

Database Settings Tab

1 CONTENTS

A. Database Connection.....	2
1. Server.....	2
2. Database Name.....	2
3. User Name	2
4. Password.....	2
5. Show Password	2
B. Generated SQL Script.....	3
1. Use Linq-to-entities (Entity Framework Core).....	3
2. Use Stored Procedures	3
a. No Prefix or Suffix	3
b. Prefix.....	3
c. Suffix	3
3. Use Ad-Hoc/Dynamic SQL	3
a. Automatic	3
b. MS SQL 2008 Below	3
1 Database Connection.....	3
1.1 Using Database Connection to Connect to MS SQL Server Database to Generate Code	3
1.2 Using Database Connection to Connect to MS SQL Server Database to Perform CRUD	5
1.2.1 Using The Connection String	6
1.2.2 Using The Database Context Class (Entity Framework).....	6
1.2.3 Using The Connection String (Ad-Hoc or Stored Procedures)	6
2 Generated SQL.....	7
2.1 Use Linq-to-Entities (Entity Framework Core).....	7
2.2 Use Stored Procedures.....	8
2.2.1 Use Stored Procedures – No Prefix or Suffix	9
2.2.2 Use Stored Procedures – Prefix.....	10
2.2.3 Use Stored Procedures – Suffix	11
2.3 Use Ad Hoc/Dynamic SQL	12
2.3.1 Automatic	13
2.3.2 MS SQL 2008 Below	13

Database Settings Tab

The Database Settings Tab is where we set database-specific information such as the database connection properties and the type of SQL script to generate.

AspCoreGen 6.0 MVC Professional+

Selected Tables | Selected Views | **Database Settings** | Code Settings | UI Settings | App Settings

Database Connection

Server: localhost

Database Name:

User Name:

Password: ☐ Show Password

Generated SQL

☒ Use Linq-to-Entities (Entity Framework Core)

☐ Use Stored Procedures

☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☒ No Prefix or Suffix

☐ Prefix:

☐ Suffix:

Generated SQL Script

☒ Automatic

☐ MS SQL 2008_Below

About Close **Generate Code for All Tables** Cancel

A. DATABASE CONNECTION

This is the MS SQL Server Database connection settings to the database. All of these items are required and cannot be blank.

1. **SERVER:** Server Name. If the MS SQL Server Database is local, you can just enter “localhost”.
2. **DATABASE NAME:** The database you’re trying to access.
3. **USER NAME:** An admin user name login that has access to the database.
4. **PASSWORD:** The password associated with the user name.
5. **SHOW PASSWORD:** Checking the *Show Password* in Figure 9 will remember the password you enter here so you don’t have to enter it again the next time you open AspCoreGen 6.0 MVC.

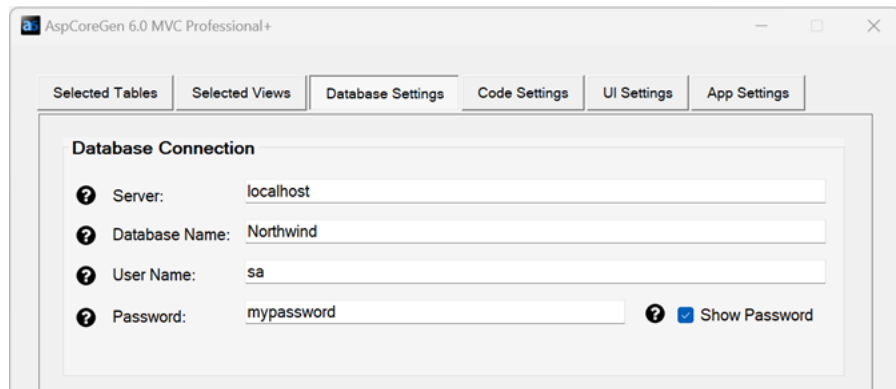
B. GENERATED SQL SCRIPT

The type of SQL script that will be generated.

1. **USE LINQ-TO-ENTITIES (ENTITY FRAMEWORK CORE):** Entity Framework.
2. **USE STORED PROCEDURES:** SQL Scripts generated straight in your MS SQL database.
 - a. **No Prefix or Suffix:** Names the generated script with no prefix or suffix. E.g. *Products_SelectAll*.
 - b. **Prefix:** Names the generated script with a prefix. For example, when you enter “*sp_*” on the Prefix text box, the generated stored procs will start with an “*sp_*”. E.g. *sp_Products_SelectAll*.
 - c. **Suffix:** Names the generated script with a suffix. For example, when you enter “*_sp*” on the Suffix text box, the generated stored procs will end with an “*_sp*”. E.g. *Products_SelectAll_sp*.
3. **USE AD-HOC/DYNAMIC SQL:** SQL Scripts generated in your C# code.
 - a. **Automatic:** Automatically determines your MS SQL Server version before generating scripts.
 - b. **MS SQL 2008 Below:** Generates scripts for MS SQL Server versions 2008 and below. There’s a slight difference with the generated script for older versions because some keywords are not available during this time.

