file. If already linked to a resource of the same type (i.e., just the ID is being changed), then ViewIt creates a copy of the linked resource (useful when you want to clone a resource that belongs to a FaceWare module).
- Edit - Creates a copy of the resource in a disk file of the same name and then passes this file to ResEdit or Resorcerer for on-line editing of the resource. If the resource is changed, then the copy of it in your program file is updated on return from the resource editor. (This feature requires Facelt, System  $\geq 7.0$ , and SIZE resource like that discussed in "Finder Resources".)

ViewIt also recognizes BaseCt controls that use multiple resources of the same type. To link a set of 4 ICON resources numbered 1011-1014 to such a control, for example, set Type = ICON, ID = 1011, Min = 1011, and Max = 1014. "Add Resource" will then add four ICON resources to the default resource file, and "Edit" can then be used to edit all four ICONs at once.

Values - The standard control values associated with all Macintosh controls (Min = "ctrlMin", Max = "ctrlMax", and Value = "ctrlValue"). Many drivers make use of these values, but others do not, so always examine the driver's on-line help to determine how these values are used. Also note that ViewIt supports a powerful data linking scheme that does not restrict control "values" to integers (see "Data Links" in the ViewIt Guide).

VarCode - A 16-bit (2-byte) field that can be used by drivers and CDEFs to store additional, control-specific information. Most drivers use the VarCode as a set of bit flags that turn on or off various control features. These flags will be documented in the driver's on-line help.

ID/RefCon - A 4-byte integer that is reserved for use by the main program (although custom CDEFs have also been known to make use of this field). If the absolute value of the ID is less than 32768, then its value is returned in wiHit when an enabled item is hit, and in ciIndex after a call to GetCtl to get control info.

## Page 2 - Instructions & Links

Instructions - An optional string used by some control drivers. View controls supported by the ViewBV driver, for example, use this string as an instruction to activate/inactivate or show/hide daughter controls (see "Views" topic in the ViewIt Guide).

Data Link - Establishes link between a program variable and the control value. See "Data Links" topic in the ViewIt Guide for more info. (Note that a value of -1 for "Byte Offset" indicates that the main program used LnkCtl to link this control to an isolated program variable.)

Override ID - Optional number that identifies an override procedure to use from a table of such procedures created by the main program. See the "Override" topic in the ViewIt Guide for more info on override procedures.