

# What a drag: Dragging a Uniform Resource Locator (URL)

[devblogs.microsoft.com/oldnewthing/20080312-00](http://devblogs.microsoft.com/oldnewthing/20080312-00)

March 12, 2008



Raymond Chen

Last time, we dragged some text around and found that the text would be interpreted as a URL if you dropped it onto Firefox, but Internet Explorer was not as willing to accept it. Today, we'll make the data object work for Internet Explorer.

The only change is that we have to provide the URL in the form of a `CFSTR_SHELLURL` clipboard format rather than as `CF_TEXT`. Take the program from last time and make two changes. First, use the handy-dandy search-and-replace function to change `DATA_TEXT` to `DATA_URL` throughout. (This step isn't technically necessary, but it's nice to have the name match its usage.) The real work happens in this change to the constructor:

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

That's all; just change the clipboard format from `CF_TEXT` to `CFSTR_SHELLURL`. It is important to note that `CFSTR_SHELLURL` represents an ANSI string. Since "URLs are written only with the graphic printable characters of the US-ASCII coded character set," there is no loss of expressiveness by restricting to ANSI.

Run this new program and now you can click in the client area and drag/drop the (invisible) object onto Internet Explorer, where it will navigate to Microsoft's home page. (If your system supports Active Desktop, you can also drag/drop the invisible object to the desktop and create an Active Desktop component.)

Okay, so we have one version of the program that can drag a URL to Internet Explorer, and another version that can drag a URL to Firefox. Next time, we'll combine them to have a single data object that can drop to both. It's quite embarrassingly simple (because I planned it that way).

Raymond Chen

**Follow**

