Save the panel as a recreation macro (Windows→Control→Window Control) to record the final control positions. Rewrite the macro as a function that initially creates the panel:

```
Function CreatePanel()
   KillWindow/Z TabPanel
   NewPanel/N=TabPanel/W=(596,59,874,175) as "Tab Demo Panel"
   TabControl tb,pos={15,19},size={250,80},proc=TabProc
   TabControl tb,tabLabel(0)="Settings"
   TabControl tb,tabLabel(1)="More Settings",value= 0
   CheckBox thisCheck,pos={53,52},size={39,14},title="This"
   CheckBox thisCheck,value= 1,mode=1
   CheckBox thatCheck,pos={53,72},size={39,14},title="That"
   CheckBox thatCheck,value= 0,mode=1
   PopupMenu colorPop,pos={126,60},size={82,20},title="Color"
   PopupMenu colorPop,mode=1,popColor= (65535,0,0)
   PopupMenu colorPop,value= #"\"*COLORPOP*\""
   CheckBox multCheck,pos={50,60},size={16,14},disable=1
   CheckBox multCheck,title="",value= 1
   SetVariable multVar,pos={69,60},size={120,15},disable=1
   SetVariable multVar,title="Multiplier",value=multiplier
End
```

See the **TabControl** operation on page V-1011 for a complete description and examples.

Creating TitleBox Controls

The TitleBox operation creates or modifies a TitleBox control. The control's text can be static or can be tied to a global string variable. See the **TitleBox** operation on page V-1038 for a complete description and examples.

Creating ValDisplay Controls

The **ValDisplay** operation (page V-1060) creates or modifies a value display control.

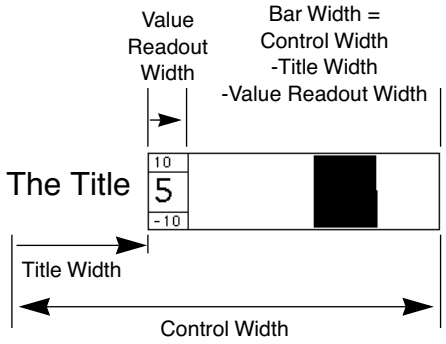
ValDisplay controls are very flexible and multifaceted. They can range from simple numeric readouts to thermometer bars or a hybrid of both. A ValDisplay control is tied to a numeric expression that you provide as an argument to the value keyword. Igor automatically updates the control whenever anything that the numeric expression depends on changes.

ValDisplay controls evaluate their value expression in the context of the root data folder. To reference a data object that is not in the root, you must use a data folder path, such as "root:Folder1:var1".

Here are a few selected keywords extracted from the **ValDisplay** operation on page V-1060:

```
size={width,height}
barmisc={lts, valwidth}
limits={low,high,base}
```

The size and appearance of the ValDisplay control depends primarily on the *valwidth* and size parameters and the width of the title. However, you can use the bodyWidth keyword to specify a fixed width for the body (non-title) portion of the control. Essentially, space for each element is allocated from left to right, with the title receiving first priority. If the control width hasn't all been used by the title, then the value readout width is the smaller of *valwidth* points or what is left. If the control width hasn't been used up, the bar is displayed in the remaining control width:



Here are the various major possible forms of ValDisplay controls. Some of these examples modify previous examples. For instance, the second bar-only example is a modification of the valdisp1 control created by the first bar-only example.

Numeric Readout Only

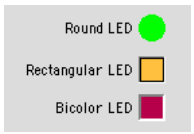
```
// Default readout width (1000) is >= default control width (50)
ValDisplay valdisp0 value=K0
```

LED Display

```
// Create the three LED types
ValDisplay led1,pos={67,17},size={75,20},title="Round LED"
ValDisplay led1,limits={-50,100,0},barmisc={0,0},mode=1
ValDisplay led1,bodyWidth= 20,value= #"K1",zeroColor=(0,65535,0)

ValDisplay led2,pos={38,48},size={104,20},title="Rectangular LED"
ValDisplay led2,frame=5,limits={0,100,0},barmisc={0,0},mode=2
ValDisplay led2,bodyWidth= 20,value= #"K2"
ValDisplay led2,zeroColor= (65535,49157,16385)

ValDisplay led3,pos={60,76},size={82,20},title="Bicolor LED"
ValDisplay led3,limits={-40,100,-100},barmisc={0,0},mode= 2
ValDisplay led3,bodyWidth= 20,value= #"K3"
```



Bar Only

```
// Readout width = 0
ValDisplay valdisp1,frame=1,barmisc={12,0},limits={-10,10,0},value=K0
K0= 5                // halfway from base of 0 to high limit of 10.
```

The nice thing about a bar-only ValDisplay is that you can make it 5 to 200 points tall whereas with a numeric readout, the height is set by the font sizes of the readout and printed limits.

```
// Set control height= 80
ValDisplay valdisp1, size={50,80}
```

Numeric Readout and Bar

```
// 0 < readout width (50) <  control width (150)
ValDisplay valdisp2 size={150,20},frame=1,limits={-10,10,0}
ValDisplay valdisp2 barmisc={0,50},value=K0  // no limits shown
```