How do I suppress the CapsLock warning on password edit controls?

devblogs.microsoft.com/oldnewthing/20081010-00

October 10, 2008

Raymond Chen

One of the features added to version 6 of the shell common controls is a warning balloon that appears if CapsLock is on in a password control. Let's demonstrate. Take the scratch program, add a manifest that requests version 6 of the common controls (perhaps by using a Visual C++ extension), and add the following:

```
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hwndChild = CreateWindow(TEXT("edit"), NULL,
            ES_PASSWORD | WS_CHILD | WS_VISIBLE, 0, 0,
            0, 0, hwnd, NULL, g_hinst, 0);
    if (!g_hwndChild) return FALSE;


    return TRUE;
}
```

Run this program and hit the CapsLock key. The warning balloon should appear. (If it doesn't, then your manifest is probably not working.)

Suppose you want to suppress this warning balloon. Why? I don't know. Maybe you want to confuse your user. Maybe you think it looks ugly. Whatever the reason, you can suppress the balloon by subclassing the edit control and swallowing the `EM_SHOWBALLOONTIP` message.

```
WNDPROC g_wpEdit;


LRESULT CALLBACK NoBalloonWndProc(HWND hwnd, UINT uMsg,
                                  WPARAM wParam, LPARAM lParam)
{
  switch (uMsg) {
  case EM_SHOWBALLOONTIP: return FALSE;
  }
  return CallWindowProc(g_wpEdit, hwnd, uMsg, wParam, lParam);
}


BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hwndChild = CreateWindow(TEXT("edit"), NULL,
           ES_PASSWORD | WS_CHILD | WS_VISIBLE, 0, 0,
           0, 0, hwnd, NULL, g_hinst, 0);
    if (!g_hwndChild) return FALSE;


    g_wpEdit = SubclassWindow(g_hwndChild, NoBalloonWndProc);


    return TRUE;
}
```

When you run this modified program, you'll see that the balloon tip no longer appears because the subclass procedure intercepts all the balloon tips before the default edit control window procedure can see them.

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

Follow