

Re: userName="machine" didn't work

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-01/msg03642.html>

- *From:* "Juan T. Llibre" <nomailreplies@xxxxxxxxxxxxx>
 - *Date:* Mon, 29 Jan 2007 17:36:49 -0400
-

re:

That's contrary to what Microsoft support told me.
They said that the .210 is higher than .42, and that indicates a later build.

Basic math review, Joe :

Which number is **larger**: .210 or .42 ?

Either the guy who told you that is redefining math,
or he was trying to pull a fast one on you.

Believe me, **nobody** plays around with build numbers.
Not Microsoft, not Sun, not Apple, and not **anybody** in the software business.

Simply put: higher numbers are later builds, not the other way around.

Plus: why would the update not update **all** build numbers?
Remember you had (have?) differing build numbers?

Do you still have them ?

Juan T. Llibre, asp.net MVP
asp.net faq : <http://asp.net.do/faq/>
foros de asp.net, en español : <http://asp.net.do/foros/>

=====
"Joe Rattz" <JoeRattz@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:E1DADF5A-2906-4008-8C74-3AB56A03E1F6@xxxxxxxxxxxxxxxxxxxxx

They might have been "newer", although I don't know how
that could be,
but they belong to an earlier build than the release version.

Re: userName="machine" didn't work

That's contrary to what Microsoft support told me. They said that the .210 is higher than .42, and that indicates a later build.

I can't find anything that defines how MS versioning works...that ones tough to google on.

Thanks.

"Juan T. Llibre" wrote:

Good to know you were able to find a solution, Joe.

re:

you hinted at it enough that I think you think that the .210 dlls are older (a previous version) than the .42 dlls. That is not the case. The .210 dlls I had for ASP.NET are newer than the .42 dlls.

They might have been "newer", although I don't know how that could be, but they belong to an earlier build than the release version.

Maybe the date on your .210 dll's refers to the *download* date, rather than the build date ?

re:

It's a little unclear to me exactly which part of this really resolved the problem

Probably deleting the machinename\ASPNET account.

Running "aspnet_regiis -i" recreates the account if it doesn't find it, and assigns the needed permissions.

Juan T. Llibre, asp.net MVP
asp.net faq : <http://asp.net.do/faq/>
foros de asp.net, en español : <http://asp.net.do/foros/>

=====
"Joe Rattz" <JoeRattz@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:C096266D-B4DB-40DA-9C9C-5EE7F6FCB753@xxxxxxxxxxxxxxxxxxxx

Re: userName="machine" didn't work

Juan, thanks for trying to help. You never quite said it, but you hinted at it enough that I think you think that the .210 dlls are older (a previous version) than the .42 dlls. That is not the case. The .210 dlls I had for ASP.NET are newer than the .42 dlls.

Ok, I have a solution that worked for me courtesy of Microsoft support.

The problem had to do with the ASPNET user, and its permissions, and apparently some bug in the ASP.NET install. I had tried all the things in these threads:

<http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=116989&SiteID=1>
<http://blog.devstone.com/Aaron/archive/2006/05/08/1586.aspx>

But none of them worked for me. Here is what worked for my box:

We followed below steps to resolve the issue –

1. Unregistered all the versions of ASP.NET with command "C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -ua".
2. Deleted the ASPNET account from "Local Users and Group – Users".
3. Then registered ASP.NET 2.0 with IIS using "C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -i".
4. Gave permissions to the ASPNET account using "C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -ga machinename\ASPNET".
5. Reset the IIS and that resolved the issue for ASP.NET 2.0.
6. We registered ASP.NET 1.1 with IIS as well using command "C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\aspnet_regiis -i".
7. Reset the IIS.

Re: userName="machine" didn't work

After following the above steps we configured virtual directory versions as required and test both ASP.NET 1.1 and ASP.NET 2.0 application.

Everything worked fine as expected.

