
Subject: Restoring TrayIcon control
Posted by [lucpolak](#) on Wed, 20 Apr 2011 07:53:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello, i have another problem with the tray icon.

When the explorer.exe crashes (oh yes it's happens), the trayicon is not correctly restored.

For example, here is my TrayIcon :

I kill explorer.exe (sorry for ms) and restart it and my trayIcon is missing

So the question is : Can I handle this ? Can i detect an explorer crash and restore the TrayIcon Ctrl ?

Thanks a lot

Lucas

File Attachments

- 1) [tray1.png](#), downloaded 771 times
 - 2) [tray2.png](#), downloaded 957 times
-

Subject: Re: Restoring TrayIcon control
Posted by [lucpolak](#) on Fri, 22 Apr 2011 07:56:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Up up

I found on the web that restoring the TrayIcon can be made by catching the WM_TASKBARCREATED message.

It can be done by declaring this in the WindowProc fct :

```
// Somewhere in the code or header ....
UINT WM_TASKBARCREATED = 0 ;

// In the WindowProc
if (message == WM_CREATE)
    WM_TASKBARCREATED = RegisterWindowMessageA("TaskbarCreated");

if (message == WM_TASKBARCREATED)
{
    // Restore the icon ... how can i do it ?
}
```

I've tried de call TrayIcon::Show() method but it crash with a Shell_NotifyIcon Exception because in the Notify Fonction (TrayIconWin32.cpp) we have this line :

```
if(visible) {
    nid.flags = NIF_ICON|NIF_MESSAGE|NIF_TIP;
    if(nid.icon)
        DestroyIcon(nid.icon);
    nid.icon = IconWin32(icon);
    String stip = ToSystemCharset(tip);
    int len = min(stip.GetLength(), 125);
    memcpy(nid.tip, stip, len);
    nid.tip[len] = 0;
    VERIFY(Shell_NotifyIcon(msg, (NOTIFYICONDATA *)&nid));
```

I change this to :

```
if(visible) {
    nid.flags = NIF_ICON|NIF_MESSAGE|NIF_TIP;
    if(nid.icon)
        DestroyIcon(nid.icon);
    nid.icon = IconWin32(icon);
    String stip = ToSystemCharset(tip);
    int len = min(stip.GetLength(), 125);
    memcpy(nid.tip, stip, len);
    nid.tip[len] = 0;
    while (Shell_NotifyIcon(msg, (NOTIFYICONDATA *)&nid) == FALSE)
    {
        Sleep(100);
    }
```

but it change nothing ...

Helps please

Subject: Re: Restoring TrayIcon control
Posted by [lucpolak](#) on Tue, 26 Apr 2011 14:25:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

nobody can helps me ?

I'm trying to understand the mecanism of UPP windows and it's not easy .

I'm find in the Win32Wnd.cpp the main WndProc witch enables me to initialize correctly my WM_TASKBARCREATED variable. This piece of code works fine and catch correctly the

explorer.exe crashes :

```
LRESULT CALLBACK Ctrl::WndProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    if (message == WM_CREATE)
    {
        WM_TASKBARCREATED = RegisterWindowMessageA("TaskbarCreated") ;
        ELOG("Ctrl::TASKBARCREATED = "<<WM_TASKBARCREATED);
    }

    if (message == WM_TASKBARCREATED)
    {
        ELOG("Ctrl::TASKBARCREATED");
        // How can i translate and recreate order to the TrayIcon class ?
    }
    [...]
}
```

But how can I recreate tray icon when the message WM_TASKBARCREATED is received ????

How can i Translate this message to the TrayIcon::WindowProc ????

HElps Helps helps

Lucas

Subject: Re: Restoring TrayIcon control [SOLVED]
Posted by [lucpolak](#) on Tue, 26 Apr 2011 15:20:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

After a long time, i've found the solution :

In CtrlCore, add a static variable to the class Ctrl to store the value of Windows Message TaskbarCreated :

```
[...]
#ifdef PLATFORM_WIN32
    static UINT WM_TASKBARCREATED;
#endif

    static LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam);
```

In Win32Wnd.cpp, initialize this static variable :

```
static bool sFinished;
```

```
UINT Ctrl::WM_TASKBARCREATED = 0;  
[...]
```