1 DATABASE CONNECTION

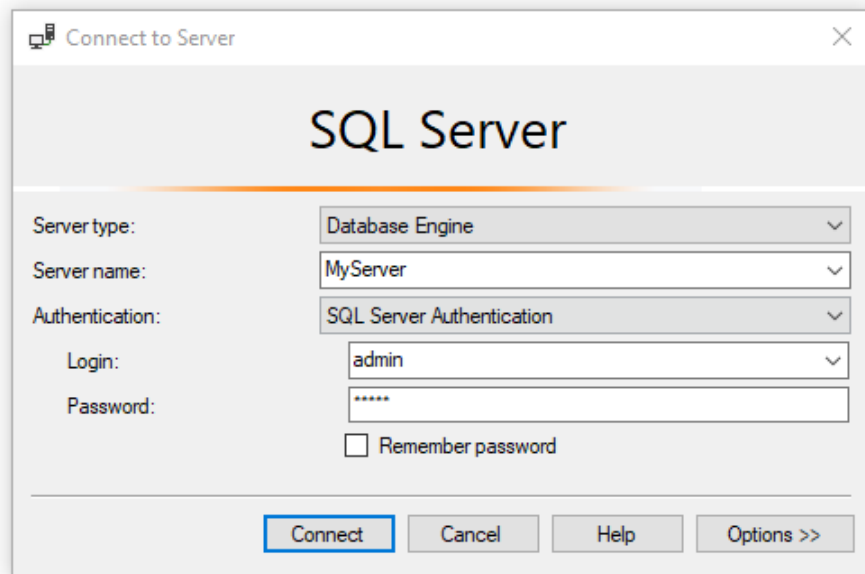
We use these fields/properties to connect to the MS SQL Server database, both to generate code and to perform CRUD (create, retrieve, update, delete) operations when you run the generated code.



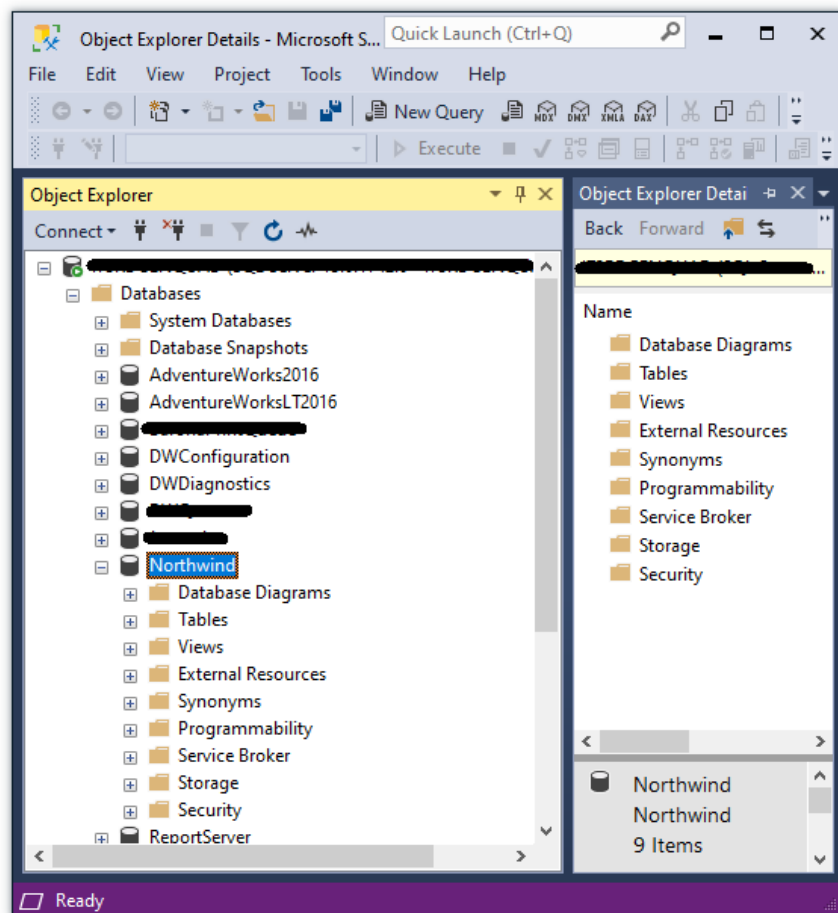
1.1 USING DATABASE CONNECTION TO CONNECT TO MS SQL SERVER DATABASE TO GENERATE CODE

When you open your MS SQL Server Database it displays the *Connect to Server* dialog. Please see the direct relationship between the Database Connection fields you enter in AspCoreGen 6.0 MVC and your MS SQL Server Database below.

