

Data Folder Syntax

Data folders are named objects like other Igor objects such as waves and variables. Data folder names follow the same rules as wave names. See **Liberal Object Names** on page III-501.

Igor Pro's data folders use the colon character (:) to separate components of a path to an object.

A data folder named "root" always exists and contains all other data folders.

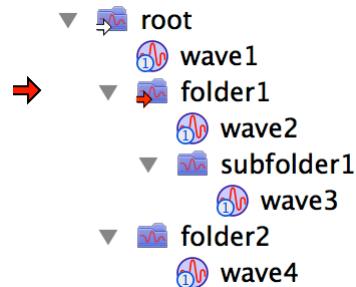
A given object can be specified in a command using:

- A full path
- A partial path
- Just the object name

The object name alone can only be used when the current data folder contains the object.

A full path starts with "root" and does not depend on the current data folder. A partial path starts with ":" and is relative to the current data folder.

Assume the data folder structure shown below, where the arrow indicates that folder1 is the current data folder.



Each of the following commands creates a graph of one of the waves in this hierarchy:

```

Display wave2
Display :subfolder1:wave3
Display root:folder1:subfolder1:wave3
Display ::folder2:wave4
  
```

The last example illustrates the rule that you can use multiple colons to walk back up the hierarchy: from folder1 (the current data folder), up one level, then down to folder2 and wave4. Here is another valid, though silly, example:

```
Display root:folder1:subfolder1::::folder2:wave4
```

Occasionally you need to specify a data folder itself rather than an object in a data folder. In that case, just leave off the object name. The path specification should therefore have a trailing colon. However, Igor will generally understand what you mean if you omit the trailing colon.

If you need to specify the current data folder, you can use just a single colon. For example:

```
KillDataFolder :
```

kills the current data folder, and all its contents, and then sets the current data folder to the parent of the current. Non-programmers might prefer to use the Data Browser to delete data folders.

Recall that the \$ operator converts a string expression into a single name. Since data folders are named, the following is valid:

```

String df1 = "folder1", df2="subfolder1"
Display root:$ (df1) :$ (df2) :wave3
  
```

This is a silly example but the technique would be useful if df1 and df2 were parameters to a procedure.