

# Configuring & Using Roles in ASP.NET Applications

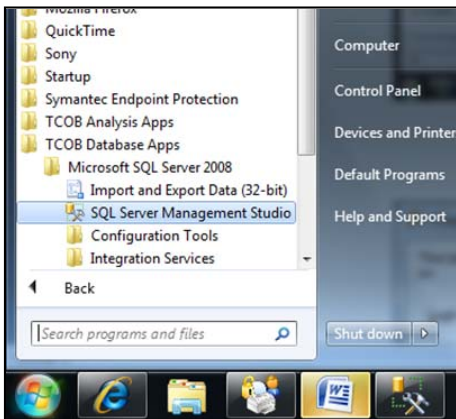
10/24/2014 MC

## Connect to your Apollo SQL database

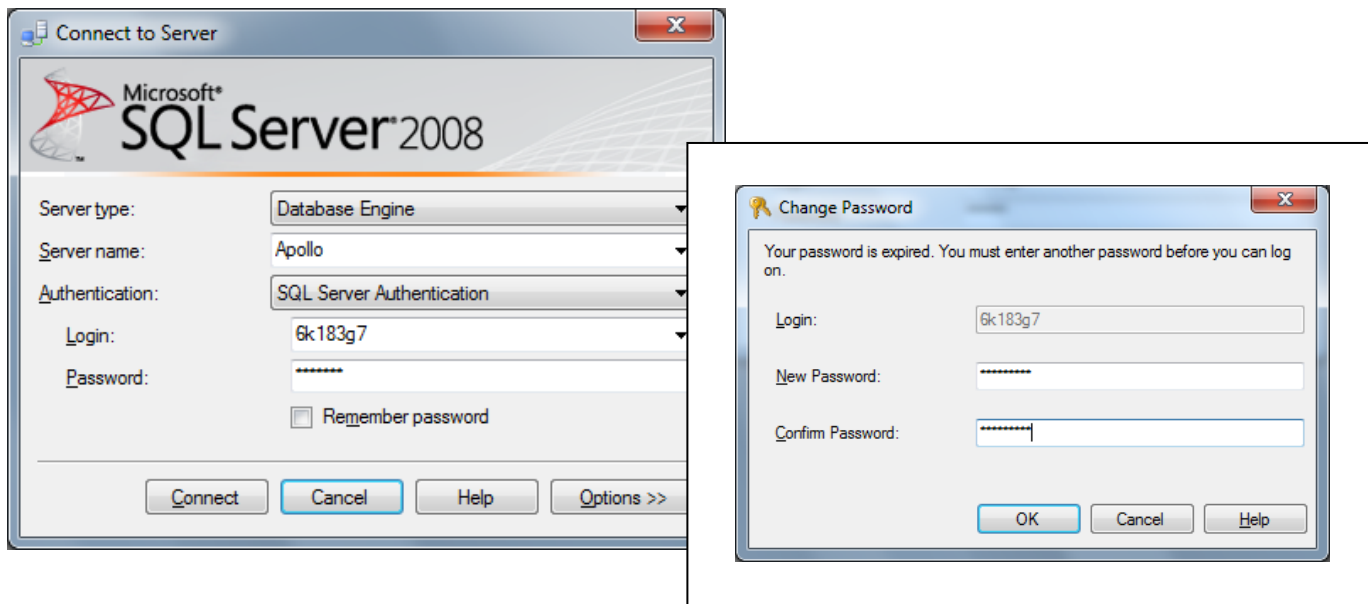
*NOTE: You can only manage Apollo from campus or through a VPN connection.*

From a computer lab or MIS lab PC, go to

**All Programs → TCOB Database Apps → Microsoft SQL Server 2008 → SQL Server Management Studio**



Connect to Apollo using **SQL Server Authentication**. **Login ID** and **password** is your group information.



