Gora LEYE ~ Expert C#.NET

# Creating the Application Services Database for SQL Server

*21 Samedi sept 2013*

PUBLIÉ PAR GORALEYE IN DATABASE, MEMBERSHIP, SECURITY, SQL SERVER

≈ UN COMMENTAIRE

*Tags*
*Application Services, aspnet_regsql, database, MemberShip, Security, sql server*

ASP.NET includes a tool for installing the SQL Server database used by the SQL Server providers, named Aspnet_regsql.exe. The Aspnet_regsql.exe tool is located in the drive:\WINDOWS\Microsoft.NET\Framework\versionNumber folder on your Web server. Aspnet_regsql.exe is used to both create the SQL Server database and add or remove options from an existing database.

You can run Aspnet_regsql.exe without any command line arguments to run a wizard that will walk you through specifying connection information for the computer running SQL Server and installing or removing the database elements for all the supported features. You can also run Aspnet_regsql.exe as a command-line tool to specify database elements for individual features to add or remove.

| Note |
|---|
| The database elements that are installed in the feature database will always be owned by the SQL Server database owner account (dbo). In order to install the feature database, a SQL Server login must be permitted to the db_ddladmin and dd_securityadmin roles for the SQL Server database. However, you do not need to be a system administrator for the SQL Server in order to install the feature database. |

To run the Aspnet_regsql.exe wizard, run Aspnet_regsql.exe without any command line arguments, as shown in the following example:

```
C:\WINDOWS\Microsoft.NET\Framework\<versionNumber>\aspnet_regsql.exe
```

You can also run the Aspnet_regsql.exe tool as a command-line utility. For example, the following command installs the database elements for membership and role management on the local computer running SQL Server:

```
aspnet_regsql.exe -E -S localhost -A mr
```

The following table describes the command line options supported by the Aspnet_regsql.exe tool.

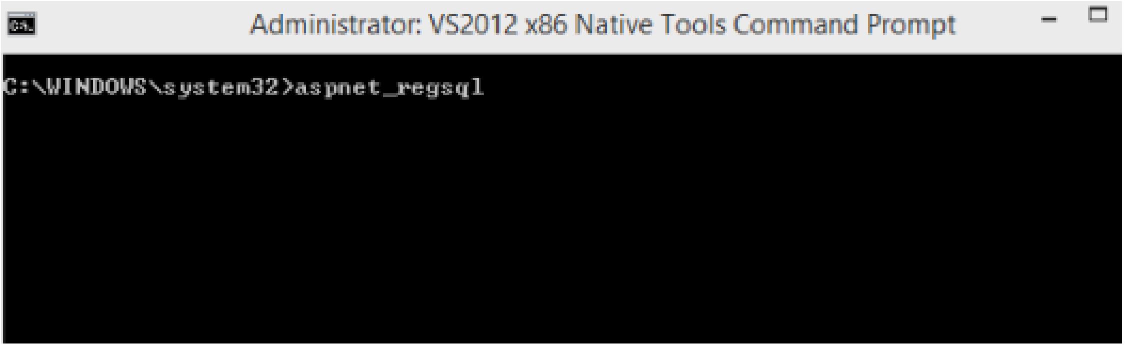
| Option | Description |
|---|---|
| -? | Prints Aspnet_regsql.exe tool Help text in the command window. |
| -W | Runs the tool in wizard mode. This is the default if no command line arguments are specified. |
| -C connection string | The connection string to the computer running SQL Server where the database will be installed, or is already installed. This option is not necessary if you only specify the server (-S) and login (-U and -P, or -E) information. |
| -S server | The name of the computer running SQL Server where the database will be installed, or is already installed. The server name can also include an instance name, such as .\INSTANCENAME. |
| -U login id | The SQL Server user id to log in with. This option also requires the password (-P) option. This option is not necessary if you are authenticating using Windows credentials (-E). |
| -P password | The SQL Server password to log in with. This option also requires the login id (-U) option. This option is not necessary if authenticating using Windows credentials (-E). |
| -E | Authenticates using the Windows credentials of the currently logged-in user. |
| -d database | The name of the database to create or modify. If the database is not specified, the default database name of "aspnetdb" is used. |
| -sqlexportonlyfilename | Generates a SQL script file that can be used to add or remove the specified features. The specified actions are not performed. |
| -A all\|m\|r\|p\|c\|w | Adds support for one or more features. The following identifiers are used for ASP.NET features.<br>IdentifierAffects<br>allAll features<br>mMembership<br>rRole management<br>pProfile<br>cWeb Parts personalization<br>wWeb events<br>Feature identifiers can be specified together or separately, as shown in the following examples.<br>aspnet_regsql.exe -E -S localhost -A mp<br>aspnet_regsql.exe -E -S localhost -A m -A p |
| -R all\|m\|r\|p\|c\|w | Removes support for one or more features. The following identifiers are used for ASP.NET features.<br>IdentifierAffects |

