update=*u*	Sets the type of updating of contour traces when the data or contour settings change. *u*=0: Turns off dynamic updates, which might be advisable if updates take a long time. *u*=1: Updates the contours only once, or until you next execute an update=1 command. *u*=2: Updates are automatic (default). *u*=3 Marks the contour plot as having been updated once (*u*=1) already. This option is used in recreation macros to prevent an extra redraw of a graph saved with *u*=1 update mode in effect. If you use it in a command, the result is similar to *u*=0, but the Modify Contour Appearance dialog will automatically select "update once, now" from the Update Contours pop-up menu.
xymarkers=*x*	Controls the visibility of XY markers. *x*=0: Hides markers showing XY coordinates of the Z data (default). *x*=1: Displays markers showing XY coordinates of Z data. Initially, this uses marker number zero. You can change this using the Modify Trace Appearance dialog.

Flags

/W=*winName*	Applies to contours in the named graph window or subwindow. When omitted, action will affect the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

See Also

AppendMatrixContour and **AppendXYZContour**.

References

Watson, David F., *nngridr - An Implementation of Natural Neighbor Interpolation*, Dave Watson Publisher, Claremont, Australia, 1994.

ModifyControl

`ModifyControl [/Z] ctrlName [keyword = value [, keyword = value …]]`

The ModifyControl operation modifies the named control. ModifyControl works on any kind of existing control. To modify multiple controls, use **ModifyControlList**.

Parameters

ctrlName specifies the name of the control to be created or changed. The control must exist.

Keywords

The following keyword=value parameters are supported:

activate	align	appearance	bodywidth	disable	fColor
focusRing	font	fSize	fStyle	help	labelBack
noproc	pos	proc	rename	size	title
userdata	valueBackColor	valueColor	win		

Coordinates are in **Control Panel Units**.

For details on these keywords, see the documentation for **SetVariable** on page V-854.

The following keywords are not supported:

mode	popmatch	popvalue	value	variable

Flags

/Z	No error reporting.

Details

Use ModifyControl to move, hide, disable, or change the appearance of a control without regard to its kind

Example

Here is a **TabControl** procedure that shows and hides all controls in the tabs appropriately, without knowing what kind of controls they are.

The "trick" here is that all controls that are to be shown within particular tab *n* have been assigned names that end with "_tab*n*" such as "_tab0" and "_tab1":

```
Function TabProc(ctrlName,tabNum) : TabControl
   String ctrlName
   Variable tabNum

   String curTabMatch= "*_tab"+num2istr(tabNum)

   String controls= ControlNameList("")
   Variable i, n= ItemsInList(controls)
   for(i=0; i<n; i+=1)
      String control= StringFromList(i, controls)
      Variable isInATab= stringmatch(control,"*_tab*")
      if( isInATab )
         Variable show= stringmatch(control,curTabMatch)
         ControlInfo $control                    // gets V_disable
         if( show )
            V_disable= V_disable & ~0x1          // clear the hide bit
         else
            V_disable= V_disable | 0x1           // set the hide bit
         endif
         ModifyControl $control disable=V_disable
      endif
   endfor
   return 0
End

// Action procedures which enable or disable the buttons
Function Tab1CheckProc(ctrlName,enableButton) : CheckBoxControl
   String ctrlName
   Variable enableButton

   ModifyControl button_tab1, disable=(enableButton ? 0 : 2 )
End

Function Tab0CheckProc(ctrlName,enableButton) : CheckBoxControl
   String ctrlName
   Variable enableButton

   ModifyControl button_tab0, disable=(enableButton ? 0 : 2 )
End

// Panel macro that creates a TabControl using TabProc
Window TabbedPanel() : Panel
   PauseUpdate; Silent 1             // building window...
   NewPanel /W=(381,121,614,237) as "Tab Demo"
   TabControl tab, pos={12,9},size={205,91},proc=TabProc,tabLabel(0)="Tab 0"
   TabControl tab, tabLabel(1)="Tab 1",value= 0
   Button button_tab0, pos={54,39},size={110,20},disable=2
   Button button_tab0, title="Button in Tab0"
   Button button_tab1, pos={54,63},size={110,20},disable=1
   Button button_tab1, title="Button in Tab1"
   CheckBox check1_tab1, pos={51,41}, size={117,14}, disable=1, value= 1
   CheckBox check1_tab1, proc=Tab1CheckProc, title="Enable Button in Tab 1"
   CheckBox check0_tab0, pos={51,73}, size={117,14}, proc=Tab0CheckProc
   CheckBox check0_tab0, value= 0, title="Enable Button in Tab 0"
EndMacro
```

Run TabbedPanel to create the panel. Then click on "Tab 0" and "Tab 1" to run TabProc.