AspCoreGen 6.0 MVC	MS SQL Server Database
Server: <i>localhost</i> (when local) or <i>MyServer</i>	Server Name: <i>localhost</i> (when local) or <i>MyServer</i>
Database Name: <i>Northwind</i>	Databases: <i>Northwind</i>
User Name: <i>admin</i>	Login: <i>admin</i>
Password: <i>admin</i>	Password: <i>admin</i>
Show Password: <i>true</i>	n/a



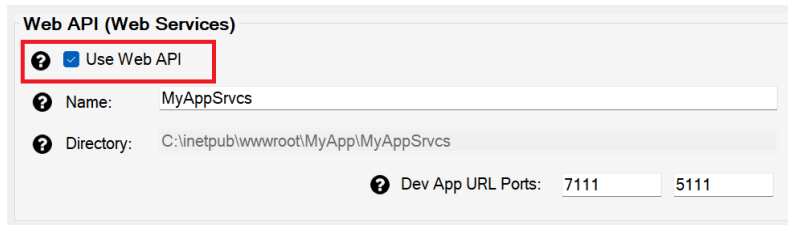
Once you're connected to MS SQL Server Database, you will see a list of *Databases* in the *Object Explorer*. The *Northwind* database is what we have in the example above (Database Name), the database we want to generate code from.



1.2 USING DATABASE CONNECTION TO CONNECT TO MS SQL SERVER DATABASE TO PERFORM CRUD

Once the code is generated, a part of the code is used to connect to the MS SQL Server Database you've chosen (*Northwind*) to perform CRUD operations on. In the generated code you will find the *Connection String* in the *appsettings.json* file.

When you choose *Use Web API* under the *Code Settings* tab, the *Connection String* will be in the *appsettings.json* file in the Web API project, otherwise it will be in the *appsettings.json* file in the Web Project.



Web API (Web Services)

☒ Use Web API

Name: MyAppSrvcs

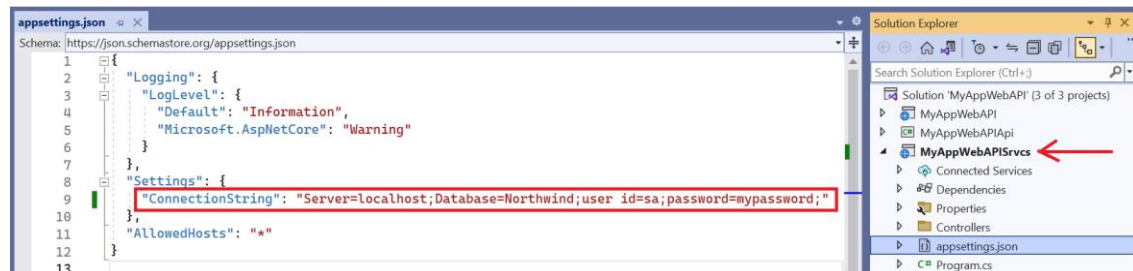
Directory: C:\inetpub\wwwroot\MyApp\MyAppSrvcs

Dev App URL Ports: 7111 5111

Use Web API	appsettings.json Location
Checked	Web API project
Not Checked	Web Application project



appsettings.json in Web Project



appsettings.json in Web API Project

Note: The *appsettings.json* file will not be overwritten by AspCoreGen 6.0 MVC the next time you re-generate code for the same project. So it is safe to add your own settings in this file. You can change the *Connection String* values once you push your web application to your production environment. Make sure to point it to your Production Database instead.

1.2.1 Using The Connection String

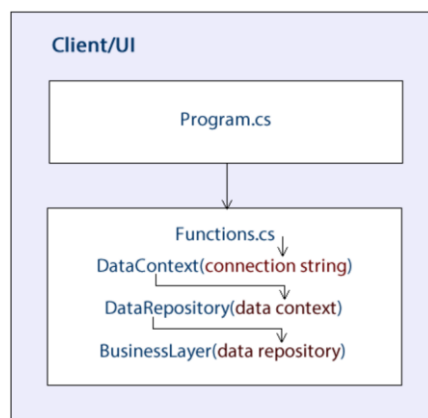
Note: In an n-tier or layered design, the *Data Repository Class* (data layer/tier) is ultimately **the only tier that calls to and from the database**.

We inject (dependency injection) the *Connection String* from the *Program.cs* (client) by calling a helper method in the *Functions.cs*.

AspCoreGen 6.0 MVC can generate SQL Script in 3 ways: Use Linq-To-Entities (Entity Framework), Ad-Hoc SQL, or Stored Procedures. Please see #2 *Generated SQL* below. Based on what you choose under the *Generated SQL*, the *Connection String* can either be injected to the *Data Repository* or to the *Data Context*.

