2011

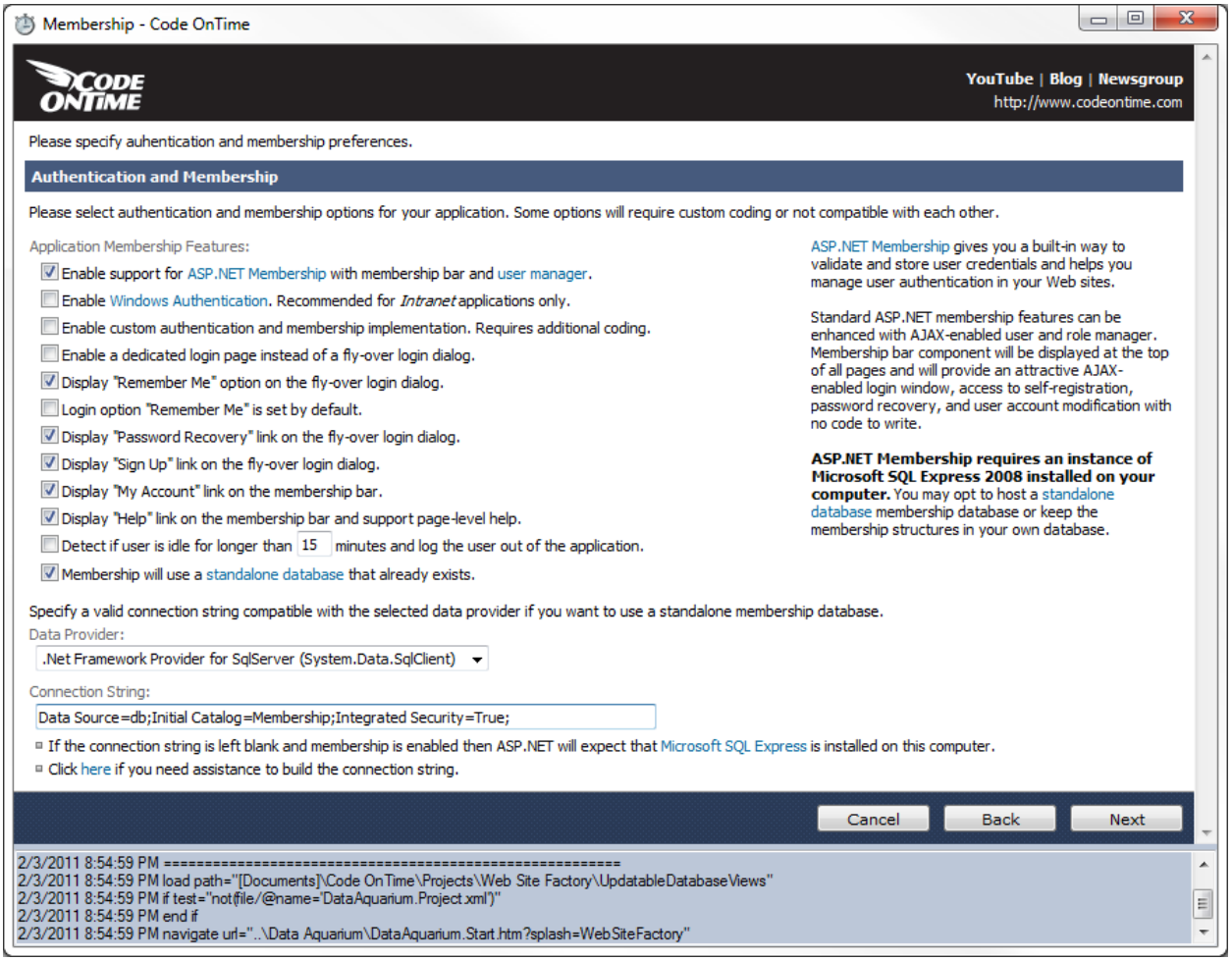
COOKBOOK

Standalone ASP.NET Membership Database

Web Site Factory and other projects integrate *ASP.NET Membership*, a built-in way to store and validate user credentials. You can enable *ASP.NET Membership* by selecting the membership option in the code generator project wizard. This will enable numerous membership features including a fly-over sign-in window, self-service membership enrollment, membership bar, and membership manager.

The configuration of your project will be automatically changed to support the default membership provider available in ASP.NET. This provider defines a connection string that points to a local instance of *Microsoft SQL Server Express*. The provider will automatically connect to the server and dynamically create a database to maintain users, roles, and other membership features. The database will be created under *~/App_Data* folder of your project.