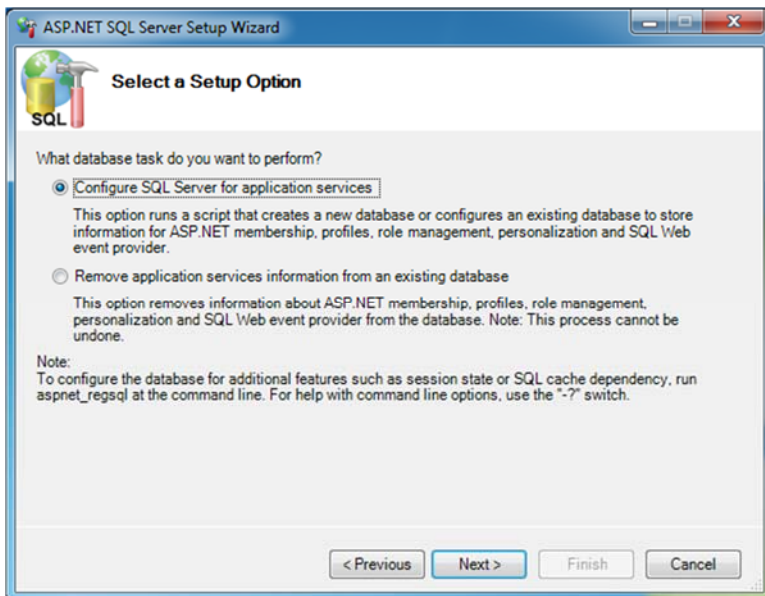
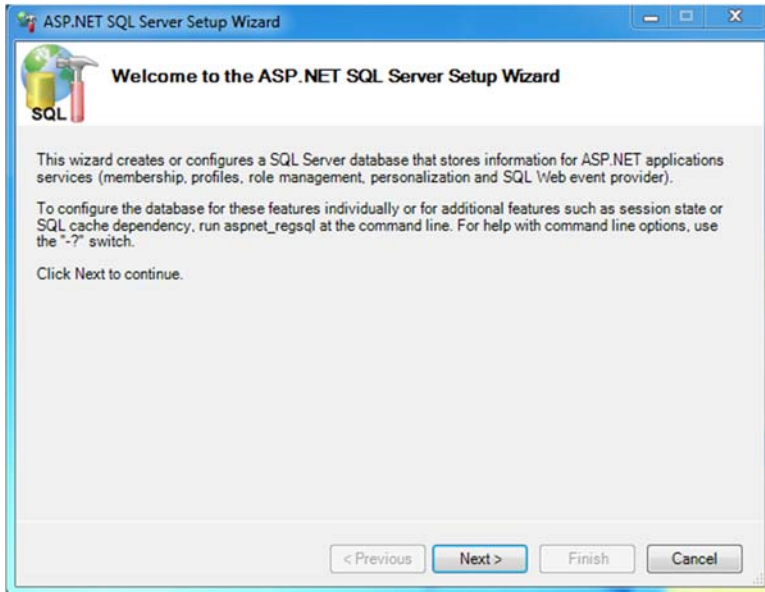
After you login you will have access only to your group's database. You can right-click Tables and Add New Table. In the new table you can create the fields, datatypes, keys, etc.

To add data directly to your table in SQL manager, right-click the table and choose "Edit top 200 rows".

# Prepare the SQL database for role based security

Open a DOS box

C:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet\_regsql.exe



ASP.NET SQL Server Setup Wizard

### Select the Server and Database

Specify the SQL Server name, database name to create or remove, and the credentials to use when connecting to the database.

**Note:** The credentials must identify a user account that has permissions to create or remove a database.

Server:

☐ Windows authentication

☒ SQL Server authentication

User name:

Password:

Database:

