Built-in DFREF Functions

Some built-in functions take string data folder paths as parameters or return them as results. Those functions can not take or return data folder references. Here are equivalent DFREF versions that take or return data folder references:

```
   CountObjectsDFR(dfr, type)

   GetDataFolderDFR()

   GetIndexedObjNameDFR(dfr, type, index)

   GetWavesDataFolderDFR(wave)
```

These additional data folder reference functions are available:

```
   DataFolderRefChanges(dfr, changeType)

   DataFolderRefStatus(dfr)

   NewFreeDataFolder()

   DataFolderRefsEqual(dfr1, dfr2)
```

Just as operations that take a data folder path accept a data folder reference, these DFREF functions can also accept a data folder path:

```
Function Test()
   DFREF dfr = root:MyDataFolder
   Print CountObjectsDFR(dfr,1)                // OK
   Print CountObjectsDFR(root:MyDataFolder,1)  // OK
End
```

Checking Data Folder Reference Validity

The **DataFolderRefStatus** function returns zero if the data folder reference is invalid. You should use it to test any DFREF variables that might not be valid, for example, when you assign a value to a data folder reference and you are not sure that the referenced data folder exists:

```
Function Test()
   DFREF dfr = root:MyDataFolder    // MyDataFolder may or may not exist
   if (DataFolderRefStatus(dfr) != 0)
      . . .
   endif
End
```

For historical reasons, an invalid DFREF variable will often act like root.

Data Folder Reference Function Results

A user-defined function can return a data folder reference. This might be used for a subroutine that returns a set of new objects to the calling routine. The set can be returned in a new data folder and the subroutine can return a reference it.

For example:

```
Function/DF Subroutine(newDFName)
   String newDFName
   NewDataFolder/O $newDFName
   DFREF dfr = $newDFName
   Make/O dfr:wave0, dfr:wave1
   return dfr
End
```