This works great on a development machine with installed *SQL Server Express*. There are many situations when you want to use a standalone membership database or store ASP.NET membership data structures directly in your own database.



Project wizard offers an option that will enable a standalone membership database configuration. Here is the screen shot of the project wizard with the standalone membership database enabled.

The connection string in the screen shot looks as follows:

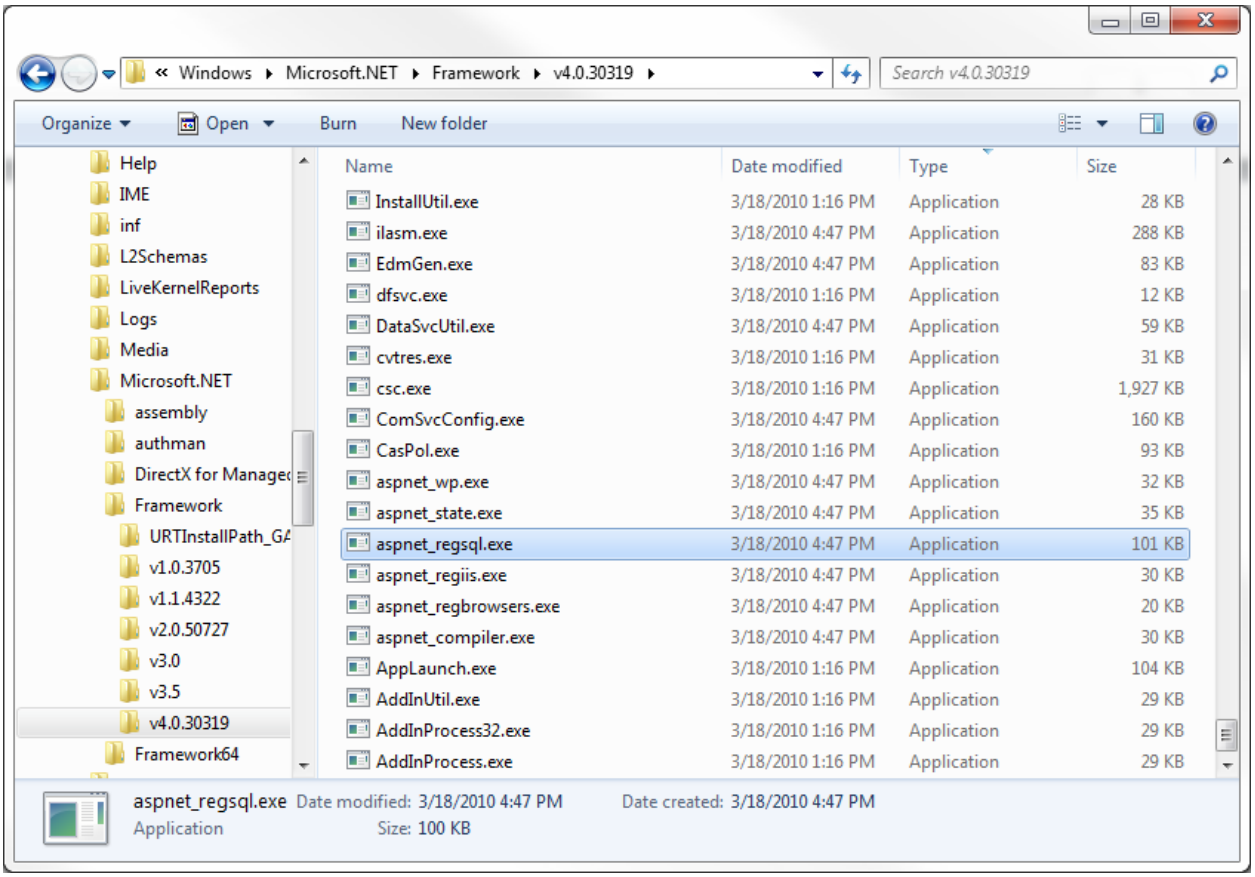
Data Source=db;Initial Catalog=Membership;Integrated Security=True;

We have configured the standalone membership database with the name *aspnetdb*.

You can read more about the configuration process at http://msdn.microsoft.com/en-us/library/ms229862(VS.80).aspx.

