```
   if (dfrStatus & 2)            // Bit 1 set means free data folder
      Print "Data folder reference refers to a free data folder"
   endif

   if (dfrStatus == 1)
      Print "Data folder reference refers a global data folder"
      DFREF dfSav = GetDataFolderDFR()
      Print GetDataFolder(1)     // Print data folder path
      SetDataFolder dfSav
   endif

   Make/O dfr:jack=sin(x/8)      // Make a wave in the referenced data folder

   return 0
End
```

See Also

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

Differentiate

Differentiate [***type flags***][***flags***] ***yWaveA*** [**/X =** ***xWaveA***]
   [**/D =** ***destWaveA***][, ***yWaveB*** [**/X =** ***xWaveB***][**/D =** ***destWaveB***][, …]]

The Differentiate operation calculates the 1D numerical derivative of a wave.

Differentiate is multi-dimension-aware in the sense that it computes a 1D differentiation along the dimension specified by the /DIM flag or along the rows dimension if you omit /DIM.

Complex waves have their real and imaginary components differentiated individually.

Flags

/DIM=*d* Specifies the wave dimension along which to differentiate when *yWave* is multidimensional.

d=-1: Treats entire wave as 1D (default).

d=0: Differentiates along rows.

d=1: Differentiates along columns.

d=2: Differentiates along layers.

d=3: Differentiates along rows.

For example, for a 2D wave, /DIM=0 differentiates each row and /DIM=1 differentiates each column.

/EP=*e* Controls end point handling.

e=0: Replaces undefined points with an approximation (default).

e=1: Deletes the point(s).

/METH=*m* Sets the differentiation method.

m=0: Central difference (default).

m=1: Forward difference.

m=2: Backward difference.

/P Forces point scaling.

Type Flags (*used only in functions*)

Differentiate also can use various type flags in user functions to specify the type of destination wave reference variables. These type flags do not need to be used except when needed to match another wave reference variable of the same name or to identify what kind of expression to compile for a wave assignment. See **WAVE Reference Types** on page IV-73 and **WAVE Reference Type Flags** on page IV-74 for a complete list of type flags and further details.

For example, when the input (and output) waves are complex, the output wave will be complex. To get the Igor compiler to create a complex output wave reference, use the /C type flag with /D=destwave: