DataFolderRefsEqual(*dfr1, dfr2*)
NewFreeDataFolder()

For information on programming with data folder references, see **Data Folder References** on page IV-78.

Data Folders and Commands

Igor normally evaluates commands in the context of the current data folder. This means that, unless qualified with a path to a particular data folder, object names refer to objects in the current data folder. For example:

```
Function Test()
   Make wave1
   Variable/G myGlobalVariable
End
```

This function creates wave1 and myGlobalVariable in the current data folder. Likewise executing:

```
WaveStats wave1
```

operates on wave1 in the current data folder.

Data Folders and User-Defined Functions

To access global variables and waves from a user-defined function, you must first create an NVAR, SVAR or WAVE reference. These references are local objects that point to global objects. See **Accessing Global Variables and Waves** on page IV-65 for details.

Data Folders and Window Recreation Macros

Window recreation macros begin with the Window keyword. They are used to recreate graphs, tables, and other Igor windows and are explained under **Saving a Window as a Recreation Macro** on page II-47.

Window recreation macros are evaluated in the context of the root data folder. In effect, Igor sets the current data folder to root when the window macro starts and restores it when the window macro ends.

Macros that begin with the Macro or Proc keywords evaluate their commands in the context of the *current* data folder.

Evaluating window recreation macros this way ensures that a window is recreated correctly regardless of the current data folder, and provides some compatibility with window macros created with prior versions of Igor Pro which didn't have data folders.

This means that object names within window recreation macros that don't explicitly contain a data folder path refer to objects in the root data folder.

Data Folders and Assignment Statements

Wave and variable assignment statements are evaluated in the context of the data folder containing the wave or variable on the left-hand side of the statement:

```
root:subfolder:wave0 = wave1 + var1
```

is a shorter way of writing the equivalent:

```
root:subfolder:wave0 = root:subfolder:wave1 + root:subfolder:var1
```

This rule also applies to dependency formulae which use := instead of = as the assignment operator.

Data Folders and Controls

ValDisplay controls evaluate their value expression in the context of the root data folder.

SetVariable controls remember the data folder in which the controlled global variable exists, and continue to function properly when the current data folder is different from the controlled variable.