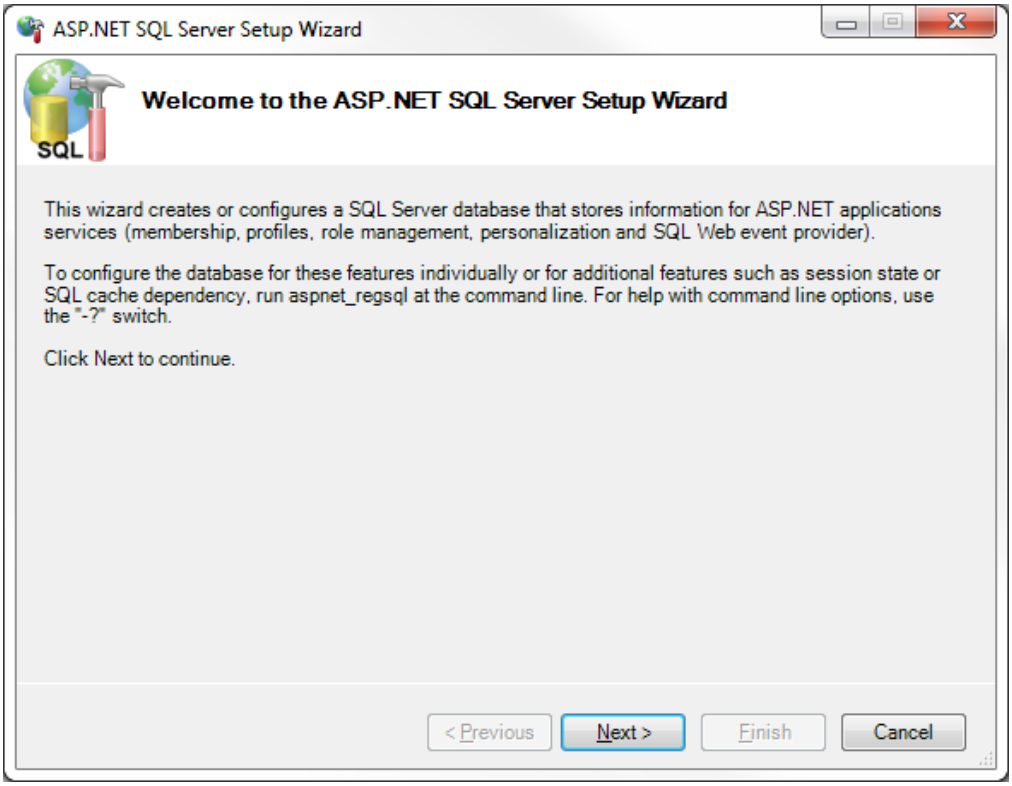
These are the steps that we have taken to create the *aspnetdb* database:

1. We have started *aspnet_regsql.exe* from *Windows Explorer* as shown in picture.

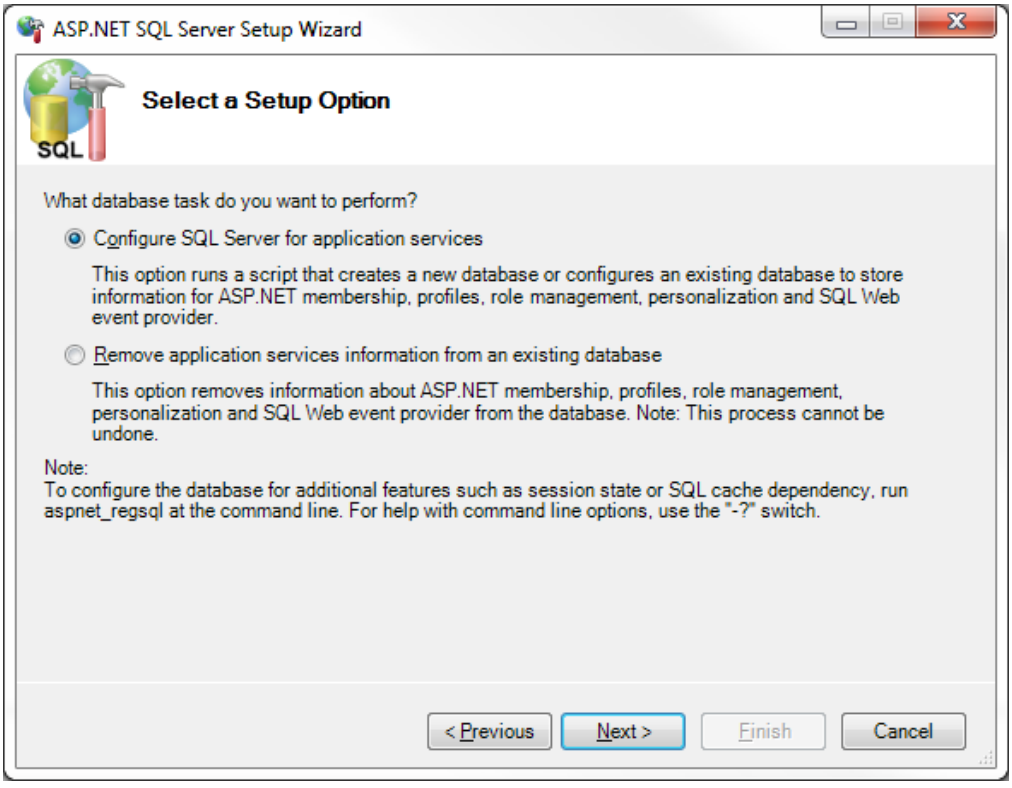


The path to your instance of *aspnet_regsql.exe*:
C:\%windir%\Microsoft.NET\Framework\<versionNumber>\aspnet_regsql.exe

