

# RE: Multiple Membership providers and AddUserToRole

---

*Source:*

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2006-09/msg00057.html>

---

- *From:* MrGrundh <[MrGrundh@xxxxxxxxxxxxxx](mailto:MrGrundh@xxxxxxxxxxxxxx)>
  - *Date:* Fri, 15 Sep 2006 02:10:02 -0700
- 

Hi mr Cheng!

The suggested solution will not work because the method Roles.AddUserToRole only accepts username and role as input and not users ID (guid)

```
Roles.AddUserToRole(CreateUserWizard1.UserName, "Friends")
```

We want to ONE roleprovider and multiple membershipproviders. The problem is when we have the same username in different membershipproviders and want to add the user to a role.

Scenario:

-A User (admin) logges in to membershipprovider1 and creates user "bjorn" and add him to role "Friends". That works alright.

-A User (admin) logges in to membershipprovider2 and creates user "bjorn" and add him to role "Friends". You got an error that says "The user 'bjorn' is already in role 'Friends!'".

Web.Config RoleProvider:

```
<roleManager defaultProvider="MyAspNetSqlRoleProvider"
enabled="true"
cacheRolesInCookie="true"
cookieName=".ASPROLES"
cookieTimeout="30"
cookiePath="/"
cookieRequireSSL="false"
cookieSlidingExpiration="true"
cookieProtection="All">
<providers>
<clear/>
<add name="MyAspNetSqlRoleProvider"
connectionStringName="LocalSqlServer"
applicationName="/MyRoles"
type="System.Web.Security.SqlRoleProvider"/>
```

## RE: Multiple Membership providers and AddUserToRole

```
</providers>  
</roleManager>
```

Web.Config MembershipProvider:

```
<membership>  
<providers>  
<clear />  
<add connectionStringName="LocalSqlServer"  
enablePasswordRetrieval="false"  
enablePasswordReset="true" requiresQuestionAndAnswer="true"  
applicationName="/" requiresUniqueEmail="false"  
passwordFormat="Hashed"  
maxInvalidPasswordAttempts="5" minRequiredPasswordLength="5"  
minRequiredNonalphanumericCharacters="0"  
passwordAttemptWindow="10"  
passwordStrengthRegularExpression=""  
name="AspNetSqlMembershipProvider"  
type="System.Web.Security.SqlMembershipProvider" />  
<add connectionStringName="LocalSqlServer"  
enablePasswordRetrieval="false"  
enablePasswordReset="true" requiresQuestionAndAnswer="true"  
applicationName="/MoreFriends" requiresUniqueEmail="false"  
passwordFormat="Hashed" maxInvalidPasswordAttempts="5"  
minRequiredPasswordLength="5"  
minRequiredNonalphanumericCharacters="0"  
passwordAttemptWindow="10"  
passwordStrengthRegularExpression="" name="MoreFriends"  
type="System.Web.Security.SqlMembershipProvider" />  
</providers>  
</membership>
```

Sample code for create user:

```
Protected Sub CreateUserWizard1_CreatingUser(ByVal sender As Object,  
ByVal e As System.Web.UI.WebControls.LoginCancelEventArgs) Handles  
CreateUserWizard1.CreatingUser
```

```
Dim myNewUser As MembershipUser  
Dim status As System.Web.Security.MembershipCreateStatus  
' Get the correct membershipprovider for the user.  
Dim myMembership As MembershipProvider =  
Membership.Providers(Session.Item("ApplicationName"))  
myNewUser = myMembership.CreateUser(CreateUserWizard1.UserName,  
CreateUserWizard1.Password, CreateUserWizard1.Email,  
CreateUserWizard1.Question, CreateUserWizard1.Answer, True, Nothing, status)  
  
' Add the created user to the role Friends.  
Roles.AddUserToRole(CreateUserWizard1.UserName, "Friends")  
e.Cancel = True
```

End Sub

"Steven Cheng[MSFT]" wrote:

Hi MrGrundh,

Nice to hear from you and sorry for the delay response.

As for the System.Web.Security.Roles class you mentioned, it actually call the Roles.Provider to perform any role management operations. And the "Provider" property refer to the default RoleProvider configured for the ASP.NET application. Just like the membership provider, ASP.NET 2.0 configure the SqlRoleProvider as the default role provider. We can also use the following web.config section to override or define our custom role providers:

=====

```
<roleManager defaultProvider="NewAspNetSqlRoleProvider">
<providers>
<add name="NewAspNetSqlRoleProvider"
connectionStringName="LocalSqlServer"
applicationName="MyApplication"
type="System.Web.Security.SqlRoleProvider, System.Web,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
</providers>
</roleManager>
```

=====

As you can see, the roleprovider also contains a "ApplicationName" which determine the application in which the role we'll manage.

so for your application, the question here is whether you'll manage the roles for users in different membership providers(different applications ) together or separate them(each membership application's user will have their own roles in each application)?

\*\*if you will let those users in different applications share the same roles, you can just define the single roleProvider or just the default one and do not need to explicitly set ApplicationName for the role provider.

\*\*if you want each Application has its own roles, you can just define multiple RoleProviders in the <providers> section above, and then, use the following code to locate the certain provider according to the ApplicationName of the certain user:

RE: Multiple Membership providers and AddUserToRole

```
=====  
RoleProvider provider = Roles.Providers["ApplicationName"];  
provider.AddUsersToRoles(....)  
=====
```

Hope this helps.

Actually, most of the new services in ASP.NET 2.0 adopt the provider-based model which is convenient for us to manage them in configuration file.

You can find more info about providers in ASP.NET 2.0 in the following site:

#Provider Toolkit  
<http://msdn.microsoft.com/asp.net/downloads/providers/default.aspx>

Sincerely,

Steven Cheng

Microsoft MSDN Online Support Lead

=====  
Get notification to my posts through email? Please refer to  
<http://msdn.microsoft.com/subscriptions/managednewsgroups/default.aspx#notifications>.

Note: The MSDN Managed Newsgroup support offering is for non-urgent issues where an initial response from the community or a Microsoft Support Engineer within 1 business day is acceptable. Please note that each follow up response may take approximately 2 business days as the support professional working with you may need further investigation to reach the most efficient resolution. The offering is not appropriate for situations that require urgent, real-time or phone-based interactions or complex project analysis and dump analysis issues. Issues of this nature are best handled working with a dedicated Microsoft Support Engineer by contacting Microsoft Customer Support Services (CSS) at <http://msdn.microsoft.com/subscriptions/support/default.aspx>.

=====  
This posting is provided "AS IS" with no warranties, and confers no rights.

RE: Multiple Membership providers and AddUserToRole