

# Using the MNS\_DRAGDROP style: Dropping in

---

 [devblogs.microsoft.com/oldnewthing/20111229-00](http://devblogs.microsoft.com/oldnewthing/20111229-00)

December 29, 2011



Raymond Chen

Last time, we looked at using the MNS\_DRAGDROP style for dragging items out of a menu. Today, we'll look at dropping them in.

Take the program from last time and make the following additions. First, let's add a second item to the menu.

```

// resource header file
#define IDM_MAIN 1
#define IDC_CLOCK 100
#define IDC_WMP 101
// resource file
IDM_MAIN MENU PRELOAD
BEGIN
    POPUP "&Test"
    BEGIN
        MENUITEM "&Clock", IDC_CLOCK
        MENUITEM "&WMP", IDC_WMP
    END
END
// scratch.cpp
HRESULT GetMenuObject(HWND hwnd, HMENU hmenu, UINT uPos,
                     REFIID riid, void **ppvOut)
{
    HRESULT hr = E_NOTIMPL;
    *ppvOut = NULL;
    if (hmenu == GetSubMenu(GetMenu(hwnd), 0)) {
        switch (GetMenuItemID(hmenu, uPos)) {
            case IDC_CLOCK:
                hr = GetUIObjectOfFile(hwnd, L"C:\\Windows\\clock.avi",
                                       riid, ppvOut);

                break;
            case IDC_WMP:
                hr = GetUIObjectOfFile(hwnd, L"C:\\Program Files"
                                       L"\\Windows Media Player\\wmplayer.exe",
                                       riid, ppvOut);

                break;
        }
    }
    return hr;
}

```

Yes, I hard-coded another path. This is a demo, not production code.

Anyway, it's time to hook up the `WM_MENUGETOBJECT` message:

```

#define HANDLE_WM_MENUGETOBJECT(hwnd, wParam, lParam, fn) \
    (fn)((hwnd), (MENUGETOBJECTINFO*)(lParam))
LRESULT OnMenuGetObject(HWND hwnd, MENUGETOBJECTINFO *pmgoi)
{
    LRESULT lres = MNGO_NOINTERFACE;
    if (!(pmgoi->dwFlags & (MNGOF_BOTTOMGAP | MNGOF_TOPGAP)) &&
        SUCCEEDED(GetMenuObject(hwnd, pmgoi->hmenu, pmgoi->uPos,
                                *(IID*)pmgoi->riid, &pmgoi->pvObj))) {
        lres = MNGO_NOERROR;
    }
    return lres;
}
HANDLE_MSG(hwnd, WM_MENUGETOBJECT, OnMenuGetObject);

```

To handle the `WM_MENUGETOBJECT` message, you convert the `hmenu` , `uPos` pair into a COM object, requesting the interface provided by the `riid` member, and putting the result into the `pvObj` member. (Exercise: Why is the `riid` member typed as `void *` rather than `REFIID` ?)

When the user tries to drop on a menu item, we just give them the corresponding object in the shell namespace. Notice that I filter out the `GAP` messages, since they indicate that the user is trying to drop *between* items rather than on them.

Run this program, open the *Test* menu, and drag the Clock menu item onto the WMP menu item. If all goes well (assuming you changed the path for `clock.avi` to some other AVI file), the AVI file will be opened by Windows Media Player, since that's the behavior of Windows Media Player when you drop an AVI file on it.

So that's menu drag/drop. It's really not all that exciting. Of course, what people tend to be most interested in is not generic drag/drop for menus but menu customization via drag/drop. That's not something that `MNS_DRAGDROP` gives you directly; that's something you need to build yourself out of the building blocks provided.

We'll snap some blocks together next time.

Raymond Chen

**Follow**