CAUSE:

=====

This problem occurs because the process model user account (in our case, ASPNET) does not have the correct user rights for the required system files and folders to run the ASP.NET worker process.

It's a little unclear to me exactly which part of this really resolved the problem because I had already done lots of these steps already without the problem being solved.. The problem was not resolved until we granted access to machine\ASPNET:

```
aspnet_regiis -ga [machinename]\ASPNET
```

But, I believe I had already tried that before and it didn't fix the problem, but I couldn't swear that I did.

I mention the machine name specifically because I had tried to grant access to ASPNET (no machine name) and while this didn't generate an error message, it also didn't work. And that's odd because if I try to grant access to a made up account name, it does give an error message. So somehow, granting access to ASPNET doesn't generate an error that it is a non-existent account, but it fails to grant access to the machinename\ASPNET account. Odd. Even odder is the number of people in those two referenced threads that granting access to ASPNET without specifying the machine name worked for. It apparently works some times without the machine name, but it didn't for me.

Re: userName="machine" didn't work

I bring this up because it may have been necessary that we deleted the ASPNET user and allowed the ASP.NET install to re-create it. If I were trying to resolve this problem now, I would try to grant access and specify the machine name before I uninstalled everything just to see if that fixes the problem.

There apparently is a problem where if the ASPNET user already exists when ASP.NET 2.0 is installed, under some (unknown) circumstances, the ASPNET user doesn't get its permissions updated properly. Obviously this doesn't happen the majority of the time. There are 3 other .NET developers here all running similar systems (same hardware, same versions of software, and it didn't happen to them. Just my luck!

"Juan T. Llibre" wrote:

Thanks for seeing this through, Joe.

re:

I researched to find out what changed the version of some of my dlls. It is a security patch. You can read about it here:

<http://support.microsoft.com/kb/922770>
– KB922770

Interesting.

They must not have realized that the patch would run the version number backwards. That's usually a no-no, precisely due to often-appearing compatibility problems.

re:

Re: userName="machine" didn't work

Since you don't have DLLs with different versions, you might want to install that security update.

I don't think I need to. I'm running the W2K3 SP2 Release Candidate, which apparently fixes the problem while maintaining the same version number, since that patch doesn't appear on my lists of critical patches when I go to Windows/Microsoft Update.

The release version of SP2 should not be long in appearing. I suggest that as soon as it comes out you apply it to all your W2K3 boxes.

You know, I find it odd that you haven't posted the operating systems in which this problem is occurring (and I haven't asked). I should have asked you way earlier, since I had assumed you were running W2K3, but now think that you're running Windows XP in the development boxes where the problem is occurring.

This might make a difference.

re:

This still leaves me though with the exact same problem I originally had. Any ideas?

You say that if you leave SYSTEM in as the process model account, everything works, right ?

Well, working with SYSTEM as the account which runs ASP.NET isn't too bad, but there *are* some directories which the account which runs ASP.NET shouldn't have access to, in order to stay within strict security guidelines.

Re: userName="machine" didn't work

There's two more things you could do to solve the problem.

1. Create a custom service account to run ASP.NET as.

Please review this article :
"Configuring ASP.NET Process Identity" :
<http://msdn2.microsoft.com/en-us/library/dwc1xthy.aspx>

....and make sure that the account you create/use to run ASP.NET as has the permissions outlined in this article:

"ASP.NET Required Access Control Lists (ACLs)"
<http://msdn2.microsoft.com/en-us/library/kwzs111e.aspx>

Also, review the guidelines in this article :
"How To: Create a Service Account for an ASP.NET 2.0 Application":
<http://msdn2.microsoft.com/en-us/library/ms998297.aspx>

You can ignore the bit about creating an Application Pool if you're not running IIS 6.0 (if you're not running W2K3).

The rest is valid if you're using Windows XP with an NTFS-formatted partition.