< Previous   Next >   Finish   Cancel

<default> will be displayed

You need to type in the actual name of your database

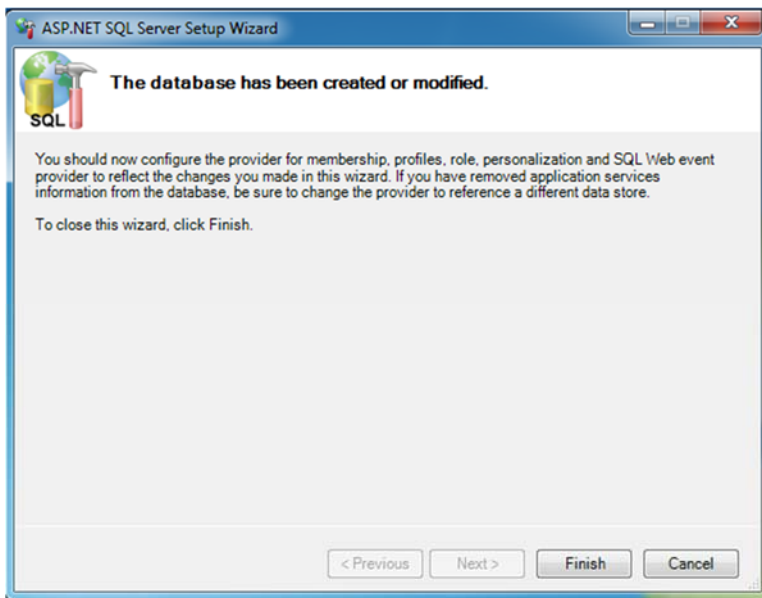
ASP.NET SQL Server Setup Wizard

### Confirm Your Settings

**Settings summary:**  
Server name: Apollo  
Database name: msci3300\_g0

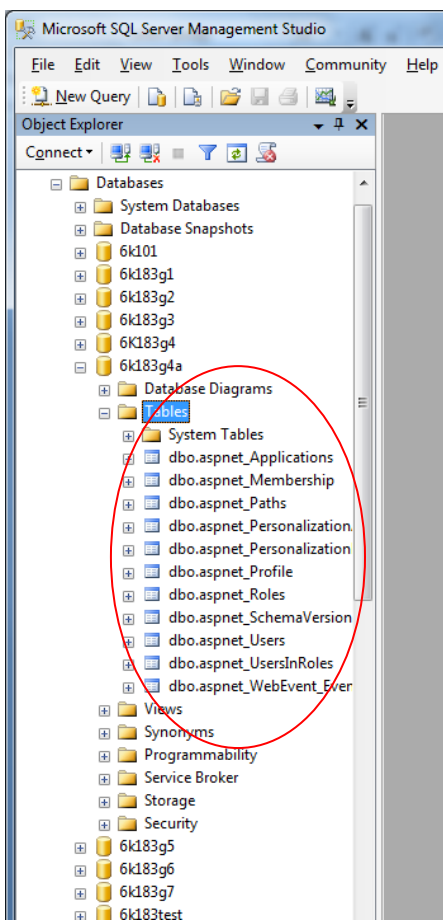
Click Next to continue or Previous to change your settings.

< Previous   Next >   Finish   Cancel



After the wizard completes with no errors, refresh your database in SQL Server Manager (right click the database and select “Refresh”).

You should see several tables created that begin with “aspnet\_”



# Prepare the ASP.NET application to use role based security

Edit web.config

Within <system.web> add

```
<roleManager enabled="true"/>

<authentication mode="Forms">
  <forms loginUrl="login/login.aspx"/>
</authentication>
```

This is the path to your login page.

Outside of <system.web> add

```
<connectionStrings>
  <remove name="LocalSqlServer"/>
  <add name="LocalSqlServer" connectionString="Data Source=apollo.biz.uiowa.edu; Initial Catalog=
msci3300_g0; Persist Security Info=True; User ID=msci3300_g0; Password=pAssw0rd!"/>
  <add name="cs_ToolRental" connectionString="Data Source=apollo.biz.uiowa.edu; Initial Catalog=
msci3300_g0; Persist Security Info=True; User ID=msci3300_g0; Password=pAssw0rd!"
providerName="System.Data.SqlClient"/>
</connectionStrings>
```

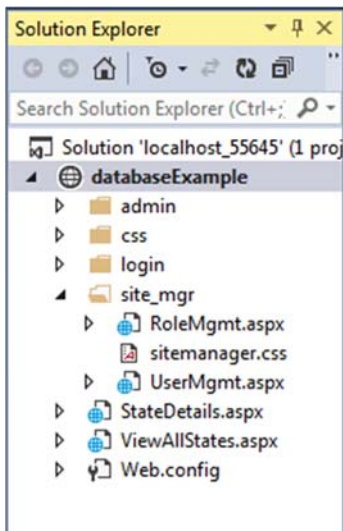
Remove any notion of LocalSqlServer

Make the application think it has a local SQL server, but it is actually mapped to Apollo

Make a connection string for your SqlDataSource controls to use to connect to the Apollo database

## Configure roles & users for your application

After the aspnet\_ tables are created using aspnet\_regsql.exe utility and your root web.config file has been configured to point your application to the correct database and configured to use role based security, then unzip **site\_mgr.zip** and add the folder to your web application.



Open **UserMgmt.aspx** in a browser and add a new user.

