Free data folders can not be used in situations where global persistence is required such as in graphs, tables and controls. In other words, you should use objects in free data folders for short-term computation purposes only.

For a discussion of multithreaded assignment statements, see **Automatic Parallel Processing with MultiThread** on page IV-323. For an example using free data folders, see **Data Folder Reference MultiThread Example** on page IV-325.

Free Data Folder Lifetime

A free data folder is automatically deleted when the last reference to it disappears.

Data folder references can be stored in:

1. Data folder reference variables in user-defined functions
2. Data folder reference fields in structures
3. Elements of a data folder reference wave (created with Make/DF)
4. Igor's internal current data folder reference variable

A data folder reference disappears when:

1. The data folder reference variable containing it is explicitly cleared using KillDataFolder.
2. The data folder reference variable containing it is reassigned to refer to another data folder.
3. The data folder reference variable containing it goes out-of-scope and ceases to exist when the function in which it was created returns.
4. The data folder reference wave element containing it is deleted or the data folder reference wave is killed.
5. The current data folder is changed which causes Igor's internal current data folder reference variable to refer to another data folder.

When there are no more references to a free data folder, Igor automatically deletes it.

In this example, a free data folder reference variable is cleared by KillDataFolder:

```
Function Test1()
   DFREF dfr= NewFreeDataFolder()      // Create new free data folder.
   // The free data folder exists because dfr references it.
   . . .
   KillDataFolder dfr   // dfr no longer refers to the free data folder
End
```

KillDataFolder kills the free data folder only if the given DFREF variable contains the last reference to it.

In the next example, the free data folder is automatically deleted when the DFREF that references it is changed to reference another data folder:

```
Function Test2()
   DFREF dfr= NewFreeDataFolder()      // Create new free data folder.
   // The free data folder exists because dfr references it.
   . . .
   DFREF dfr= root:
   // The free data folder is deleted since there are no references to it.
End
```

In the next example, a free data folder is created and a reference is stored in a local data folder reference variable. When the function ends, the DFREF ceases to exist and the free data folder is automatically deleted:

```
Function Test3()
   DFREF dfr= NewFreeDataFolder()      // Create new free data folder.
   // The free data folder exists because dfr references it.
```