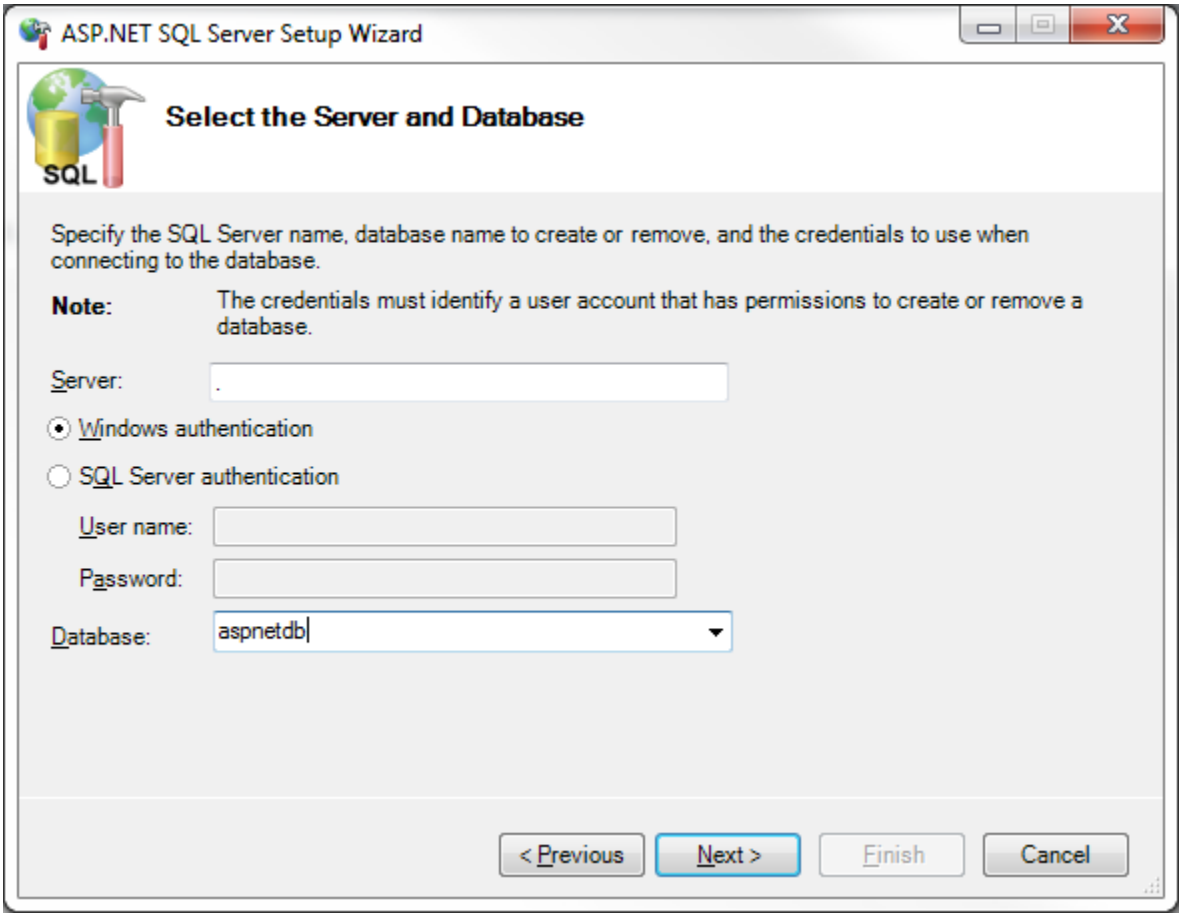
2. We have clicked *Next* button in *ASP.NET SQL Server Setup Wizard*:



3. We have continued to the next step to configure *SQL Server* for application services:



4. We have entered "." as a server name and "aspnetdb" as database name.



A few more clicks on the *Next* button have done the job for us. The database has been created. We have returned to the project wizard of our code generation project and configure the provider name and connection string of the newly created membership database.