**Sign Up for Your New Account**

User Name:

Password:

Confirm Password:

E-mail:

Security Question:

Security Answer:

You can also delete users from this page.

### Delete An Existing User

Open **RoleMgmt.aspx** in a browser and add a new role.

Add A New Role:

r\_colbert\_Admin

Add This New Role

*Remember since we are sharing databases to include your hawkid in both your role name and user names to make them unique to you.*

You can also delete roles from this page.

Current Roles:

r\_colbert\_Admin

Delete An Existing Role

Next add the new user to the new role using **RoleMgmt.aspx**

Click on the user, click the role and then click “Add This User To This Role”

**Add A User To A Role:**

Users:

mikec  
mikecolbert

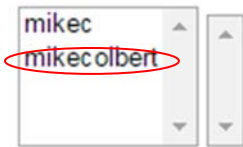
Roles:

r\_colbert\_Admin

Add This User To This Role

You can see what roles a particular user has and remove roles from the individual users by selecting the user, then selecting their role to be removed and choosing to remove.

### Remove A Role From A User

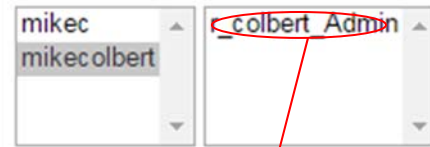


mikec
mikecolbert

Remove This Role From This User



### Remove A Role From A User



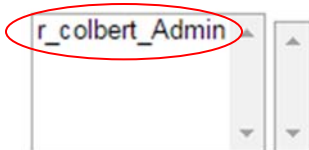
mikec
mikecolbert

colbert Admin
---------------

Remove This Role From This User

You can also see what users are a member of a particular role and remove users from the role by selecting the role, then selecting the user to be removed and choosing to remove.

### Remove A User From A Role

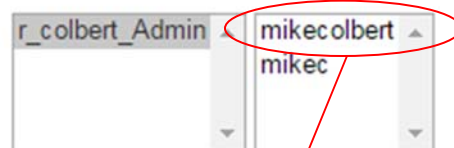


r_colbert_Admin
-----------------

Remove This User From This Role



### Remove A User From A Role



r_colbert_Admin
-----------------

mikecolbert
mikec

Remove This User From This Role



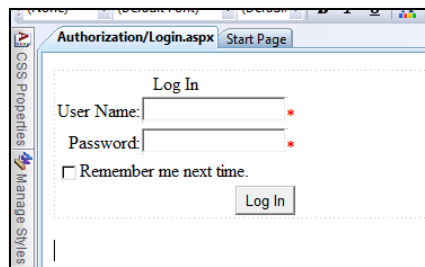
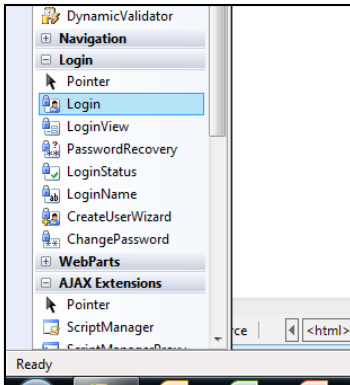
## Test Your Roles

Create a new folder at the root of you website named **login**.

Create a new web form in the Authorization folder named **login.aspx**

*NOTE: This folder and filename are specified in the <authentication> piece you added to web.config. Make sure everything matches.*

From the Toolbox → Login → Add a **login control** to Login.aspx and save the page



## Restrict access to folders based on roles:

Right-click the **Admin** folder, choose **Add New Item**, choose **Web Configuration File** and **Add**

*Note: Leave the file name as the default.. It should be Web.config*

Double-click Web.config under the **Admin** folder to open it.

Within the <system.web> element add the following

```
<authorization>

  <deny users="?" />
  <deny roles="Patron, Librarian" />
  <allow roles="SuperLibrarian" />

</authorization>
```

You can customize security for this folder using the sample authorizations above.

```
<deny users="?" />
```

You can deny “unauthenticated users” using the “?”. Unauthenticated users are users who do not have a username and password for your site or have not entered their credentials yet.

```
<allow roles="SuperLibrarian" />
```

You can allow specific roles into a folder. This role name should match the role name you created above.

```
<deny roles="Patron, Librarian" />
```

You can deny one or more specific roles into a folder. This role name should match the role name you created above. Multiple roles are separated as shown. You can also allow multiple roles using the same notation.