

What a drag: Dragging a Uniform Resource Locator (URL) and text

devblogs.microsoft.com/oldnewthing/20080313-00

March 13, 2008



Raymond Chen

We've learned [how to drag text](#) and [how to drag a uniform resource locator](#), but what if want to drag both? Well, it's actually a simply matter of saying that you have both (and actually producing it on demand).

Start by extending the enumeration of data types to include both URL and text:

```
enum {
    DATA_URL,
    DATA_TEXT,
    DATA_NUM,
    DATA_INVALID = -1,
};
```

There is a subtlety in the way we set up this enumeration: We put `DATA_URL` ahead of `DATA_TEXT` so that the clipboard formats that are produced by `SHCreateStdEnumFormatEtc` are generated in priority order (highest quality first). By enumerating URL first, a program that understands both URL and text formats will know to prefer URL. In this particular case, it's not that critical since most text editors do URL auto-detection (at least if the URL begins with "http:."), but in the more general case, enumerating the formats in priority order can make a big difference. For example, your data object might provide text in both Rich Text Format as well as plain text, and it would probably be better if a program that understood both formats used the rich text version by default rather than the plain text version.

Once we have both formats available, we need to respond to both of them. In our constructor, we have to describe both of these formats so that `GetDataIndex` and `EnumFormatEtc` know about them.

```

CTinyDataObject::CTinyDataObject() : m_cRef(1)
{
    SetFORMATETC(&m_rgfe[DATA_URL],
                RegisterClipboardFormat(CFSTR_SHELLURL));
    SetFORMATETC(&m_rgfe[DATA_TEXT], CF_TEXT);
}

```

And then the change to the heart of the data object, the `IDataObject::GetData` method, is anticlimactic:

```

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

    switch (GetDataIndex(pfe)) {
    case DATA_URL:
    case DATA_TEXT:
        pmed->tymed = TYMED_HGLOBAL;
        return CreateHGlobalFromBlob(c_szURL, sizeof(c_szURL),
                                    GMEM_MOVEABLE, &pmed->hGlobal);
    }

    return DV_E_FORMATETC;
}

```

Whether the caller asks for either text or a URL, we give them the same string back.

When you run this program, observe that it has the combined functionality of the two previous programs. You can drag text into Wordpad, drop an URL onto Firefox, and drop an URL onto Internet Explorer.

These were all just warm-ups. After a short break, we'll roll up our sleeves and begin providing more complicated data in our data object.

Raymond Chen

Follow

