

## Re: Setting a password on an AD account...

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-02/msg03075.html>

---

- *From:* Joe <Joe@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
  - *Date:* Thu, 22 Feb 2007 08:56:12 -0800
- 

Willy,

I misunderstood the question. The development box is not a part of the domain, but when I run it on the webserver within the domain the error is: Access is denied.

The account has full admin access.

I just want to verify that the code I am using does not have an error in it that I am not seeing.

Thanks.

See inline...

Willy.

"Joe" <Joe@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message <news:3D246C50-5087-4880-A287-D9DBAE31156D@xxxxxxxxxxxxxxxxxxxx>

Willy,

This is a Web App as I stated initially. The user does get created but is disabled. No problems there.

Right, but this doesn't tell me about the "security context" of the web application. Anyhow, I assume it's running in a restricted account (network service or aspnet) right? You don't use SSL to bind, and as this runs from a server which is not a domain member (a BAD thing if you ask me), Kerberos cannot be used to pass the password in a secured way either.

That means that "SetPassword" will try Win32 API "NetUserSetInfo" to change the users password. Now, this one fails when the current user is not an administrator on the DC. So I guess it works from a windows application started from a session which runs with administrative privileges on the DC.

## Re: Setting a password on an AD account...

One solution is to use SSL with server certificates, or delegate the AD access stuff to a COM+ server style application which runs as a Domain administrator, note that the latter will need to be a "shadow account" as you are running this on a non-domain member.

```
DirectoryEntry parent = new DirectoryEntry(  
"LDAP://dc.mydomain.local/OU=MyOU,DC=mydomain,DC=local,  
szUsername,  
szPassword,  
AuthenticationTypes.Secure);  
  
DirectoryEntry user = parent.Children.Add("CN=" + szFName + " " +  
szLName,  
"user");  
  
using(user)  
{  
...Set properties...  
user.CommitChanges();  
AdsUserFlags newValue = AdsUserFlags.NormalAccount;  
user.Properties["userAccountControl"].Value = newValue;  
user.Invoke("SetPassword", new object[] {szPassword});  
user.CommitChanges();  
user.Dispose();  
parent.Dispose();  
}
```

The client is NOT a member of the domain. This code did work when it was a Windows App.

Here are the errors:

ex.Message: Reason: Exception has been thrown by the target of an invocation.

ex.InnerException.Message: The network path was not found.

Weird thing, this InnerException....

Willy.