The stream pointer position in IDataObject::GetData and IDataObject::GetDataHere is significant

devblogs.microsoft.com/oldnewthing/20140918-00

September 18, 2014

Raymond Chen

An oft-overlooked detail of the `IDataObject::GetData` and `IDataObject::GetDataHere` methods is the position of the stream pointer when the result is a stream. These rules are buried in the documentation, so I'm going to call them out louder.

Let's look at `IDataObject::GetData` first.

If `IDataObject::GetData` returns a stream, then the stream pointer must be positioned at the *end* of the stream before the stream is returned. In other words, the last thing you do before returning the stream is seek it to the end. The contents of the data object are assumed to extend from the start of the stream to the stream's position as returned by `IDataObject::GetData`. (And no, I don't know why this rule exists.)

I messed this up in my demonstration of how to drag a stream. Let's fix it.

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

But what if you don't know the stream size? For example, what if the stream is coming from a live download? What if the stream doesn't support seeking? What if the stream is infinite? In those cases, you don't really have a choice. You just leave the stream pointer at the beginning and hope for the best. (Fortunately, at least in the case of virtual file content, the shell is okay with people who leave the stream pointer at the start of the stream. Probably for reasons like this.)

There is a similar detail with `IDataObject::GetDataHere`: If you are asked to produce the data into an existing stream, you should write the data starting at the stream's current position and leave the stream pointer at the end of the data you just wrote.

Raymond Chen

Follow