

How can I emulate the `REG_NOTIFY_THREAD_AGNOSTIC` flag on systems that don't support it? part 1

 devblogs.microsoft.com/oldnewthing/20201221-00

December 21, 2020



Raymond Chen

Windows 8 introduced the `REG_NOTIFY_THREAD_AGNOSTIC` flag to the `RegNotifyChangeKeyValue` function. But what if you need to run on systems older than that?

One of the recommended workarounds for the lack of support for the `REG_NOTIFY_THREAD_AGNOSTIC` flag is to do the registration from a persistent thread, or at least a thread that is guaranteed to exist for as long as you need the registration. Maybe your program already has such a persistent thread. In that case, you can marshal the request to that thread. For example, if the registration is made by a class whose lifetime is tied to some UI, then you can perform the registration on the UI thread.

```

struct RegNotifyChangeKeyValueAsyncArgs
{
    HKEY hkey;
    BOOL bWatchSubtree;
    DWORD dwNotifyFilter;
    HANDLE hEvent;
    LONG result;
};

#define WM_REGNOTIFYCHANGEKEYVALUEASYNC (WM_APP + somevalue)

LONG MyClass::RegNotifyChangeKeyValueAsync(
    HWND hwnd,
    HKEY hkey,
    BOOL bWatchSubtree,
    DWORD dwNotifyFilter,
    HANDLE hEvent)
{
    RegNotifyChangeKeyValueAsyncArgs args =
        { hkey, bWatchSubtree, dwNotifyFilter, hEvent,
          ERROR_INVALID_PARAMETER };
    SendMessage(m_hwnd, WM_REGNOTIFYCHANGEKEYVALUEASYNC,
                0, reinterpret_cast<LPARAM>(&args));
    return args.result;
}

// in the WndProc
case WM_REGNOTIFYCHANGEKEYVALUEASYNC:
{
    auto args = reinterpret_cast<
        RegNotifyChangeKeyValueAsyncArgs*>(lParam);
    args->result = RegNotifyChangeKeyValue(
        args->hkey,
        args->bWatchSubtree,
        args->dwNotifyFilter,
        args->hEvent,
        TRUE);
    return TRUE;
}

```

This version marshals the call to the UI thread. I'm assuming the purpose of this class is to implement some UI element, so the UI thread will exist for as long as the corresponding window exists.

Next time, we'll discuss what we can do if we don't have a UI thread or other persistent thread that we can rely upon to do our bidding.

Raymond Chen

Follow

