Advanced programmers should also be aware of userdata that can be associated with windows using the **SetWindow** operation (see page V-865). Userdata is binary data that persists with individual windows; it is suitable for storing structures. Storing structures in a window's userdata is very handy in eliminating the need for global variables and reduces the bookkeeping needed to synchronize those globals with the window's life cycle. Userdata is also available for use with controls. See the **ControlInfo**, **GetWindow**, **GetUserData**, and **SetWindow** operations.

Here is an example illustrating built-in and user-defined structures along with userdata in a control. Put the following in the procedure window of a new experiment and run the Panel0 macro. Then click on the buttons. Note that the buttons remember their state even if the experiment is saved and reloaded. To fully understand this example, examine the definition of WMButtonAction in the **Button** operation (see page V-55).

```
#pragma rtGlobals=1        // Use modern global access method.

Structure mystruct
   Int32 nclicks
   double lastTime
EndStructure

Function ButtonProc(bStruct) : ButtonControl
   STRUCT WMButtonAction &bStruct

   if( bStruct.eventCode != 1 )
      return 0         // we only handle mouse down
   endif

   STRUCT mystruct s1
   if( strlen(bStruct.userdata) == 0 )
      Print "first click"
   else
      StructGet/S s1,bStruct.userdata
      String ctime= Secs2Date(s1.lastTime, 1 )+" "+Secs2Time(s1.lastTime,1)
   // Warning: Next command is wrapped to fit on the page.
      Printf "button %s clicked %d time(s), last click =
%s\r",bStruct.ctrlName,s1.nclicks,ctime
   endif
   s1.nclicks += 1
   s1.lastTime= datetime
   StructPut/S s1,bStruct.userdata
   return 0
End

Window Panel0() : Panel
   PauseUpdate; Silent 1         // building window…
   NewPanel /W=(150,50,493,133)
   SetDrawLayer UserBack
   Button b0,pos={12,8},size={50,20},proc=ButtonProc,title="Click"
   Button b1,pos={65,8},size={50,20},proc=ButtonProc,title="Click"
   Button b2,pos={119,8},size={50,20},proc=ButtonProc,title="Click"
   Button b3,pos={172,8},size={50,20},proc=ButtonProc,title="Click"
   Button b4,pos={226,8},size={50,20},proc=ButtonProc,title="Click"
EndMacro
```

Limitations of Structures

Although structures can reduce the need for global variables, they do not eliminate them altogether. A structure variable, like all local variables in functions, disappears when its host function returns. In order to maintain state information, you need to store and retrieve structure information using global variables. You can do this using a global variable for each field or, with certain restrictions, you can store entire structure variables in a single global using the **StructPut** operation (see page V-1004) and the **StructGet** operation (see page V-1003).

As of Igor Pro 5.03, a structure can be passed to an external operation or function. See the Igor XOP Toolkit manual for details.

Using StructFill, StructPut, and StructGet

Igor Pro 8 provides the new convenience operation StructFill, which reads in NVAR, SVAR and WAVE fields, along with relaxed limitations for the StructPut and StructGet operations. The following example illustrates these operations.