Getting the location of the Close button in the title bar, from Windows 2000 or Windows XP

devblogs.microsoft.com/oldnewthing/20140630-00

June 30, 2014

Raymond Chen

Today's Little Program locates the × button in the corner of the window and displays a balloon tip pointing at it. We did this some time ago with the help of the `WM_GETTITLEBAR-INFOEX` message, which is new for Windows Vista. But what if you don't have that message available, say, because you're running on Windows 2000 or Windows XP or (gasp) Windows 98?

You can use the classic Accessibility interface `IAccessible` to enumerate the buttons in the title bar and see which one the window reports as the Close button.

Let's take the program from last time and change the `GetCloseButtonCenter` function:

```
#include <oleacc.h>
#include <atlbase>
BOOL GetCloseButtonCenter(HWND hwnd, POINT *ppt)
{
 CComPtr<IAccessible> spacc;
 if (FAILED(AccessibleObjectFromWindow(hwnd, OBJID_TITLEBAR,
                  IID_PPV_ARGS(&spacc)))) return FALSE;
 CComQIPtr<IEnumVARIANT> spenum(spacc);
 if (!spenum) return FALSE;
 for (CComVariant vtChild; spenum->Next(1, &vtChild, nullptr) == S_OK;
      vtChild.Clear()) {
  CComVariant vtState;
  if (FAILED(spacc->get_accState(vtChild, &vtState))) continue;
  if (vtState.vt != VT_I4) continue;
  if (vtState.lVal & (STATE_SYSTEM_INVISIBLE |
                      STATE_SYSTEM_OFFSCREEN |
                      STATE_SYSTEM_UNAVAILABLE)) continue;
  long left, top, width, height;
  if (FAILED(spacc->accLocation(&left, &top, &width, &height,
                                vtChild))) continue;
  POINT pt = { left + width / 2, top + height / 2 };
  if (SendMessage(hwnd, WM_NCHITTEST, 0,
                  MAKELPARAM(pt.x, pt.y)) == HTCLOSE) {
   *ppt = pt;
   return TRUE;
  }
 }
 return FALSE;
}
```

We obtain the `IAccessible` interface for the title bar and proceed to enumerate its children. For each child, we get its location, and then use the `WM_NCHITTEST` message to determine programmatically what that location corresponds to. If the answer is "This is the Close button," then we found the button and report its center.

Note that this once again highlights the distinction between `WM_NCMOUSEMOVE` and `WM_NCHITTEST`. Hit-testing can occur for reasons other than mouse movement.

Exercise: Why couldn't we use the `IAccessible::get_accName` method to figure out which button each child represents?

Raymond Chen

Follow