Control Background Color

The background color of control panel windows and the area at the top of a graph as reserved by the **ControlBar** operation (page V-88) is a shade of gray chosen to match the operating system look. This gray is used when the control bar background color, as set by ModifyGraph cbRGB or ModifyPanel cbRGB, is the default pure white, where the red, green and blue components are all 65535. Any other cbRGB setting, including not quite pure white, is honored. However, some controls or portions of controls are drawn by the operating system and may look out of place if you choose a different background color.

For special purposes, you can specify a background color for an individual control using the labelBack keyword. See the reference help of the individual control types for details.

Control Structures

Control action procedures can take one of two forms: structure-based or an old form that is not recommended. This section assumes that you are using the structure-based form.

The action procedure for a control uses a predefined, built-in structure as a parameter to the function. The procedure has this format:

```
Function ActionProcName(s)
   STRUCT <WMControlTypeActio>& s   // <WMControlTypeActio> is one of the
   …                                // structures listed below
End
```

The names of the various control structures are:

Control Type	Structure Name
Button	**WMButtonAction**
CheckBox	**WMCheckboxAction**
CustomControl	**WMCustomControlAction**
ListBox	**WMListboxAction**
PopupMenu	**WMPopupAction**
SetVariable	**WMSetVariableAction**
Slider	**WMSliderAction**
TabControl	**WMTabControlAction**

Action functions should respond only to documented `eventCode` values. Other event codes may be added along with more fields in the future. Although the return value is not currently used, action functions should always return zero.

The constants used to specify the size of structure `char` arrays are internal to Igor Pro and may change.

You can use the same action procedure for different controls of the same type, for all the buttons in one window, for example. Use the ctrlName field of the structure to identify the control and the win field to identify the window containing the control.

Control Structure Example

This example illustrates the extended event codes available for a button control. The function prints various text messages to the history area of the command window, depending what actions you take while in the button area.