You can also provide custom labels in two waves, one numeric and another providing the corresponding text label. For example:

```
NewPanel
Make/O tickNumbers= {0,25,60,100}
Make/O/T tickLabels= {"Off","Slow","Medium","Fast"}
Slider speed,pos={86,28},size={74,73}
Slider speed,limits={0,100,0},value= 40
Slider speed,userTicks={tickNumbers,tickLabels}
```

Often it is sufficient to query the value using **ControlInfo** and you there is no need for an action procedure. If you want to do something every time the value is changed, or to implement a repeating slider, then you need to create an action procedure.

Igor calls the action procedure when the user drags the thumb, when the user clicks the thumb, when the user clicks on either side of the thumb, and when a procedure modifies the slider’s global variable, if any. For a repeating slider, it calls the action procedure periodically while the user clicks the thumb.

See the **Slider** operation on page V-874 and **Repeating Sliders** on page III-418 for further information.

Handling Slider Events

A slider can call your action procedure only when the mouse is released (live mode off) or it can call it each time the slider position changes while the mouse is pressed (live mode on). This section demonstrates how to create both kinds of sliders.

Enter this code in the procedure window of a new experiment. Then execute

```
SliderDemoPanel()
```

in the command line and play with both sliders.

```
Window SliderDemoPanel() : Panel
   PauseUpdate; Silent 1         // building window...
   NewPanel /W=(262,115,665,287)

   TitleBox Title0,pos={46,21},size={139,15},fSize=12,frame=0,fStyle=1
   TitleBox Title0,title="Live Mode Off"
   Slider slider0,pos={197,23},size={150,44},proc=Slider0Proc
   Slider slider0,limits={0,2,0},value=0,live=0,vert=0

   TitleBox Title1,pos={52,114},size={135,15},fSize=12,frame=0,fStyle=1
   TitleBox Title1,title="Live Mode On"
   Slider slider1,pos={197,113},size={150,44},proc=Slider1Proc
   Slider slider1,limits={0,2,0},value=0,live=0,vert=0
EndMacro

Function Slider0Proc(sa) : SliderControl  // Action procedure for slider0
   STRUCT WMSliderAction &sa

   switch(sa.eventCode)
      case -3:       // Control received keyboard focus
      case -2:       // Control lost keyboard focus
      case -1:       // Control being killed
         break
      default:
         if (sa.eventCode & 1) // Value set
            Printf "Value = %g, event code = %d\r", sa.curval, sa.eventCode
         endif
         break
   endswitch

   return 0
```