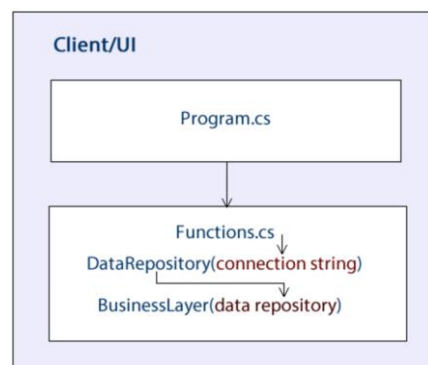
1.2.2 Using The Database Context Class (Entity Framework)

When you generate code with *Linq-To-Entities (Entity Framework)*, the *Connection String* is injected to the *Database Context Class* from the *Program.cs* (client). The *Database Context Class* is then injected to the *Data Repository* from the *Business Layer*. The *Data Repository* can then use the *Database Context Class* to connect to the database.



1.2.3 Using The Connection String (Ad-Hoc or Stored Procedures)

When you generate code with *Ad-Hoc SQL* or *Stored Procedures*, the *Connection String* is injected to the *Data Repository* from the *Program.cs* (client). The *Data Repository* is then injected to the *Business Layer*. The *Data Repository* can then use the *Connection String* to connect to the database.



2 GENERATED SQL

We use these fields/properties to determine the type of SQL Script that will be generated.

2.1 USE LINQ-TO-ENTITIES (ENTITY FRAMEWORK CORE)

Generated SQL

☒ Use Linq-to-Entities (Entity Framework Core)

☐ Use Stored Procedures ☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☒ No Prefix or Suffix

☐ Prefix:

☐ Suffix:

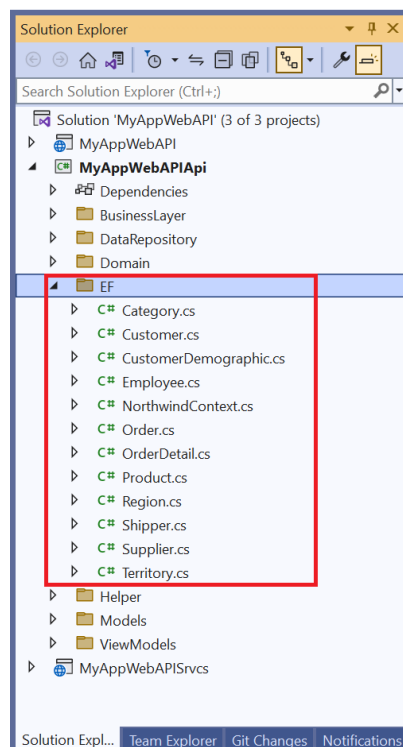
Generated SQL Script

☒ Automatic

☐ MS SQL 2008_Below

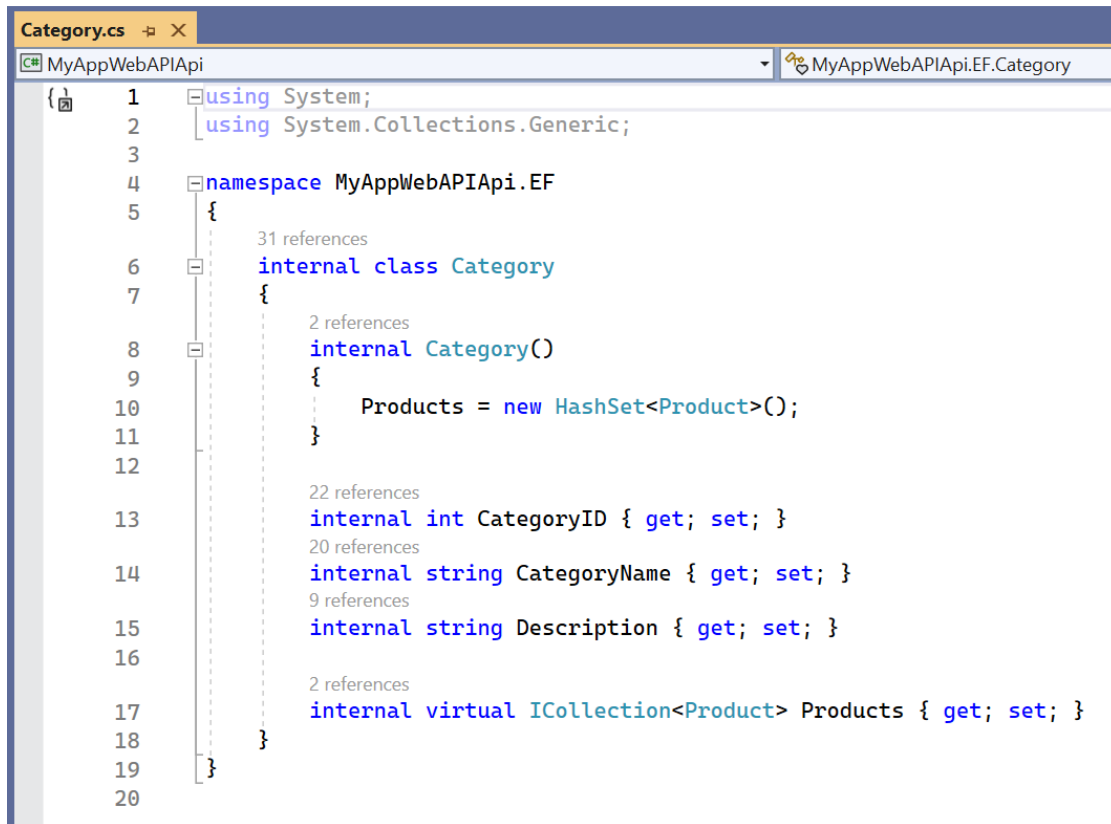
When you select *Use Linq-to-Entities (Entity Framework Core)* (default), the generated *Data Repository* code will be in an *Entity Framework Core Linq-to-Entities* format. The code will be generated in the: *Business Layer and Data Repository API Project* (Class Library Project).

