Dragging a shell object, part 4: Adding a prettier drag icon

devblogs.microsoft.com/oldnewthing/20041209-00

December 9, 2004

Raymond Chen

You may have noticed that the drag feedback is rather sad-looking. Just a box, maybe with a plus sign or an arrow; you don't even know what it is you're dragging.

Let's fix that. We'll drag the icon of the file around. We'll need to add the drag image to the data object.

```
void OnLButtonDown(HWND hwnd, BOOL fDoubleClick, int x, int y, UINT keyFlags)
{
  IDataObject *pdto;
  if (SUCCEEDED(GetDataObjectOfFileWithCuteIcon(
              hwnd, g_pszTarget, &pdto))) {
     IDropSource *pds = new CDropSource();
     …
```

This new function `GetDataObjectOfFileWithCuteIcon` creates the data object and then attaches the cute icon to it.

```
HRESULT GetDataObjectOfFileWithCuteIcon(HWND hwnd,
 LPCWSTR pszPath, IDataObject **ppdto)
{
  HRESULT hr = GetUIObjectOfFile(hwnd, pszPath,
                    IID_IDataObject, (void**)ppdto);
  if (SUCCEEDED(hr)) {
    IDragSourceHelper *pdsh;
    if (SUCCEEDED(CoCreateInstance(CLSID_DragDropHelper, NULL, CLSCTX_ALL,
                                IID_IDragSourceHelper, (void**)&pdsh))) {
      SHDRAGIMAGE sdi;
      if (CreateDragImage(pszPath, &sdi)) {
        pdsh->InitializeFromBitmap(&sdi, *ppdto);
        DeleteObject(sdi.hbmpDragImage);
      }
      pdsh->Release();
    }
  }
  return hr;
}
```

We use the shell drag/drop helper object to attach the bitmap to the data object. The shell drag/drop helper object requires that the data object be able to accept arbitrary blobs, but fortunately, the standard shell data object does this.

The nasty part is generating the drag image. This is not the fun part, and you're not going to learn anything from this function. It just has to be written.

```
BOOL CreateDragImage(LPCWSTR pszPath, SHDRAGIMAGE *psdi)
{
  psdi->hbmpDragImage = NULL;
  SHFILEINFOW sfi;
  HIMAGELIST himl = (HIMAGELIST)
    SHGetFileInfoW(pszPath, 0, &sfi, sizeof(sfi), SHGFI_SYSICONINDEX);
  if (himl) {
    int cx, cy;
    ImageList_GetIconSize(himl, &cx, &cy);
    psdi->sizeDragImage.cx = cx;
    psdi->sizeDragImage.cy = cy;
    psdi->ptOffset.x = cx;
    psdi->ptOffset.y = cy;
    psdi->crColorKey = CLR_NONE;
    HDC hdc = CreateCompatibleDC(NULL);
    if (hdc) {
      psdi->hbmpDragImage = CreateBitmap(cx, cy, 1, 32, NULL);
      if (psdi->hbmpDragImage) {
        HBITMAP hbmPrev = SelectBitmap(hdc, psdi->hbmpDragImage);
        ImageList_Draw(himl, sfi.iIcon, hdc, 0, 0, ILD_NORMAL);
        SelectBitmap(hdc, hbmPrev);
      }
      DeleteDC(hdc);
    }
  }
  return psdi->hbmpDragImage != NULL;
}
```

To create the drag image, we ask the `SHGetFileInfo` function to give us the imagelist handle and icon index for the icon that represents the file. The icon size in the imagelist goes into the `SHDRAGIMAGE` structure as the bitmap dimensions and as the cursor point. (We put the cursor at the bottom right corner of the image.) Since we are creating an alpha-blended bitmap, we don't need a color-key. Finally, we create a memory DC to house an ARGB bitmap into which we draw the icon.

If you run this program, you should see the icon for a text file being dragged around as you drag your throwaway file around the screen.

Next time, a way to make somebody else do the heavy lifting for you.

Raymond Chen

Follow