

# What a drag: Dragging a virtual file (IStream edition)

 [devblogs.microsoft.com/oldnewthing/20080319-00](http://devblogs.microsoft.com/oldnewthing/20080319-00)

March 19, 2008



Raymond Chen

Last time, we saw how to drag a virtual file whose contents are expressed as a block of bytes in memory (HGLOBAL). Often, a block of bytes is not a convenient way to express the contents of a virtual file. You might prefer to express it as a stream. For example, the contents might be dynamically generated (say by the output of an algorithm), or it might come in from an external source (say, a web page that is being downloaded). Let's take our program from last time and convert it to return the file contents in the form of a stream. The first change we need to make is to our constructor, telling it to report file contents as a stream rather than as an `HGLOBAL` :

```
#include <shlwapi.h> // for SHOpenRegStream

CTinyDataObject::CTinyDataObject() : m_cRef(1)
{
    SetFORMATETC(&m_rgfe[DATA_FILEGROUPDESCRIPTOR],
                RegisterClipboardFormat(CFSTR_FILEDESCRIPTOR));
    SetFORMATETC(&m_rgfe[DATA_FILECONTENTS],
                RegisterClipboardFormat(CFSTR_FILECONTENTS),
                TYMED_ISTREAM, /* lindex */ 0);
}
```

Next, we need to produce that stream and its corresponding descriptor in our `IDataObject::GetData` handler:

```

HRESULT CTinyDataObject::GetData(FORMATETC *pfe, STGMEDIUM *pmed)
{
    ZeroMemory(pmed, sizeof(*pmed));

    switch (GetDataIndex(pfe)) {
    case DATA_FILEGROUPDESCRIPTOR:
    {
        FILEGROUPDESCRIPTOR fgd;
        ZeroMemory(&fgd, sizeof(fgd));
        fgd.cItems = 1;
        StringCchCopy(fgd.fgd[0].cFileName,
                     ARRAYSIZE(fgd.fgd[0].cFileName),
                     TEXT("Dummy"));
        pmed->tymed = TYMED_HGLOBAL;
        return CreateHGlobalFromBlob(&fgd, sizeof(fgd),
                                     GMEM_MOVEABLE, &pmed->hGlobal);
    }

    case DATA_FILECONTENTS:
        pmed->tymed = TYMED_ISTREAM;
        pmed->pstm = SHOpenRegStream(HKEY_LOCAL_MACHINE,
                                   TEXT("Hardware\\Description\\System\\CentralProcessor\\0"),
                                   TEXT("~MHz"), STGM_READ);
        // set the stream position properly
        if (pmed->pstm) {
            LARGE_INTEGER liZero = { 0, 0 };
            pmed->pstm->Seek(liZero, STREAM_SEEK_END, NULL);
        }
        return pmed->pstm ? S_OK : E_FAIL;
    }

    return DV_E_FORMATETC;
}

```

Of course, in real life, you would use a more interesting stream than your CPU speed. I just chose that one as an example.

As with our **HGLOBAL**-based data object, you can drop this data object onto an Explorer folder to create a file, into an Outlook message to create an attachment, and anywhere else a program supports the shell virtual file transfer model. And as with the **HGLOBAL** example, you can set various optional information in the **FILEGROUPDESCRIPTOR** in order to make the transfer go more smoothly, particularly the expected stream size. But I won't go into it because the theme of this series is "It's the least you can do".

But already you know enough to solve this customer's problem:

We need to know what directory the user dropped a file onto. We need to transfer data from another computer, so what we do is have the user drag a single dummy file, and then once we find out where the user dropped the dummy file, we can go in, delete the dummy file, and start transferring the data from the remote computer and saving it into real files in the destination directory.

Next time, we'll look at the final storage medium that can be used for file transfer, the `TYMED_ISTORAGE` .

Raymond Chen

**Follow**