- EF Directory



Generated Entity Classes

Each table selected for code generation will also have their respective class generated in the EF folder. E.g. *Categories* table will generate *Category.cs* (singular) class, etc.



Generated Category.cs Entity Class

2.2 USE STORED PROCEDURES

Generated SQL

☐ Use Linq-to-Entities (Entity Framework Core)

☒ Use Stored Procedures ☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☒ No Prefix or Suffix

☐ Prefix:

☐ Suffix:

Generated SQL Script

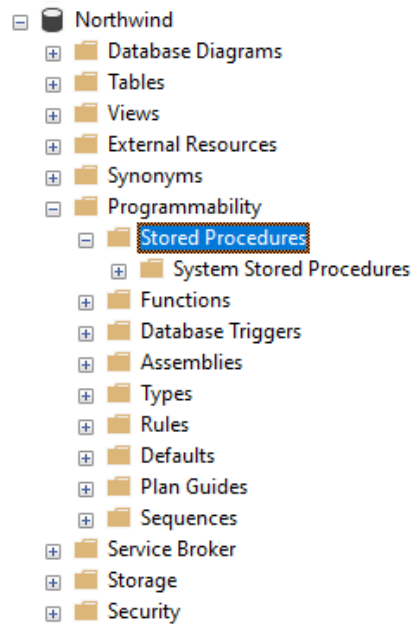
☒ Automatic

☐ MS SQL 2008 _Below

When you select *Use Stored Procedures*, the generated *SQL Script* code will be in a *Stored Procedure* form. The code will be generated in the MS SQL Server Database, in our example, SQL Scripts will be generated in the *Northwind* database. You will find the generated Stored Procedures:

Under the *Northwind* database

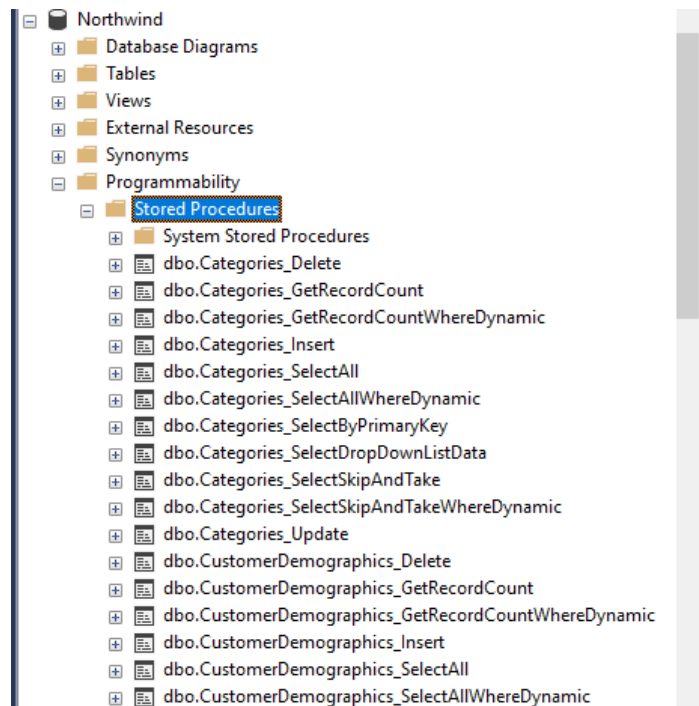
- *Programmability* folder
 - o *Stored Procedures* folder