| | |
|---|---|
| | allAll features<br>mMembership<br>rRole management<br>pProfile<br>cWeb Parts personalization<br>wWeb events<br>Feature identifiers can be specified together or separately, as shown in the following examples.<br>aspnet_regsql.exe -E -S localhost -R mp<br>aspnet_regsql.exe -E -S localhost -R m -R p |
| -Q | Runs the tool in quiet mode and does not confirm before removing a feature. |

In this section , we can create our security database according to our business model and store it on sql server, oracle, mysql or other.

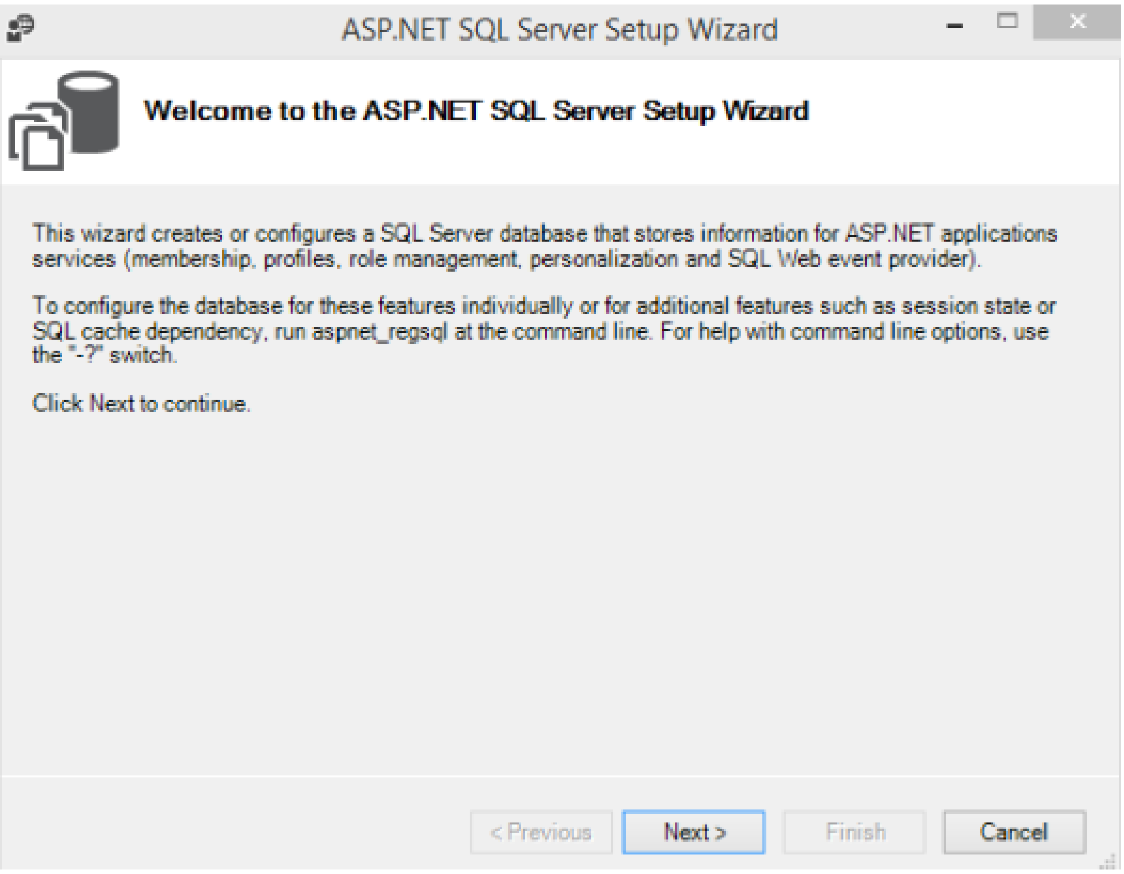
*If you have already a security database, go to the next section.*

