

RE: IE Hosted UserControl – javascript interaction not working.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.windowsforms/2006-09/msg00332.h>

- *From:* sunev <sunevnuahs@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 19 Sep 2006 02:21:02 -0700
-

Linda,

Thanks very much for your response – I'd found the ComSourceInterfaces attribute and that seemed to get things going for me.

Just got to work out the security side of things now and we'll be done – should be OK as I've had things working with file system and web access, but needx to tighten it down as much as possible.

For any one interested my final code is:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Text;
using System.Windows.Forms;

using System.Runtime.InteropServices;
using System.Security.Permissions;

[assembly: ClassInterface(ClassInterfaceType.AutoDual)]
[assembly: SecurityPermission(SecurityAction.RequestMinimum, UnmanagedCode = true)]
namespace MyControls
{
    public delegate void SubmitClickedHandler();

    [ComVisible(true)]
    [Guid("67CD00DB-E9BC-484d-9033-83D41112EBD4")]
    public interface HostNameControlCOMIncoming
    {
        string HostName { get; set; }
        void GetHostName();
    }
}
```

RE: IE Hosted UserControl – javascript interaction not working.

```
[ComVisible(true)]
[Guid("61CD00DB-E9BC-484d-9033-83D41112EBD4")]
[InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]
public interface HostNameControlEvents
{
    [DispId(0)]
    void GetHostNameComplete();
}

[ComVisible(true)]
[ClassInterface(ClassInterfaceType.None)]
[ComSourceInterfaces(typeof(HostNameControlEvents))]
[ComDefaultInterface(typeof(HostNameControlCOMIncoming))]
public partial class HostNameControl : UserControl,
    HostNameControlCOMIncoming
{

    private string _hostname;

    public HostNameControl()
    {
        InitializeComponent();
    }

    [Category("Action")]
    [Description("Fired when the Submit button is clicked.")]
    public event SubmitClickedHandler GetHostNameComplete;

    protected virtual void OnGetHostNameComplete()
    {
        if (GetHostNameComplete != null)
            GetHostNameComplete();
    }

    [Category("Appearance")]
    [Description("Gets or sets the name in the text box")]
    public string HostName
    {
        get { return _hostname; }
        set { _hostname = value; }
    }

    [Description("Gets hostname of local machine")]
    public void GetHostName()
    {
        _hostname = System.Net.Dns.GetHostName();
        OnGetHostNameComplete();
    }

}

}
```

RE: IE Hosted UserControl – javascript interaction not working.

}

Shaun Venus

"Linda Liu [MSFT]" wrote:

Hi Shaun,

To access public properties or invoke public methods of the control from javascript in the web page, all we need to do is to build the control as a COM class. To do this, right-click the project in the Solution Explorer and choose Properties. In the Project Designer, select the Application tab and click the 'Assembly Information' button on the right panel. In the Assembly Information window, select the checkbox before the 'Make assembly COM-Visible' option and press OK.

To expose events in the control to web pages and handle the events in the web pages, we need to do more work. The following is the walkthrough.

1. Pack the events to be exposed into an interface, which is applied the InterfaceTypeAttribute and then apply this interface to the control class. The following is a sample.

```
[InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]  
public interface IMyControlCOMEvents  
{  
    [DispId(0x60020000)]  
    void HostIdentificationComplete();  
}
```

```
[ComSourceInterfaces(typeof(IMyControlCOMEvents))]  
public partial class MyControl2 : UserControl  
{  
    .....  
}
```

2. Because we need to handle the event in the script in the web page, we need assign the control assembly embeded in the web page the permission to call unmanaged code. To do this, follow the steps below.

a. Type 'mscorcfg.msc' in the SDK command to open .Net Framework 2.0 Configuration.

b. In the configuration, navigate to My Computer->Runtime Security Policy->Machine->Permission Sets. Right-click on the LocalIntranet sub node and choose 'Duplicate'. A new permission set called 'Copy of

RE: IE Hosted UserControl – javascript interaction not working.

LocalIntranet' is generated.

c. Right-click on the new permission set and choose 'Change Permissions'. In the Create Permission Set window, double-click the 'Security' item in the right listbox. In the 'Permission Settings' window, select the checkbox before the 'Allow calls to unmanaged assembly' option. Press OK and then Finish button to close the windows.

d. Navigate to My Computer->Runtime Security Policy->Machine->Code Groups. Right-click on the 'All_Code' node and choose 'New'. In the 'Create Code Group' window, select the 'Create a new code group' option and type a name for the new group and press Next. In the 'Choose the condition type for this code group' combobox, select 'URL' and type the web site url into the URL textbox, e.g. <http://localhost/mytest/> and press Next. Choose 'Use existing permission set' option and select 'Copy of LocalIntranet' in the combobox and press Next. Press Finish.

3. Because the assembly that load the assembly of our control into the web page doesn't have the permission to call the unmanaged code, we must assign this permission to the assembly. To do this, we call the Assert method of the Securitypermission class. The lines of code you wrote in the DoHostIndenticationComplete method are correct.

Please follow the above steps to solve your problem and let me know the result.

Sincerely,
Linda Liu
Microsoft Online Community Support