Generated Stored Procedures will be under Stored Procedures folder

2.2.1 Use Stored Procedures – No Prefix or Suffix

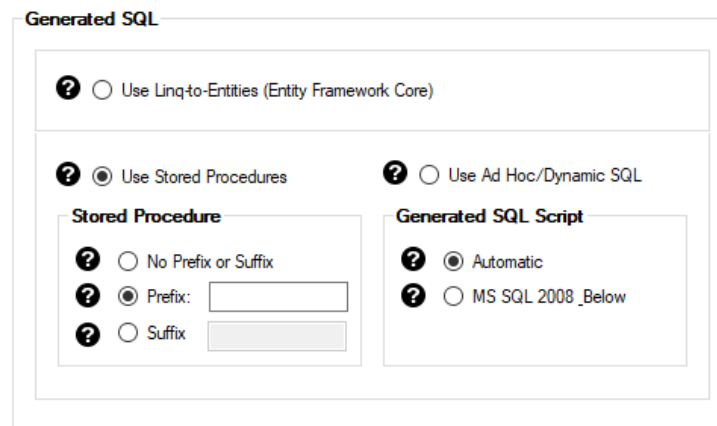
When you select *No Prefix or Suffix* (default), the generated *Stored Procedure Names* will have no prefixes or suffixes. E.g. *Categories_Delete*, *Categories_GetRecordCount*



Stored Procedure Names with No Prefix or Suffix

2.2.2 Use Stored Procedures – Prefix

When you select *Prefix*, the generated *Stored Procedure Names* will have a prefix. The *Prefix* box is **required** when *Prefix* under *Stored Procedure* is selected.



Generated SQL

☐ Use Linq-to-Entities (Entity Framework Core)

☒ Use Stored Procedures ☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☐ No Prefix or Suffix

☒ Prefix:

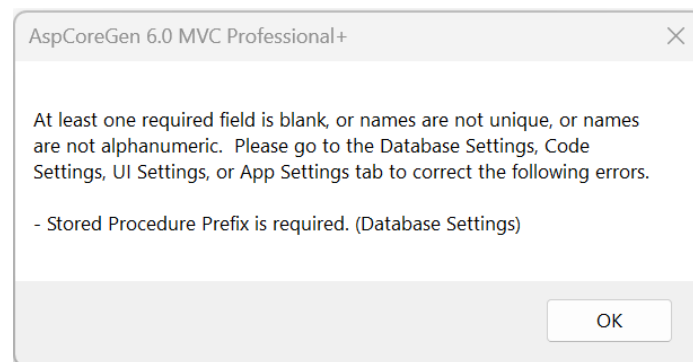
☐ Suffix:

Generated SQL Script

☒ Automatic

☐ MS SQL 2008 _Below

Prefix with Blank Box



AspCoreGen 6.0 MVC Professional+

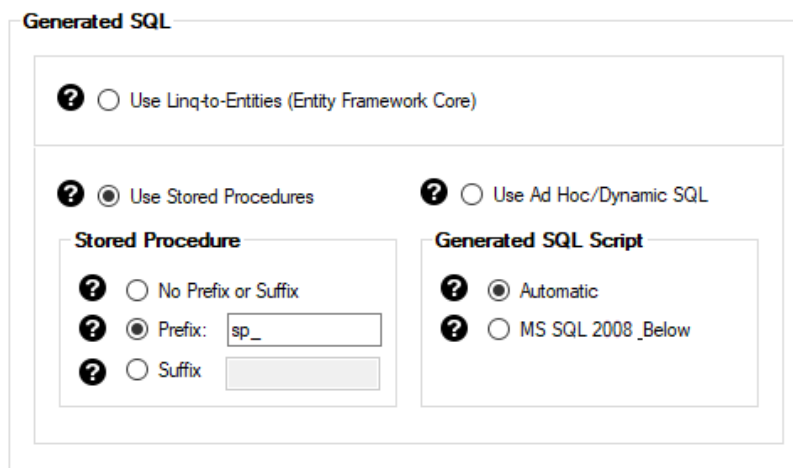
At least one required field is blank, or names are not unique, or names are not alphanumeric. Please go to the Database Settings, Code Settings, UI Settings, or App Settings tab to correct the following errors.

- Stored Procedure Prefix is required. (Database Settings)

OK

Prefix is required

The *Prefix* you enter will be used as the *Prefix* for the generated *Stored Procedure Names*. E.g. *sp_Categories_Delete*, *sp_Categories_GetRecordCount*.



