

RE: Windows Service cannot create text files?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2005-01/2589.html>

From: Joseph MCAD (*JosephMCAD_at_discussions.microsoft.com*)

Date: 01/14/05

Date: Fri, 14 Jan 2005 08:39:05 -0800

Jan. 14, 2005

I'm sure my users wouldn't like that either! :) I just thought you should know that running your service as the Local System account is Very Risky! If your service has a single security flaw that is exploited, then the attacker could do Anything to your system! This one is not a joke! I'm taking one of the Microsoft Certified Professional exams on security and this is never recommended. It is possible to grant your assembly the FileIO permission in code instead of doing it manually with the instructions I gave you. (I didn't think of this until you said that changes would have to be done in code.) You use the System.Security.Policy namespace. This contains the classes that modify the Machine, User, etc. policies. You would still want to strong name your assembly before you deploy it. Then after the user installs the assembly, he/she would then run a .msi file that calls your code which changes the policy. The steps for changing the policy for the user would be just like installing an application. I don't know how to change the policy in code, but I do know that you use that namespace and that you put your code under a certain method in a class that inherits from Installer. Then in your deployment project's custom actions you would have it run this class. I hope you can find some easy examples on how to do this! :) Hope this helps!

Joseph MCAD

"Amjad" wrote:

- > *Thanks for the reply.. but this is definetly not what I'm looking for. If any*
- > *settings have to be changed on the "Local System" then they've to be changed*
- > *programatically from within the Service. Imagine deploying this Service for a*
- > *client with a lengthy sheet of instructions for his administrator to work on!*
- >
- > *Apparently, the problem was much simpler than that. In the "Service Process*
- > *Installer" properties, I changed the Account value from "Local Service" to*
- > *"Local System" and that did the trick!*
- >

>
> *On another note, this Windows Service doesn't start automatically! I had to*
> *go to Control Panel --> Admin Tools --> Services, select the Service and then*
> *click on Start. Mind that "Startup Type" in "Services" for my service is set*
> *to "Automatic".*
> *Do you know how to make the service start with Windows?*
>
> *Amjad*
>
> *"Joseph MCAD" wrote:*
>
>
> *Jan. 13, 2005*
>
> *Because you are unauthorized! :) That was a joke. :) What you could do*
> *is create a new seperate folder and then grant the account that your service*
> *runs under the rights to read/write in that folder. You would right click the*
> *folder and on the security tab add the account. If you can't do this or you*
> *want to grant a specific path then it might get a bit more complicated... :(*
> *You would want to strong name your assembly... at the VS command prompt*
> *switch to the application folder and type Sn.exe -k MyKey.pvk ... This will*
> *create the private key necessary to sign the assembly. Then open your project*
> *and go to the AssemblyInfo... Add:*
>
> *<Assembly: AssemblyKeyFile("c:\...Path Must Be Absolute To Work")>*
>
> *Then goto your control panel and (depending on your computer this part might*
> *be slightly different) goto the administrative tools and run the .Net 1.1*
> *Configuration tool. Then expand Runtime Security Policy and expand the node*
> *of the scope you want. I would suggest the Machine if your service does not*
> *run under a user account. Then expand Code Groups and then right click*
> *All_Code and click New. (Keep in mind that this will modify your computer's*
> *settings.) What you are going to do is create a code group that only*
> *assemblies with your strong name are allowed in. This group will grant the*
> *filesystem privilage that is specific to a path. Type in a name for the group*
> *such as FilePrivForMyService and a description if desired. Then click next*
> *and then select Strong Name from the drop down box. Then click browse and*
> *select your assembly. This will import the strong name and only assemblies*
> *with this strong name are qualified to be in this group. Then click the Name*
> *and Version checkboxes if you only want assemblies with the exact name and*
> *version to be allowed in this group too. Then click next and select Create A*
> *New Permission set. Then type a name for it (this set will only contain the*
> *FileIO permission for the path you specify). On the next page double click*
> *the File IO or click it and select properties. Then make sure the button for*
> *granting assemblies specific paths is selected. Then under file path enter*
> *your path such as c:\ or c:\myservice\ . Then select (at least the write for*
> *your case) the read/write/append/path disc. permissions that you need. Then*
> *click on OK. Then click next and click finish! That was long! :) Quite a lot*
> *of work to just write to a file! This needs some improvement! I hope this*
> *solves your question and good night!*
>
>

>>
>> *Joseph MCAD*
>>
>>
>>
>> *"Amjad" wrote:*
>>
>>> *Hi,*
>>> *I just wrote a test Windows Service that creates a text file on startup*
>>> *(please see my code below). The file is never created.*
>>>
>>> *Protected Overrides Sub OnStart(ByVal args() As String)*
>>> *Dim swLog As StreamWriter = File.CreateText("C:\myLog.txt")*
>>> *swLog.WriteLine("My Windows Service has just started.")*
>>> *swLog.Close() : swLog.Flush()*
>>> *End Sub*
>>>
>>>
>>> *I checked the Event Viewer in the Control Panel and I found the following*
>>> *error associated to my Windows Service:*
>>>
>>> *"Service cannot be started. System.UnauthorizedAccessException: Access to*
>>> *the path "C:\myLog.txt" is denied.*
>>> *at System.IO.__Error.WinIOError(Int32 errorCode, String str)*
>>> *at System.IO.FileStrea ... etc."*
>>>
>>>
>>> *Does anyone know why my Windows Service is unable to create a simple text*
>>> *file?*
>>>