

```

Display yWave vs xWave
SetDataFolder dfSave           // Restore current data folder

// Method 3: Using data folder references
DFREF dfr = root:Packages:'My Package'
Display dfr:yWave vs dfr:xWave
End

```

Using data folder references instead of data folder paths can streamline programs that make heavy use of data folders.

Using Data Folder References

In an advanced application, the programmer often defines a set of named data objects (waves, numeric variables and string variables) that the application acts on. These objects exist in a data folder. If there is just one instance of the set, it is possible to hard-code data folder paths to the objects. Often, however, there will be a multiplicity of such sets, for example, one set per graph or one set per channel in a data acquisition application. In such applications, procedures must be written to act on the set of data objects in a data folder specified at runtime.

One way to specify a data folder at runtime is to create a path to the data folder in a string variable. While this works, you wind up with code that does a lot of concatenation of data folder paths and data object names. Using data folder references, such code can be streamlined.

You create a data folder reference variable with a DFREF statement. For example, assume your application defines a set of data with a wave named wave0, a numeric variable named num0 and a string named str0 and that we have one data folder containing such a set for each graph. You can access your objects like this:

```

Function DoSomething(graphName)
    String graphName
    DFREF dfr = root:Packages:MyApplication:$graphName
    WAVE w0 = dfr:wave0
    NVAR n0 = dfr:num0
    SVAR s0 = dfr:str0
    .
    .
    .
End

```

Igor accepts a data folder reference in any command in which a data folder path would be accepted. For example:

```

Function Test()
    Display root:MyDataFolder:wave0 // OK

    DFREF dfr = root:MyDataFolder
    Display dfr:wave0           // OK

    String path = "root:MyDataFolder:wave0"
    Display $path                // OK. $ converts string to path.

    path = "root:MyDataFolder"
    DFREF dfr = $path            // OK. $ converts string to path.
    Display dfr:wave0           // OK

    String currentDFPath
    currentDFPath = GetDataFolder(1) // OK
    DFREF dfr = GetDataFolder(1)    // ERROR: GetDataFolder returns a string
                                    // not a path.

    DFREF dfr = GetDataFolderDFR() // OK
End

```