Generated SQL

☐ Use Linq-to-Entities (Entity Framework Core)

☒ Use Stored Procedures ☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☐ No Prefix or Suffix

☒ Prefix:

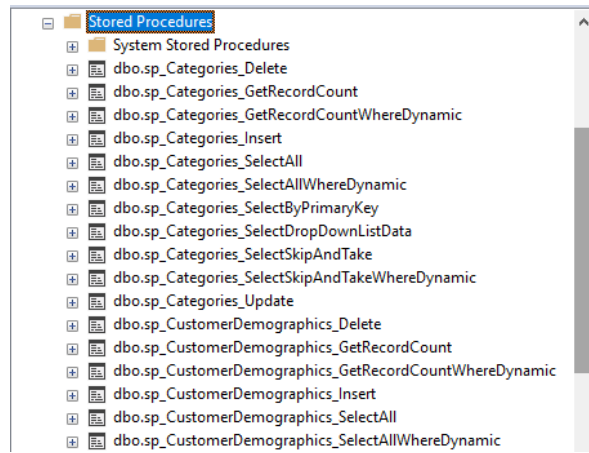
☐ Suffix:

Generated SQL Script

☒ Automatic

☐ MS SQL 2008 _Below

Prefix (filled)



Stored Procedure Names with Prefix

2.2.3 Use Stored Procedures – Suffix

When you select *Suffix*, the generated *Stored Procedure Names* will have a prefix. The *Suffix* box is **required** when *Suffix* under *Stored Procedure* is selected. The *Suffix* you enter will be used as the *Suffix* for the generated *Stored Procedure Names*. E.g. *Categories_Delete_sp*, *Categories_GetRecordCount_sp*.

Generated SQL

☐ Use Linq-to-Entities (Entity Framework Core)

☒ Use Stored Procedures ☐ Use Ad Hoc/Dynamic SQL

Stored Procedure

☐ No Prefix or Suffix

☐ Prefix:

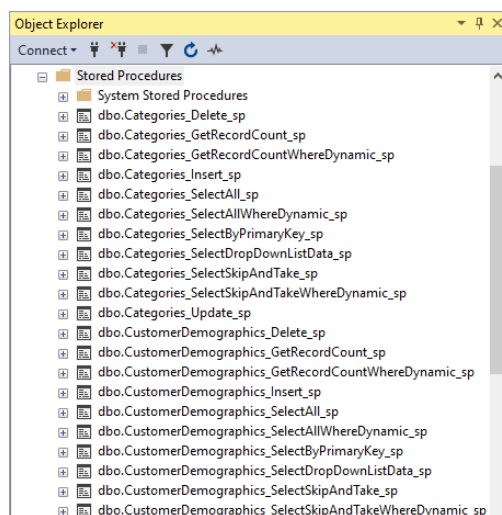
☒ Suffix

Generated SQL Script

☒ Automatic

☐ MS SQL 2008_Below

Suffix (filled)



Stored Procedure Names with Prefix

2.3 USE AD Hoc/DYNAMIC SQL

Generated SQL

☐ Use Linq-to-Entities (Entity Framework Core)

☐ Use Stored Procedures ☒ Use Ad Hoc/Dynamic SQL

Stored Procedure

☐ No Prefix or Suffix

☐ Prefix:

☐ Suffix:

