```
    String curTabMatch= "*_tab"+num2istr(tabNum)
    String controlsInCurTab= ListMatch(controlsInATab, curTabMatch)
    String controlsInOtherTabs=ListMatch(controlsInATab,"!"+curTabMatch)

    ModifyControlList controlsInOtherTabs disable=1          // hide
    ModifyControlList controlsInCurTab disable=0             // show

    return 0
End

// Panel macro that creates a TabControl using TabProc2():
Window TabbedPanel2() : Panel
    PauseUpdate; Silent 1                    // building window…
    NewPanel /W=(35,208,266,374) as "Tab Demo"
    TabControl tab,pos={12,9},size={205,140},proc=TabProc2
    TabControl tab,tabLabel(0)="Tab 0"
    TabControl tab,tabLabel(1)="Tab 1",value= 0
    Button button_tab0,pos={26,43},size={110,20},title="Button in Tab0"
    Button button2_tab0,pos={26,74},size={110,20},title="Button in Tab0"
    Button button3_tab0,pos={26,106},size={110,20},title="Button in Tab0"
    Button button_tab1,pos={85,43},size={110,20},title="Button in Tab1"
    Button button2_tab1,pos={85,75},size={110,20},title="Button in Tab1"
    Button button3_tab1,pos={84,108},size={110,20},title="Button in Tab1"
    ModifyControlList ControlNameList("",";","*_tab1") disable=1
EndMacro
```

Run TabbedPanel2 and then click on "Tab 0" and "Tab 1" to run TabProc2.

See Also

See Chapter III-14, **Controls and Control Panels** for details about control panels and controls.

Related functions **ModifyControl** and **ControlNameList**.

The **Button**, **Chart**, **CheckBox**, **GroupBox**, **ListBox**, **PopupMenu**, **SetVariable**, **Slider**, **TabControl**, **TitleBox**, and **ValDisplay** controls.

ModifyFreeAxis

ModifyFreeAxis [**/W=*winName***] ***axisName*, master=*mastName***
    [, **hook=*funcName***]

The ModifyFreeAxis operation designates the free axis (created with **NewFreeAxis**) to follow a controlling axis from which it gets axis range and units information. The free axis updates whenever the controlling axis changes. The axis limits and units can be modified by a user hook function.

Parameters

axisName is the name of the free axis (which must have been created by **NewFreeAxis**).

masterName is the name of the master axis controlling *axisName*.

funcName is the name of the user function that modifies the limits and units properties of the axis. If *funcName* is `$""`, the named hook function is removed.

Flags

/W=*winName*	Modifies *axisName* in the named graph window or subwindow. If /W is omitted the command affects the top graph window or subwindow. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

Details

The free axis can also be designated to call a user-defined hook function that can modify limits and units properties of the axis. The hook function must be of the following form:

```
Function MyAxisHook(info)
    STRUCT WMAxisHookStruct &info

    <code to modify graph units or limits>
    return 0
End
```

where WMAxisHookStruct is a built-in structure with the following members:

WMAxisHookStruct Structure Members

Member	Description
char win[MAX_WIN_PATH+1]	Host (sub)window
char axName[MAX_OBJ_NAME+1]	Name of the axis
char mastName[MAX_OBJ_NAME+1]	Name of controlling axis or ""
char units[MAX_UNITS+1]	Axis units
double min, max	Axis range minimum and maximum values

The constants used to size the char arrays are internal to Igor and are subject to change in future versions.

The hook function is called when refreshing axis range information (generally early in the update of a graph). Your hook must never kill a graph or an axis.

Example

This example demonstrates how to program a free axis hook function, whose most important task is to change the values of info.min and info.max to alter the axis range of the free axis. The example free axis displays Fahrenheit values for data in Celsius.

```
Function CentigradeAndFahrenheit()
   Make/O/N=20 temperatures = -2+p/3+gnoise(0.5)  // sample data
   Display temperatures// default left axis will indicate data's centigrade range
   String graphName = S_name
   Label/W=$graphName left "°C"
   ModifyGraph/W=$graphName zero(left)=1
   Legend/W=$graphName

   // make a right axis whose range will be Fahrenheit
   NewFreeAxis/R/O/W=$graphName fahrenheit
   ModifyGraph/W=$graphName freePos(fahrenheit)={0,kwFraction},lblPos(fahrenheit)=43
   Label/W=$graphName fahrenheit "°F"

   ModifyFreeAxis/W=$graphName fahrenheit, master=left, hook=CtoF_FreeAxisHook
   // NOTE master=left part which makes the "free" axis
   // actually a "slave" to the left ("master") axis.
End

Function CtoF_FreeAxisHook(info)
   STRUCT WMAxisHookStruct &info

   GetAxis/Q/W=$info.win $info.mastName    // get master axis range in V_min, V_Max
   Variable minF = V_min*9/5+32
   Variable maxF = V_max*9/5+32

//  SetAxis/W=$info.win $info.axName, minF, maxF
//  SetAxis here is fruitless. These values get overwritten by Igor
//  after reading info.min and info.max, which we now set:
   info.min = minF           // new min for free axis
   info.max= maxF            // new max for free axis
   return 0
End
```

See Also

The **SetAxis**, **KillFreeAxis**, and **NewFreeAxis** operations.

The **ModifyGraph (axes)** operation for changing other aspects of a free axis.