

## Chapter IV-3 — User-Defined Functions

Another way is to call an Igor operation that creates an output wave inside a free data folder. This happens if you set a free data folder as the current data folder and then call an operation, such as ColorTab2Wave, that creates an output wave in the current data folder.

### Converting a Free Wave to a Global Wave

You can use **MoveWave** to move a free wave into a global data folder, in which case it ceases to be free. If the wave was created by NewFreeWave its name will be '\_free\_'. You can use MoveWave to provide it with a more descriptive name.

Here is an example illustrating Make/FREE:

```
Function Test()
    Make/FREE/N=(50,50) w
    SetScale x,-5,8,w
    SetScale y,-7,12,w
    w= exp(-(x^2+y^2))
    NewImage w
    if( GetRTError(1) != 0 )
        Print "Can't use a free wave here"
    endif
    MoveWave w,root:wasFree
    NewImage w
End
```

Note that MoveWave requires that the new wave name, wasFree in this case, be unique within the destination data folder.

To determine if a given wave is free or global, use the **WaveType** function with the optional selector = 2.

## Free Data Folders

Free data folders are data folders that are not part of any data folder hierarchy. Their principal use is in multithreaded wave assignment using the MultiThread keyword in a function. They can also be used for temporary storage within functions.

Free data folders are recommended for advanced programmers only.

A wave that is stored in a free data folder or its descendants is called a “local” wave to distinguish it from a “global” wave which is stored in the root data folder or its descendants and from a “free” wave which is stored in no data folder.

**Note:** Free data folders are saved only in packed experiment files. They are not saved in unpacked experiments and are not saved by the SaveData operation or the Data Browser's Save Copy button. In general, they are intended for temporary computation purposes only.

You create a free data folder using the NewFreeDataFolder function. You access it using the data folder reference returned by that function.

After using SetDataFolder with a free data folder, be sure to restore it to the previous value, like this:

```
Function Test()
    DFREF dfrSave = GetDataFolderDFR()

    SetDataFolder NewFreeDataFolder()    // Create new free data folder.

    . . .

    SetDataFolder dfrSave
End
```

This is good programming practice in general but is especially important when using free data folders.