Clearing a Data Folder

There are times when you might want to clear a data folder before running a procedure, to remove things left over from a preceding run. If the data folder contains no child data folders, you can achieve this with:

```
KillWaves/A/Z; KillVariables/A/Z
```

If the data folder does contain child data folders, you could use the KillDataFolder operation. This operation kills a data folder and its contents, including any child data folders. You could kill the main data folder and then recreate it. A problem with this is that, if the data folder or its children contain a wave that is in use, you will generate an error which will cause your function to abort.

Here is a handy function that kills the contents of a data folder and the contents of its children without killing any data folders and without attempting to kill any waves that may be in use.

```
Function ZapDataInFolderTree(path)
   String path

   String saveDF = GetDataFolder(1)
   SetDataFolder path

   KillWaves/A/Z
   KillVariables/A/Z
   KillStrings/A/Z

   Variable i
   Variable numDataFolders = CountObjects(":", 4)
   for(i=0; i<numDataFolders; i+=1)
      String nextPath = GetIndexedObjName(":", 4, i)
      ZapDataInFolderTree(nextPath)
   endfor

   SetDataFolder saveDF
End
```

Using Strings

This section explains some common ways in which Igor procedures use strings. The most common techniques use built-in functions such as StringFromList and FindListItem. In addition to the built-in functions, there are a number of handy Igor procedure files in the WaveMetrics Procedures:Utilities:String Utilities folder.

Using Strings as Lists

Procedures often need to deal with lists of items. Such lists are usually represented as semicolon-separated text strings. The StringFromList function is used to extract each item, often in a loop. For example:

```
Function Test()
   Make jack,sam,fred,sue
   String list = WaveList("*",";","")
   Print list

   Variable numItems = ItemsInList(list), i
   for(i=0; i<numItems; i+=1)
      Print StringFromList(i,list)     // Print ith item
   endfor
End
```

For lists with a large number of items, using StringFromList as shown above is slow. This is because it must search from the start of the list to the desired item each time it is called. In Igor7 or later, you can iterate quickly using the optional StringFromList offset parameter:

```
Function Test()
   Make jack,sam,fred,sue
```