# SHAutoComplete giveth, and SHAutoComplete taketh away

devblogs.microsoft.com/oldnewthing/20100521-00

May 21, 2010

Raymond Chen

The `SHAutoComplete` function lets you attach autocomplete functionality to an edit control, and there are flags that describe what sources you want the autocomplete to draw from. If you call `SHAutoComplete` a second time, the second set of flags replace the original flags. The flags do not accumulate. For example, if you first call `SHAuto-Complete(SHACF_FILESYS_ONLY)`, and then you later call `SHAuto-Complete(SHACF_URLHISTORY)`, the result is that the autocompletion uses only the URL history.

This replacement behavior (as opposed to accumulation behavior) is handy if you want to *remove* an autocompletion that you previously added. You just call `SHAutoComplete` a second time and leave off the flags for autocomplete sources you don't want. There's a catch, though: If you want to turn off everything, then you cannot pass zero, because that gets interpreted as `SHACF_DEFAULT`. You have to pass a nonzero value, and fortunately there's a handy nonzero value which means *Turn off everything*: `SHACF_AUTOSUGGEST_FORCE_OFF`.

Let's illustrate this technique by disabling autocomplete in the common dialog, a problem which commenter Ian mistakenly solved by modifying a global setting.

```
#include <windows.h>
#include <shlwapi.h>
#include <commctrl.h>
#include <commdlg.h>
#include <dlgs.h>
UINT_PTR CALLBACK HookProc(HWND hdlg, UINT uMsg,
                           WPARAM wParam, LPARAM lParam)
{
    switch (uMsg) {
    case WM_INITDIALOG:
        PostMessage(hdlg, WM_APP, 0, 0);
        break;
    case WM_APP:
        SHAutoComplete(
          (HWND)SendDlgItemMessage(GetParent(hdlg), cmb13,
                                   CBEM_GETEDITCONTROL, 0, 0),
          SHACF_AUTOSUGGEST_FORCE_OFF);
        break;
    }
    return 0;
}
int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
                   LPSTR lpCmdLine, int nShowCmd)
{
    TCHAR szFile[MAX_PATH];
    szFile[0] = TEXT('\0');
    OPENFILENAME ofn = { sizeof(ofn) };
    ofn.hInstance = hinst;
    ofn.lpstrFilter = TEXT("All files\0*.*\0");
    ofn.lpstrFile = szFile;
    ofn.nMaxFile = MAX_PATH;
    ofn.Flags = OFN_ENABLEHOOK | OFN_EXPLORER;
    ofn.lpfnHook = HookProc;
    GetOpenFileName(&ofn);
    return 0;
}
```

The hook procedure uses the `SHAutoComplete` function to turn off autocompletion on the file name edit control in the common dialog. There are a few annoying bits that I have to get through before I finally make that `SHAutoComplete` call: First I have to find the edit control, which means finding the combo box and then asking the combo box for the interior edit control. (Fortunately, this is already called out in the documentation for `SHAuto-Complete`, so I didn't have to puzzle over it for long.) And second, I couldn't disable autocomplete directly in `WM_INITDIALOG` because that happens too early in the common file dialog initialization process. Instead, I post myself a message and do the "final initialization" later. (This I discovered by trial and error.)

And there you have it, a common dialog box with no autocomplete.

**Update**: Joylon Smith points out that the documentation for `SHAutoComplete` explicitly cautions against calling it more than once on the same window because it results in a memory leak.

That caution was written based on information I provided back in Windows XP. The memory leak was fixed in Windows Vista, but the documentation was not updated to match. So please mentally insert "On versions of Windows prior to Windows Vista (and versions of Windows Server prior to Windows Server 2008)" at the start of that paragraph. A doc change request has also been submitted, so hopefully the revised documentation will appear soon.

Raymond Chen

**Follow**