Open visual studio prompt command tool and run command line **aspnet_regsql** as following



```
C:\WINDOWS\system32>aspnet_regsql
```

The screen explain the wizard scenario  so Click next



ASP.NET SQL Server Setup Wizard

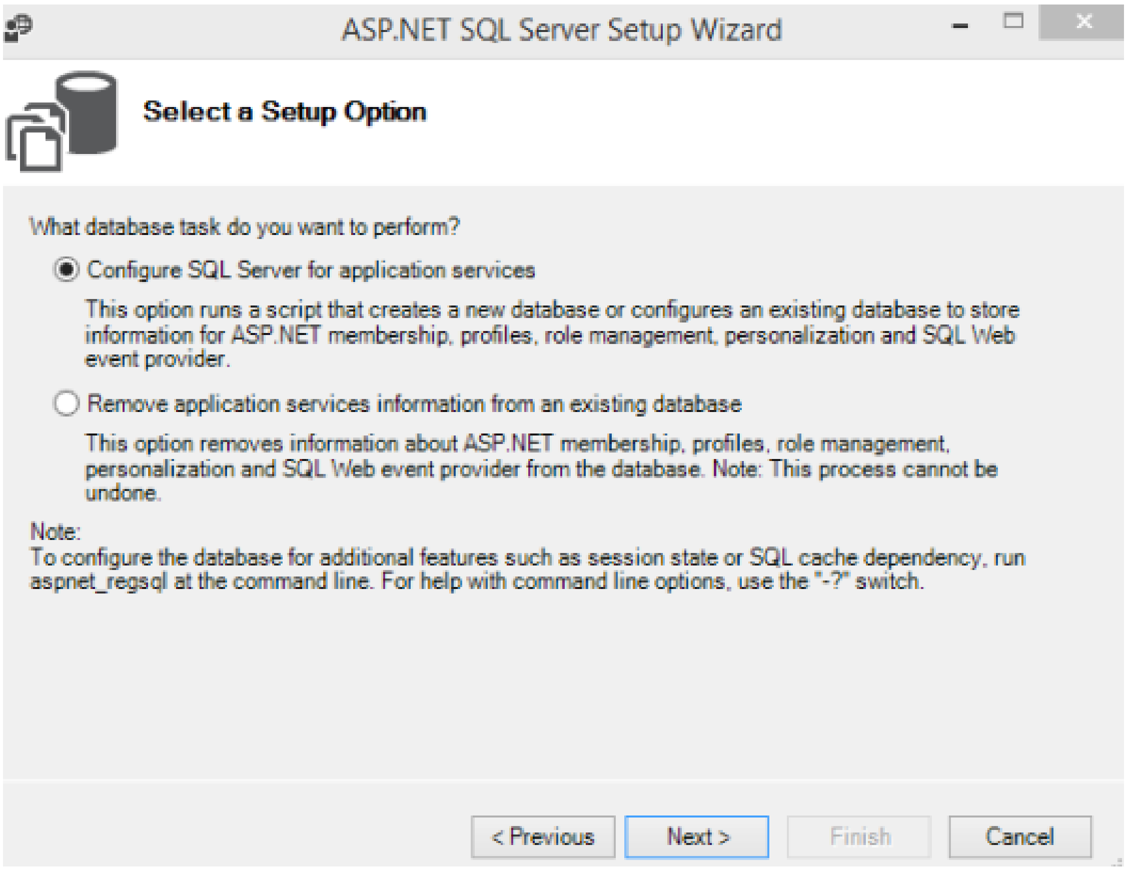
**Welcome to the ASP.NET SQL Server Setup Wizard**

This wizard creates or configures a SQL Server database that stores information for ASP.NET applications services (membership, profiles, role management, personalization and SQL Web event provider).

