Trace User Data

For advanced procedures that manage traces in graphs, you can attach user data to a trace using the userDat a keyword of the ModifyGraph operation. You can retrieve the user data using the GetUserData function.

User-defined Trace Names

As of Igor Pro 6.20, you can provide user-defined names for traces using /TN=<name> with **Display** and **AppendToGraph**. For example:

```
Make/O jack=sin(x/8)
NewDataFolder/O foo; Make/O :foo:jack=sin(x/9)
NewDataFolder/O bar; Make/O :bar:jack=sin(x/10)
Display jack/TN='jack in root', :foo:jack/TN='jack in foo'
AppendToGraph :bar:jack/TN='jack in bar'
ModifyGraph mode('jack in bar')=7,hbFill('jack in bar')=6
ModifyGraph rgb('jack in bar')=(0,0,65535)
```

As of Igor Pro 9.00, you can change the name of an existing trace using ModifyGraph with the traceName keyword.

Trace Name Programming Example

This example illustrates applying some kind of process to each trace in a graph. It appends a smoothed version of each trace to the graph. To try it, copy the code below into the procedure window of a new experiment and execute these commands one-at-a-time:

```
SetupForSmoothWavesDemo()
AppendSmoothedWavesToGraph("", 5)      // Less smoothing
AppendSmoothedWavesToGraph("", 15)     // More smoothing

Function SetupForSmoothWavesDemo()
   Variable numTraces = 3

   Display /W=(35,44,775,522)          // Create graph

   Variable i
   for(i=0; i<numTraces; i+=1)
      String xName, yName
      sprintf xName, "xWave%d", i
      sprintf yName, "yWave%d", i
      Make /O /N=100 $xName = p + 20*i
      Wave xW = $xName
      Make /O /N=100 $yName = p + gnoise(5)
      Wave yW = $yName
      AppendToGraph yW vs xW
   endfor
End

Function CopyTraceOffsets(graphName, sourceTraceName, destTraceName)
   String graphName              // Name of graph or "" for top graph
   String sourceTraceName        // Name of source trace
   String destTraceName          // Name of dest trace

   // info will be "" if no offsets or something like "offset(x)={10,20}"
   String info = TraceInfo(graphName, sourceTraceName, 0)

   String offsetStr = StringByKey("offset(x)", info, "=")   // e.g., "{10,20}"
   Variable xOffset=0, yOffset=0
   if (strlen(offsetStr) > 0)
      sscanf offsetStr, "{%g,%g}", xOffset, yOffset
   endif
```

```
   ModifyGraph offset($destTraceName) = {xOffset, yOffset}
End

Function AppendSmoothedWavesToGraph(graphName, numSmoothingPasses)
   String graphName              // Name of graph or "" for top graph
   Variable numSmoothingPasses   // Parameter to Smooth operation, e.g., 15

   // Get list of all traces in graph
   String traceList = TraceNameList(graphName, ";", 3)
   Variable numTraces = ItemsInList(traceList)
   Variable traceIndex

   // Remove traces representing smoothed waves previously added
   for(traceIndex=0; traceIndex<numTraces; traceIndex+=1)
      String traceName = StringFromList(traceIndex, traceList)
      if (StringMatch(traceName, "*_sm"))
         traceList = RemoveFromList(traceName, traceList)
         numTraces -= 1
         traceIndex -= 1
      endif
   endfor

   // Create smoothed versions of the traces
   for(traceIndex=0; traceIndex<numTraces; traceIndex+=1)
      traceName = StringFromList(traceIndex, traceList)

      Variable isXYTrace = 0

      Wave yW = TraceNameToWaveRef(graphName, traceName)
      DFREF dfr = $GetWavesDataFolder(yW, 1)
      String ySmoothedName = NameOfWave(yW) + "_sm"
      // Create smoothed wave in data folder containing Y wave
      Duplicate /O yW, dfr:$ySmoothedName
      Wave yWSmoothed = dfr:$ySmoothedName
      Smooth numSmoothingPasses, yWSmoothed

      Wave/Z xW = XWaveRefFromTrace(graphName, traceName)
      if (WaveExists(xW))        // It is an XY pair?
         isXYTrace = 1
      endif

      // Append smoothed wave to graph if it is not already in it
      CheckDisplayed /W=$graphName yWSmoothed
      if (V_flag == 0)           // Not yet already in graph?
         if (isXYTrace)
            AppendToGraph yWSmoothed vs xW
         else
            AppendToGraph yWSmoothed
         endif
         ModifyGraph /W=$graphName rgb($ySmoothedName) = (0, 0, 65535)
      endif

      // Copy trace offsets from input trace to smoothed trace
      CopyTraceOffsets(graphName, traceName, ySmoothedName)
   endfor
End
```