And in the WndProc function, Register the TaskbarCreated Message :

```
if (message == WM_CREATE)  
{  
    WM_TASKBARCREATED = RegisterWindowMessageA("TaskbarCreated") ;  
    ELOG("Ctrl::TASKBARCREATED = "<<WM_TASKBARCREATED);  
}
```

Finally, in the TrayIconWin32.cpp, restore the icon if needed :

```
LRESULT TrayIcon::WindowProc(UINT message, WPARAM wParam, LPARAM lParam)  
{  
    if (message == WM_TASKBARCREATED)  
    {  
        RLOG("TrayIcon::WM_TASKBARCREATED");  
        visible = false;  
        Show();  
    }  
    else if(message == UM_TASKBAR_)  
[...]
```

So, can you integrate it in the next revision of Upp ?

Thanks,

Lucas

Subject: Re: Restoring TrayIcon control [SOLVED]
Posted by [lucpolak](#) on Tue, 26 Apr 2011 16:15:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have another patch to do about the Shell_NotifyIcon function.

There are some problems with this function like discussed here :

[http://msdn.microsoft.com/en-us/library/bb762159\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/bb762159(v=vs.85).aspx)

or here :

http://issuetracker.delphi-jedi.org/bug_view_advanced_page.php?bug_id=3747

A solution can be to modify the TrayIcon::Notify function like this :

```
void TrayIcon::Notify(dword msg)
{
    if(visible) {
        nid.flags = NIF_ICON|NIF_MESSAGE|NIF_TIP;
        if(nid.icon)
            DestroyIcon(nid.icon);
        nid.icon = IconWin32(icon);
        String stip = ToSystemCharset(tip);
        int len = min(stip.GetLength(), 125);
        memcpy(nid.tip, stip, len);
        nid.tip[len] = 0;
        BOOL Status = Shell_NotifyIcon(msg, (NOTIFYICONDATA *)&nid);

        // To prevent from Shell_NotifyIcon bugs...
        // discussed here : http://msdn.microsoft.com/en-us/library/bb762159\(v=vs.85\).aspx
        // and here : http://issuetracker.delphi-jedi.org/bug\_view\_advanced\_page.php?bug\_id=3747
        if (Status == FALSE)
        {
            // The status of Shell_NotifyIcon is FALSE, in the case of NIM_ADD, we will try to Modify
            // If the modify is OK then we can consider that the add was worked.
            // Same, case with delete, we can try modify and if KO then we can consider that is icon
            // was deleted correctly. In other cases, we will retry after 100ms
            DWORD ErrorCode = GetLastError();
            if ( (ErrorCode == ERROR_SUCCESS) || (ErrorCode == ERROR_TIMEOUT) )
            {
                int retryCount = 0;
                BOOL retryResult;
                do
                {
                    Sleep(100);
                    if (msg == NIM_ADD) retryResult = Shell_NotifyIcon(NIM_MODIFY, (NOTIFYICONDATA
                    *)&nid);
                    else if (msg == NIM_DELETE) retryResult = !Shell_NotifyIcon(NIM_MODIFY,
                    (NOTIFYICONDATA *)&nid);
                    retryCount++;
                }while( (!retryResult) && (retryCount<50) );
            }
        }
    }
}
```

Thanks for patching.

Lucas

Subject: Re: Restoring TrayIcon control [SOLVED]
Posted by [mirek](#) on Wed, 27 Apr 2011 04:57:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

lucpolak wrote on Tue, 26 April 2011 11:20
So, can you integrate it in the next revision of Upp ?

Thanks,

Lucas

Yes! And sorry for delay. Good find.

Subject: Re: Restoring TrayIcon control [SOLVED]
Posted by [mirek](#) on Wed, 27 Apr 2011 05:14:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

lucpolak wrote on Tue, 26 April 2011 12:15I have another patch to do about the Shell_NotifyIcon function.

There are some problems with this function like discussed here :

[http://msdn.microsoft.com/en-us/library/bb762159\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/bb762159(v=vs.85).aspx)
or here :

http://issuetracker.delphi-jedi.org/bug_view_advanced_page.php?bug_id=3747

Applied in altered form (what if notify is NIM_MODIFY?

Mirek

Subject: Re: Restoring TrayIcon control [SOLVED]
Posted by [lucpolak](#) on Thu, 28 Apr 2011 06:59:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 27 April 2011 07:14

Applied in altered form (what if notify is NIM_MODIFY?

Mirek

It's right

Thanks a lot for patching. ^^