To configure the database for these features individually or for additional features such as session state or SQL cache dependency, run aspnet_regsql at the command line. For help with command line options, use the "-?" switch.

Click Next to continue.

Here we can remove existing security database or create a new one. We want to create a new security database. So check the first option and click next



ASP.NET SQL Server Setup Wizard

**Select a Setup Option**

What database task do you want to perform?

● Configure SQL Server for application services

This option runs a script that creates a new database or configures an existing database to store information for ASP.NET membership, profiles, role management, personalization and SQL Web event provider.

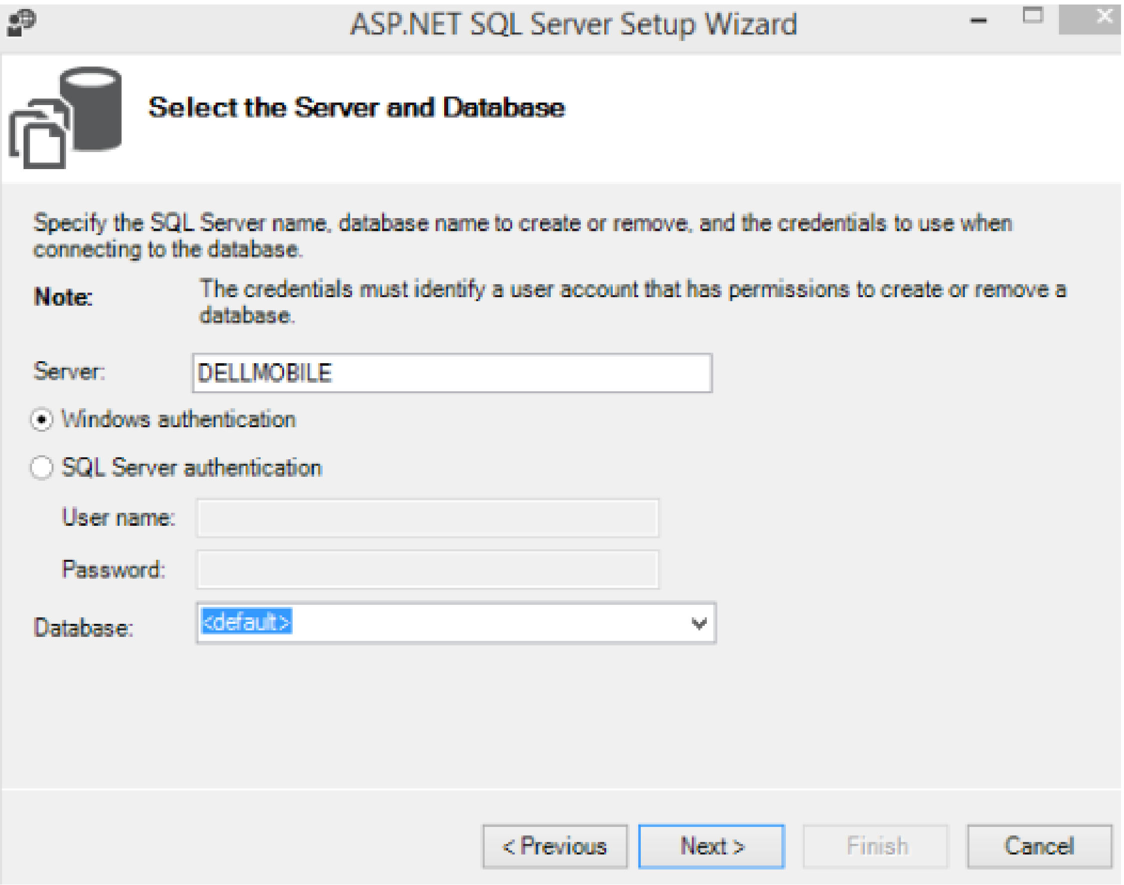
○ Remove application services information from an existing database

This option removes information about ASP.NET membership, profiles, role management, personalization and SQL Web event provider from the database. Note: This process cannot be undone.

Note:
To configure the database for additional features such as session state or SQL cache dependency, run aspnet_regsql at the command line. For help with command line options, use the "-?" switch.