2. Impersonate the identity ASP.NET runs as for an individual application.

Review this article :
"ASP.NET Impersonation"
<http://msdn2.microsoft.com/en-us/library/xh507fc5.aspx>

The main thrust is configuring an identity in web.config of your application, instead of configuring a process account (which runs all your apps as the configured account).

```
<identity impersonate="true"  
userName="Domain\User"  
password="s@omePassword!" />
```

That would solve the problem for a single application

Re: userName="machine" didn't work

(whose web.config you modify to reflect a user identity).

You could leave SYSTEM as the process model identity, since the impersonation would replace that identity in the specified app.

In this case, you would **also** need to assign the necessary permissions, to the directories listed in the "ASP.NET Required Access Control Lists (ACLs)" article, to whichever account you choose to impersonate ASP.NET as.

Those two alternatives are your only choices at this point, until such time as a new patch can be released by Microsoft which will get rid of the different versioning created by the KB922770 patch.

Finally, make sure that the newly-configured application identity, or the newly-configured process model identity are actually running ASP.NET.

Copy this little file I wrote and run it after you've made the identity changes:

identity.aspx:

```
-----  
<% @ Page Language="VB" %>  
<% @ Import Namespace =  
System.Security.Principal %>  
<script runat="server">  
Sub Page_Load()  
Dim tmp As String =  
WindowsIdentity.GetCurrent.Name()  
Label1.Text = "ASP.NET is running as the  
account : " & tmp  
End Sub  
</script>  
<html>  
<head>  
<title>What account is ASP.NET running as  
</title>  
</head>  
<body>  
<form id="form1" runat="server">  
<div>
```

Re: userName="machine" didn't work

```
<asp:Label ID="Label1" Runat="server"
Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

That will confirm, or not, that the identity you configured was successfully configured.

Please post back what you decided to do and how you did, OK?
Either method outlined above, in 1. and 2., will work.

If you have any doubts or further questions, please post them.

Good luck!

Juan T. Llibre, asp.net MVP
asp.net faq : <http://asp.net.do/faq/>
foros de asp.net, en español :
<http://asp.net.do/foros/>

=====
"Joe Rattz"
<JoeRattz@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message
news:70A71306-7C44-41D1-A92E-1493187FD59D@xxxxxxxxxxxxxxxxxxxx

Ok, here is a rundown of what I did and the results, just for the record. I uninstalled BLINQ, and tested, but the DLL versions still didn't match, and the error still happened. Then I uninstalled LINQ, and the versions still didn't match, and the error still happened. I then uninstalled the .NET Framework 2.0. I then deleted the v2.XXXX directory for safe measure. I installed the .NET Framework 2.0 from the

Re: userName="machine" didn't work

link you recommended. The DLL versions did match at that point, but the error still happened. But, at least the error message had matching versions. Also, immediately after the install, the processModel section of my machine.config file looked just like I showed you mine looked like before:

```
<processModel  
autoConfig="true"/>
```

I changed it to:

```
<processModel  
userName="machine"  
password="AutoGenerate"  
autoConfig="true"/>
```

But it still didn't work. Remember though that if I change the userName to System and leave out the password attribute, it does work.

```
<processModel  
userName="System"  
autoConfig="true"/> <- this  
works just fine.
```

Then I left for the day.

When I came in this morning, I tested again to send you the new error message to show you the versions matched, but this morning, they no longer match. So, I researched to find out what changed the version of some of my dlls. It is a security patch. You can read about it here:

Re: userName="machine" didn't work

<http://support.microsoft.com/kb/922770>
– KB922770

That page has a link on it to
here:

<http://www.microsoft.com/technet/security/bulletin/ms06-056.msp>

On that page, they show the exact files with the exact versions that I have, the ones ending in .210. I guess its not so strange for the versions to not match after all. Since you don't have DLLs with different versions, you might want to install that security update.

I also ran "aspnet_regiis -ga ASPNET" just to be sure, but it still didn't fix the problem.