

Re: Accessing Directory Services from a SharePoint Web Part

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2004-10/0255.html>

From: Joe Kaplan \((MVP - ADSI)\) (joseph.e.kaplan_at_remove_this.accenture.com)

Date: 10/19/04

Date: Tue, 19 Oct 2004 13:50:29 -0500

Inline below:

"Jondis" <jondis@bellatlantic.net> wrote in message
news:D8D46148-C49A-4ED4-B3A0-00F2B2CC4B47@microsoft.com...

> *Joe,*

>

> *Thanks for the suggestions... I will try encapsulating my Active Directory
> code into a single DLL that I install into the GAC.*

>

> *You've raised two additional questions:*

> *1. How do I mark my assembly in the GAC as*

> *'AllowPartiallyTrustedCallers'?*

in your assemblyinfo file, you add:

```
<assembly: AllowPartiallyTrustedCallers(>
```

or

```
[assembly: AllowPartiallyTrustedCallers()]
```

in C#

> *2. Are the changes I've already made to SharePoint's security*

> *configuration*

> *below sufficient to allow my Web Part to call my new AD Assembly in the*

> *GAC?*

> *What else would I need to do?*

I think your perm set is correct, but test to be sure! If you do the Assert thing that I mentioned, you won't need the DirectoryServicesPermission, but it won't hurt. Doing the Assert is probably not a good idea anyway, especially if you can grant the required permission to the code.

> *3. Lastly, has anyone release a commercial product that manipulates*

> *Active*

> *Directory either as a Web Part, ASP.NET Control or other .NET*

> *Implementation?*

I have no idea on this. A lot of people are still struggling with CAS, so even if there was, it might not have its CAS story straight either.

I'm not sure if S.DS will have the APTCA in .NET 2.0 or not. I've discussed it with the product team, but it is a potentially big security risk for MS and not something they would consider lightly.

HTH,

Joe K.

>
> *Thanks again for your help,*
> *JD*
>
> *"Joe Kaplan (MVP – ADSI)" wrote:*
>
>> *S.DS can only be called from Full Trust. It does not have the*
>> *AllowPartiallyTrustedCallersAttribute set on the assembly.*
>>
>> *There are a couple of ways you might go about this. One way is to allow*
>> *Full Trust. However, you might not want to do that for security reasons.*
>>
>> *You might create a component that does all of your S.DS logic and put it*
>> *in*
>> *the GAC. Assemblies in the GAC always have full trust. Make sure it has*
>> *AllowPartiallyTrustedCallers set on it.*
>>
>> *Then, you'll need to either make sure upstream callers have the*
>> *appropriate*
>> *DirectoryServicesPermission OR you can Assert that permission in your*
>> *code*
>> *so that the stack walk is interrupted. Note that you are creating a*
>> *potential security issue if you assert permissions, so you have to be*
>> *sure*
>> *you know what you are doing before taking on that responsibility.*
>>
>> *Note that you would have this problem with any assembly that doesn't*
>> *allow*
>> *partially trusted callers.*
>>
>> *Joe K.*
>>
>> *"Jondis" <jondis@bellatlantic.net> wrote in message*
>> *news:78C3AABD-21F1-4851-931D-B74B10E482C9@microsoft.com...*
>> *>I get a vague 'Security Error' when I try to access Active Directory via*
>> *>the*
>> *> 'System.DirectoryServices' DLL. I am doing this within a SharePoint*
>> *> Web*
>> *> Part. I also stripped the code out of the SharePoint Web Part and put*
>> *> it*

>>> on
>>> a regular ASP.NET Form and it worked fine.
>>>
>>> The error message is literally "Security Error". The stack trace
>>> returns
>>> the line number of the sub-routine that calls into the sub-routine that
>>> actually touches Active Directory (System.DirectoryServices)... so I
>>> guess
>>> Microsoft's "Code Access Security" really works!!!
>>>
>>> I think I need help with my trust file. I started with the WSS_Minimal
>>> trust file (that ships with SPS 2003) and I've gradually added
>>> permissions
>>> into it (such as SQL Permission).
>>>
>>> I added this to <SecurityClasses>:
>>> <SecurityClasses>
>>> <SecurityClass Name="DirectoryServicesPermission"
>>> Description="System.DirectoryServices.DirectoryServicesPermission,
>>> System.DirectoryServices, Version=1.0.5000.0, Culture=neutral,
>>> PublicKeyToken=b03f5f7f11d50a3a"/>
>>> </SecurityClasses>
>>>
>>> Then I added the "DirectoryServicesPermission" to the
>>> "NamedPermissionSet"
>>> of "ASP.NET":
>>> <PermissionSet class="NamedPermissionSet" version="1"
>>> Name="ASP.Net">
>>> <IPermission class="AspNetHostingPermission" version="1"
>>> Level="Medium" />
>>> <IPermission class="DirectoryServicesPermission"
>>> version="1"
>>> Unrestricted="true"/>
>>> <IPermission class="SecurityPermission" version="1"
>>> Flags="Execution" />
>>> <IPermission class="SharePointPermission" version="1"
>>> ObjectModel="True" />
>>> <IPermission class="WebPartPermission" version="1"
>>> Connections="True" />
>>> </PermissionSet>
>>>
>>> I haven't modified anything in the <CodeGroup> section... my
>>> understanding
>>> is that ASP.NET is the first CodeGroup that my Web Part should match --
>>> so
>>> it
>>> should run under that code group.
>>>
>>> The changes I've made have at least allowed my Web Part to execute and
>>> return the Security Error... previously I would get the "Web Part
>>> Maintenance Page" and have to remove my Web Part, change some code and

```
>>> try
>>> again (glad I got thru that!).
>>>
>>> My AD code is very simple right now... I'll post it ... but I'm
>>> reasonably
>>> certain CAS is not letting me anywhere near it anyway:
>>>
>>> Public Class PortalUser
>>> Private Const m_User As String = "User"
>>>
>>> Private _SPUser As Microsoft.SharePoint.SPUser
>>> Private _ADUser, _ADHelper As
>>> System.DirectoryServices.DirectoryEntry
>>>
>>> Public Property SPUser() As Microsoft.SharePoint.SPUser
>>> Get
>>> Return _SPUser
>>> End Get
>>> Set(ByVal Value As Microsoft.SharePoint.SPUser)
>>> _SPUser = Value
>>> End Set
>>> End Property
>>> Public Property ADUser() As
>>> System.DirectoryServices.DirectoryEntry
>>> Get
>>> Return _ADUser
>>> End Get
>>> Set(ByVal Value As System.DirectoryServices.DirectoryEntry)
>>> _ADUser = Value
>>> End Set
>>> End Property
>>>
>>> Public Sub New()
>>>
>>> End Sub
>>>
>>> Public Sub GetADUser()
>>> If (_SPUser.LoginName > [String].Empty) Then
>>> _ADHelper = New
>>> System.DirectoryServices.DirectoryEntry("WinNT://MyDomain",
>>> "PowerfulUserName", "PowerfulUserNamePassword")
>>> _ADUser = _ADHelper.Children.Find(_SPUser.LoginName,
>>> m_User)
>>> End If
>>> End Sub
>>> End Class
>>>
>>> Please let me know any recommendations for modifying the Trust file
>>> additionally, changing other security files related to SharePoint, etc.
>>> If
>>> there's something I'm missing in my code, please let me know as well.
```

microsoft.public.dotnet.security: Re: Accessing Directory Services from a SharePoint Web Part

>> >

>> > *Thanks,*

>> > *JD*

>> >

>> >

>> >

>>

>>

>>

>>