Enter our database server name .\SQLEXPRESS ( enter the appropriate server name). if you already an existing database, you can select it. So the wizard will create the security tables on the selected datase.

If you do not have a database, let default. the default database name that will be created is aspnetdb



**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: DELLMOBILE

● Windows authentication

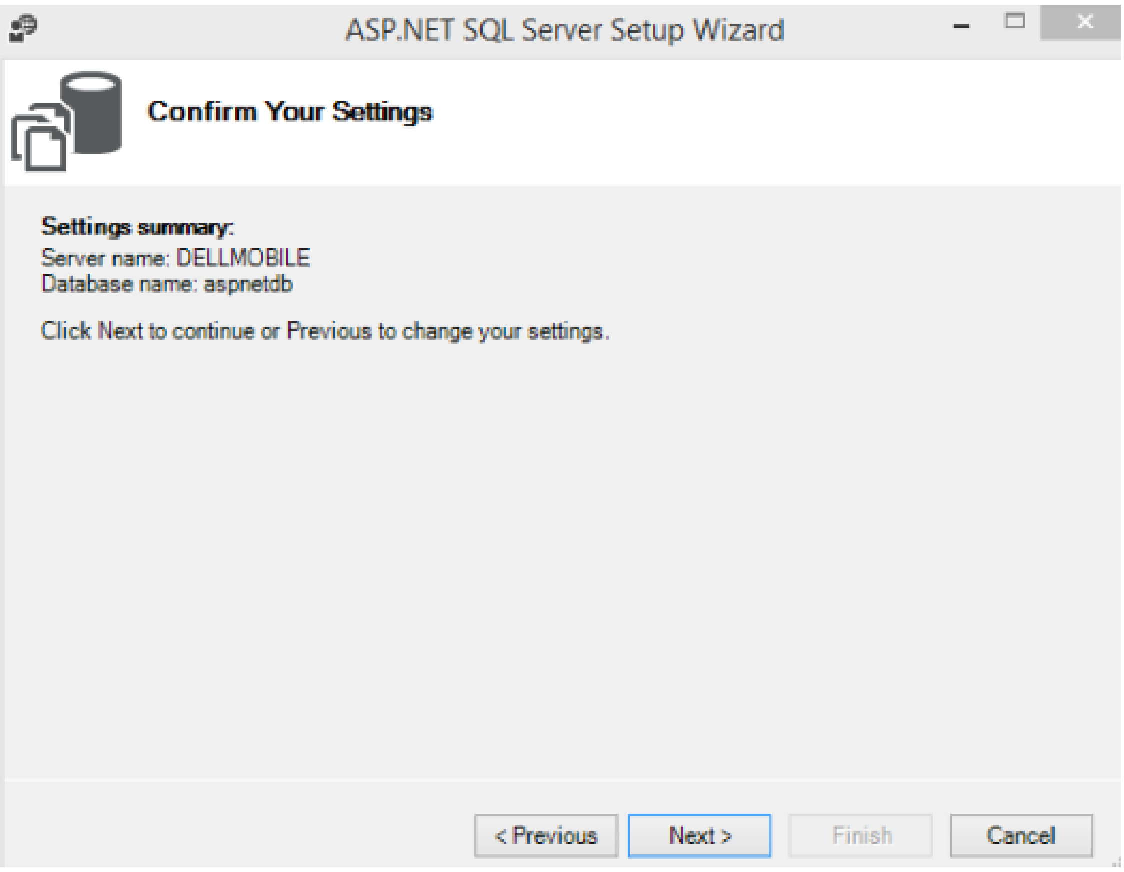
○ SQL Server authentication

User name:

Password:

