ThreadGroupPutDF clips the specified data folder, and everything it contains, out of the source thread's data hierarchy and puts it in the queue. From the standpoint of the source thread, it is as if KillDataFolder had been called. While a data folder resides in a queue, it is not accessible by any thread. See the documentation for **ThreadGroupPutDF** for some warnings about its use.

ThreadGroupGetDFR removes the data folder from the queue and returns it, as a free data folder, to the calling thread. Because it is a free data folder, Igor will automatically delete it when there are no more references to it, for example, when the thread returns.

Except for waves passed to the thread worker function as parameters and the thread worker's return value, the input and output queues are the only way for a thread to share data with the main thread. Examples below illustrate the use of these queues.

Parallel Processing - Group-at-a-Time Method

In this example, we attempt to improve the speed of filling columns of a 2D wave with a sin function. The traditional method is compared with parallel processing. Notice how much more complicated the multithreaded version, MTFillWave, is compared to the single threaded STFillWave.

```
ThreadSafe Function MyWorkerFunc(w,col)
    WAVE w
    Variable col

    w[][col]= sin(x/(col+1))

    return stopMSTimer(-2)            // Time when we finished
End

Function MTFillWave(dest)
    WAVE dest

    Variable ncol= DimSize(dest,1)
    Variable i,col,nthreads= ThreadProcessorCount
    Variable threadGroupID= ThreadGroupCreate(nthreads)

    for(col=0; col<ncol;)
        for(i=0; i<nthreads; i+=1)
            ThreadStart threadGroupID,i,MyWorkerFunc(dest,col)
            col+=1
            if( col>=ncol )
                break
            endif
        endfor

        do
            Variable threadGroupStatus= ThreadGroupWait(threadGroupID,100)
        while( threadGroupStatus != 0 )
    endfor
    Variable dummy= ThreadGroupRelease(threadGroupID)
End

Function STFillWave(dest)
    WAVE dest

    Variable ncol= DimSize(dest,1)
    Variable col

    for(col= 0;col<ncol;col+=1)
        MyWorkerFunc(dest,col)
    endfor
End

Function ThreadTest(rows)
    Variable rows

    Variable cols=10

    make/o/n=(rows,cols) jack

    Variable i

    for(i=0;i<10;i+=1)      // get any pending pause events out of the way
    endfor
```