

## Re: Solution: Asp.Net and Smb shares – without impersonation

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-07/0852.html>

---

**From:** Granger Godbold ([granger.godbold\\_at\\_invalid.prometric.com](mailto:granger.godbold_at_invalid.prometric.com))

**Date:** 07/03/04

Date: Sat, 03 Jul 2004 10:07:20 -0600

No. You don't. I didn't have to change anything on the ASPNET account for it to work correctly.

Ken Cox [Microsoft MVP] wrote:

> *Don't you have to give the ASPNET account broad permissions to execute*  
> *"net use"? It seems like that's a pretty big security hole.*

>

> *I'd still prefer to use impersonation to authenticate to the network share.*

>

> *"Granger Godbold" <[granger.godbold@invalid.prometric.com](mailto:granger.godbold@invalid.prometric.com)> wrote in*  
> *message news:u0xJs8IYEHA.1684@tk2msftngp13.phx.gbl...*

>

>> *I've found a solution to this that I think I like, but I thought it*  
>> *wise to put it out for all to see so people could punch holes in it as*  
>> *they wished. (Is there a better way?)*

>>

>> *I want a page in an Asp.Net site to be able to open/access a file*  
>> *that's on an Smb share (ie. "Windows Share", "NetBios Share", etc.). I*  
>> *cannot use the "ASPNET" username; that's a bad route to take anyhow*  
>> *(got to mess with the "automatic" password settings and other worms in*  
>> *the can you'll be opening).*

>>

>> *At first glance, everyone seems to suggest using the "identity" tag in*  
>> *web.config to do impersonation. However, it's a huge PITA; there's*  
>> *more to it than what's described in KB #317012 or at the following*  
>> *<http://msdn.microsoft.com/library/en-us/secmod/html/secmod15.asp>*  
>> *For example, your new user won't have permission to access*  
>> *System.Diagnostics.Process.GetCurrentProcess().Handle*  
>> *among other various "gotchas". Impersonation is overkill.*

>>

>>

>>

>> *The solution?*

>> *Use System.Diagnostics.Process and run "net use" from the Asp.Net*  
>> *application to authenticate to your network share. The authentication*

>> *will last just like it does when you call it from the command line*  
>> *(for the user you execute it as).*  
>>  
>> *I believe this method to be the least invasive on existing code. For*  
>> *me, I have a File.Exists check first. If that fails, then I try to*  
>> *open the file (File.OpenText is what I tested with, but File.Open*  
>> *should be the same). If the exception is*  
>>  
>> *[System.IO.IOException] Logon failure: unknown user name or bad password.*  
>>  
>> *then I do the "net use" call to authenticate and try again. If it's*  
>> *not that exception, then the file will actually not exist, and you get*  
>> *this error:*  
>>  
>> *[System.IO.FileNotFoundException] Could not find file "<filename>"*  
>>  
>> *and I then let the exception propagate to my error-handling stuff.*  
>  
>