```
   NVAR gVar2B = $path              // Create NVAR gVar2B

   String/G root:gStr2A = "Two A"
   SVAR gStr2A = root:gStr2A        // Create SVAR gStr2A

   path = "root:gStr2B"
   String/G $path = "Two B"
   SVAR gStr2B = $path              // Create SVAR gStr2B
End
```

In Igor Pro 8.00 or later, you can combine the creation of the reference with the creation of the variable by using the /N flag with **Variable** or **String**:

```
Function Example3()
   String path

   Variable/G root:gVar3A/N=gVar3A = 3

   path = "root:gVar3B"
   Variable/G $path/N=gVar3B = 3

   String/G root:gStr3A/N=gStr3A = "Three A"

   path = "root:gStr3B"
   String/G $path/N=gStr3B = "Three B"
End
```

Wave References

A wave reference is a value that identifies a particular wave. Wave references are used in commands that operate on waves, including assignment statements and calls to operations and functions that take wave reference parameters.

Wave reference variables hold wave references. They can be created as local variables, passed as parameters and returned as function results.

Here is a simple example:

```
Function Test(wIn)
   Wave wIn          // Reference to the input wave received as parameter

   String newName = NameOfWave(wIn) + "_out" // Compute output wave name

   Duplicate/O wIn, $newName                 // Create output wave

   Wave wOut = $newName       // Create wave reference for output wave
   wOut += 1                  // Use wave reference in assignment statement
End
```

This function might be called from the command line or from another function like this:

```
Make/O/N=5 wave0 = p
Test(wave0)                   // Pass wave reference to Test function
```

A Wave statement declares a wave reference variable. It has both a compile-time and a runtime effect.

At compile time, it tells Igor what type of object the declared name references. In the example above, it tells Igor that wOut references a wave as opposed to a numeric variable, a string variable, a window or some other type of object. The Igor compiler allows wOut to be used in a waveform assignment statement (wOut += 1) because it knows that wOut references a wave.