Database: <default>

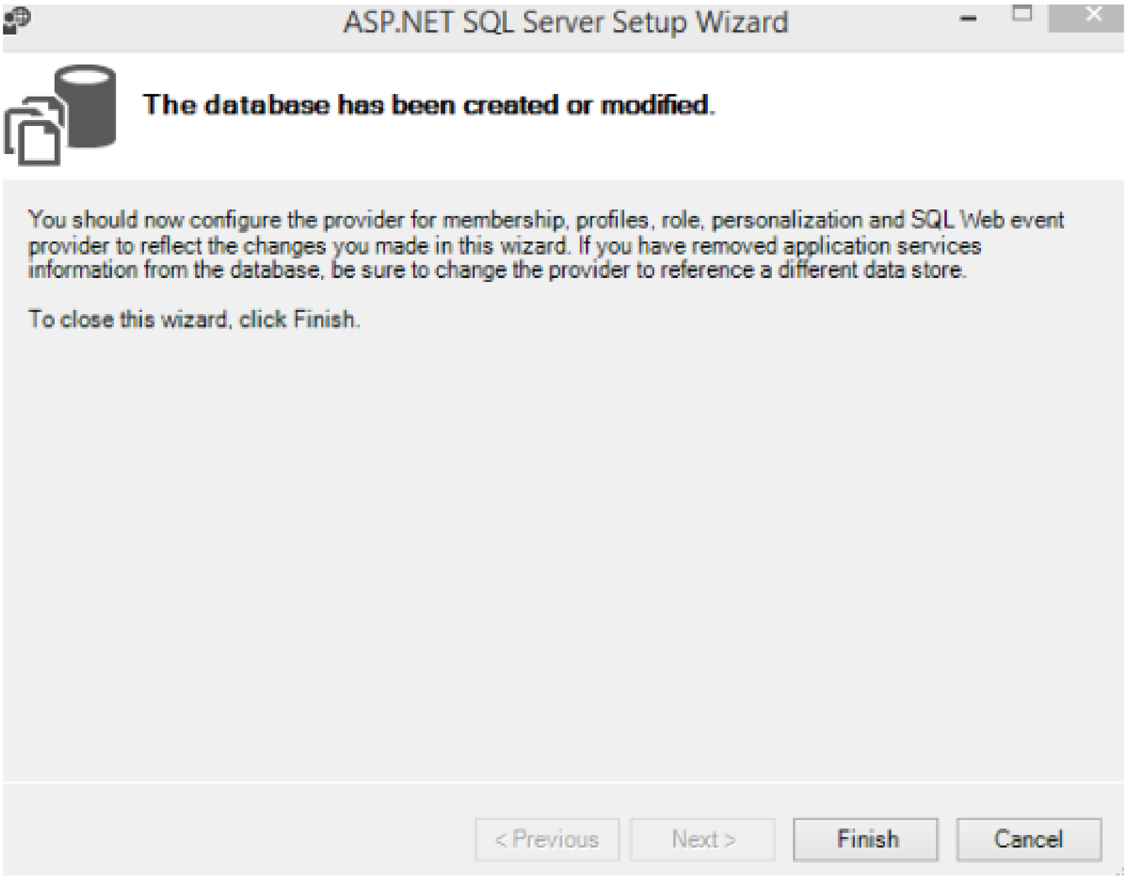
The wizard dispays the summary action, so click next to confirm  and finish the action



Confirm Your Settings

Settings summary:
Server name: DELLMOBILE
Database name: aspnetdb

Click Next to continue or Previous to change your settings.

