Now we can write the bottleneck function:

```
Function/DF GetPackageDFREF()
   DFREF dfr = root:Packages:'My Package'
   if (DataFolderRefStatus(dfr) != 1)     // Data folder does not exist?
      DFREF dfr = CreatePackageData()      // Create package data folder
   endif
   return dfr
End
```

GetPackageDFREF would be used like this:

```
Function/DF DemoPackageDFREF()
   DFREF dfr = GetPackageDFREF()

   // Read a package variable
   NVAR gVar1 = dfr:gVar1
   Printf "On entry gVar1=%g\r", gVar1

   // Write to a package variable
   gVar1 += 1
   Printf "Now gVar1=%g\r", gVar1
End
```

All functions that access the package data folder should do so through GetPackageDFREF. The calling functions do not need to worry about whether the data folder has been created and initialized because GetPackageDFREF does this for them.

Creating and Accessing the Package Per-Instance Data Folders

Here we extend the technique of the preceding section to handle per-instance data. This example shows how you might handle per-channel data in a data acquisition package. If your package does not use per-instance data then you can skip this section.

First we write a function to create and initialize the per-instance package data folder:

```
Function/DF CreatePackageChannelData(channel)   // Called only from
   Variable channel       // 0 to 3                // GetPackageChannelDFREF

   DFREF dfr = GetPackageDFREF()    // Access main package data folder

   String dfName = "Channel" + num2istr(channel)   // Channel0, Channel1, ...

   // Create the package channel data folder
   NewDataFolder/O dfr:$dfName

   // Create a data folder reference variable
   DFREF channelDFR = dfr:$dfName

   // Initialize per-instance data
   Variable/G channelDFR:gGain = 5.0
   Variable/G channelDFR:gOffset = 0.0

   return channelDFR
End
```

Now we can write the bottleneck function:

```
Function/DF GetPackageChannelDFREF(channel)
   Variable channel       // 0 to 3

   DFREF dfr = GetPackageDFREF()    // Access main package data folder
```