

# Remoting Questions

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2007-05/msg00026.html>

---

- *From:* "Wimpie van Lingen" <[van\\_diesel@xxxxxxxxxxxxx](mailto:van_diesel@xxxxxxxxxxxxx)>
  - *Date:* Fri, 18 May 2007 19:18:13 +0200
- 

Hey

I have some more questions with regards to Remoting in .NET 2. I'm using TCP with the Binary formatter.

My solution consists of 4 projects:

- Class Library containing the server classes which Inherits MarshalByRefObject (ok, at this stage it only contains one class... but its gonna grow)
- Class Library containing common classes and interfaces that will be shared between all projects. This include interfaces for the server objects so the clients won't have to reference the server class library directly and serializable objects like UserInfo, etc.
- Console Application that will host the server classes. Will later convert this to a Windows Service
- Windows forms application that will serve as the user interface

Ok, so this setup works. I've created a login form on the client, a UserIdentity class and ISecurityManagement interface in the common class library and a SecurityManagement class on the server.

So my code looks basically like this:

When the app starts, the following code executes:

```
Private Sub MyApplication_Startup(ByVal sender As Object, ByVal e As ApplicationServices.StartupEventArgs) Handles Me.Startup
```

```
RemotingConfiguration.ApplicationName = "MyApplication"
```

```
Channels.ChannelServices.RegisterChannel(New Channels.Tcp.TcpChannel, True)
```

```
Dim strUrl As String = My.Settings.Protocol & "://" & My.Settings.Server & ":" & My.Settings.Port.ToString.Trim
```

```
'This basically translates to "tcp://10.0.0.11:1981'
```

```
My.Application.ServerUrl = strUrl
```

```
RemotingConfiguration.RegisterWellKnownServiceType (GetType(ISecurityManagement), _
```

## Remoting Questions

```
strUrl & "/SecurityManagement.rem", _  
WellKnownObjectMode.SingleCall)  
LoginForm.ShowDialog()
```

End Sub

When the user clicks Login, the following code executes:

```
Private Sub OK_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles OK.Click
```

```
Dim args() As Object = Nothing  
Try  
Dim serverObj As ISecurityManagement = _  
Activator.GetObject(  
GetType(CommonLib.ServerInterfaces.ISecurityManagement), _  
My.Application.ServerUrl & "/SecurityManagement")
```

```
Dim Usr As CommonLib.Security.UserIdentity =  
ServerObj.Login(txtUsername.Text, txtPassword.Text)
```

```
MsgBox(Usr.ToString)  
Me.Close()  
Catch ex As Exception  
MsgBox(ex.Message)  
End Try
```

End Sub

Ok, this works if I'm working on one computer as long as the Console Application / Service hosting the server components are running. The problem I have is that as soon as I copy the client exe with the CommonLib.dll over to another computer, it doesn't work. I get an error message when attempting to log on (the code in brown).

If the service / console application is not running, I get a connection error, which is great because it tells me the app is trying to connect to the correct location. This error obviously occurs on both computers.

If I do start the service / console app, the client located on my computer works fine. The client on the other computer gives me the error:

"The server has rejected the client credentials."

If I set the EnsureSecurity option to False when registering the channel (code marked in orange), the application freeze on that line and I either have to stop the service to raise an error or terminate the client app.

Any idea how I can fix this?

Next question. The line of code marked in Navy. The application works if I leave that line in or if I remove it either way. So what is the use to have that code there if it doesn't affect whether the app works or not?

## Remoting Questions

Finally I have a question about which type of remoting objects would work best for my app.

Here are some details on my app:

The database and the components talking directly to the database will be located on the server. On the client side, the windows forms will be quite a large application that will communicate with the server components. They will for instance, open the Training Module and there they will constantly be busy filtering training records, scheduling new training, cancelling training, etc. Almost for every action they take there will be call(s) made to the database.

Remoting will be done over TCP with a binary formatter.

So what would be the best way to create the server components?

SingleCall, connection to the database is created and opened each time a call is made to the object.

or

Singleton where one connection will be created for the application and Opened / Closed for each call.

I don't know too much about remoting at this stage so I don't know all my options. If someone has some suggestions, please post 'em.

Thanks

William

.