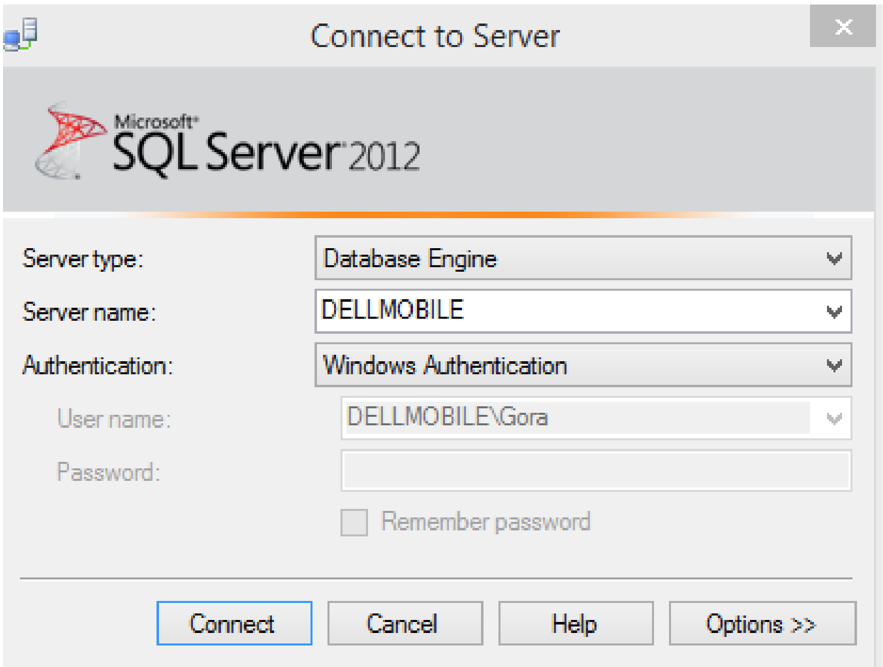


The database has been created or modified.

You should now configure the provider for membership, profiles, role, personalization and SQL Web event provider to reflect the changes you made in this wizard. If you have removed application services information from the database, be sure to change the provider to reference a different data store.

To close this wizard, click Finish.

Connect to sql server , locate aspnetdb database, expand tables, views, stored procedure.  you can now explore the default microsof security database business model.



Server type: Database Engine

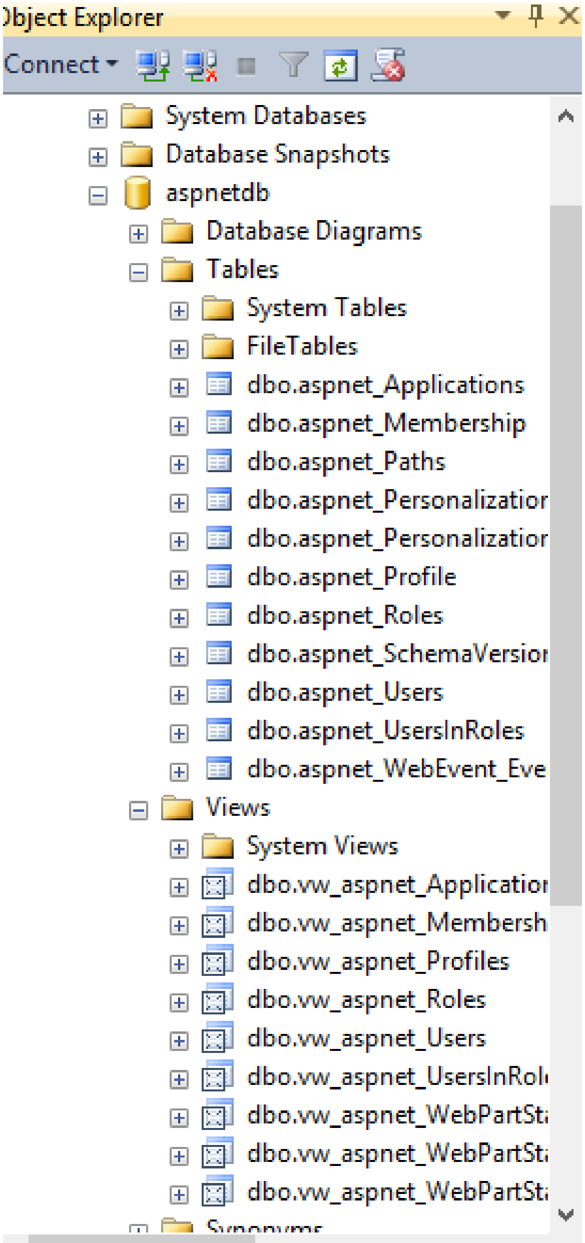
Server name: DELLMOBILE

Authentication: Windows Authentication

User name: DELLMOBILE\Gora

Password:

□ Remember password



for more informations please visit msdn web site

This tutorial  is the first of a series Mastering Custum ASP.NET MemberShip Provider using ASP.NET MVC , please see next  Introduction to entity framework database first

## réflexion à propos de “ Creating the Application Services Database for SQL Server ”

1. Ping: ASP.NET MVC Custom Membership Password Hashing based on SALT key using SHA-3 Algorithm | Gora LEYE

*1*