Generated SQL Script

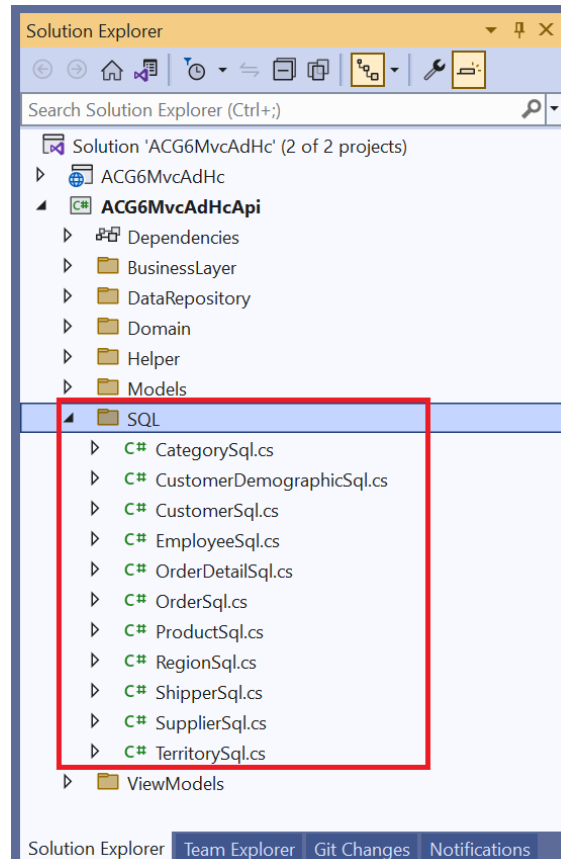
☒ Automatic

☐ MS SQL 2008 _Below

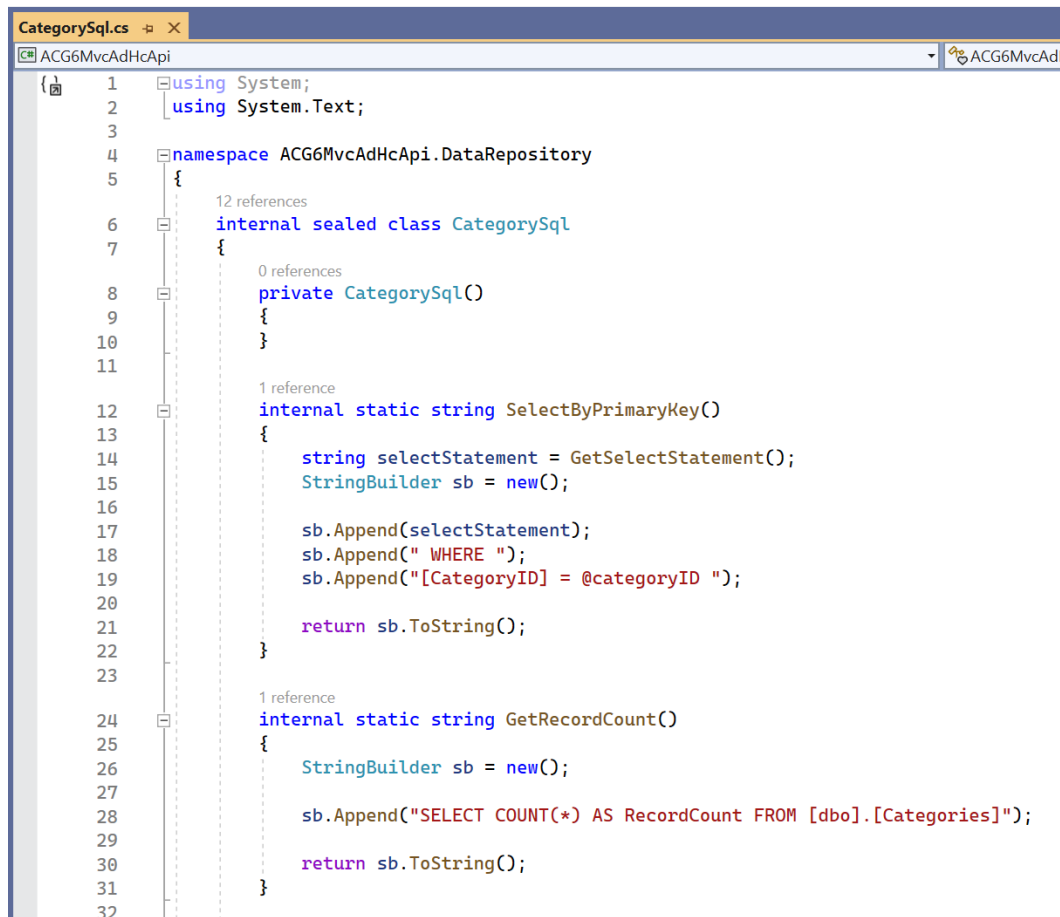
When you select *Use Ad Hoc/Dynamic SQL*, the generated *SQL Script* code will be in generated in a class file. The code will be generated in the:

Business Layer and Data Repository API Project (Class Library Project).

- SQL Directory



Generated Ad Hoc SQL Classes



```
1 using System;
2 using System.Text;
3
4 namespace ACG6MvcAdHcApi.DataRepository
5 {
6     12 references
7     internal sealed class CategorySql
8     {
9         0 references
10        private CategorySql()
11        {
12        }
13
14        1 reference
15        internal static string SelectByPrimaryKey()
16        {
17            string selectStatement = GetSelectStatement();
18            StringBuilder sb = new();
19
20            sb.Append(selectStatement);
21            sb.Append(" WHERE ");
22            sb.Append("[CategoryID] = @categoryID ");
23
24            return sb.ToString();
25        }
26
27        1 reference
28        internal static string GetRecordCount()
29        {
30            StringBuilder sb = new();
31
32            sb.Append("SELECT COUNT(*) AS RecordCount FROM [dbo].[Categories]");
33
34            return sb.ToString();
35        }
36    }
```

CategorySql.cs – Generated SQL – Ad Hoc

2.3.1 Automatic

Automatic is selected by default. This will automatically determine the MS SQL Server you're using and generate scripts based on the version. Newer generated SQL script keywords are used for MS SQL Server versions that is 2008 and above.

2.3.2 MS SQL 2008 Below

When selected, SQL script keywords used in older (older than 2008) versions are used in the generated scripts. Although these keywords may still run using the newer versions of MS SQL Server, they may not be as fast.

You can read end-to-end tutorials on more subjects on using ASPCoreGen 6.0 MVC Professional Plus that came with your purchase. These tutorials are available to customers and are included in a link on your invoice when you purchase ASPCoreGen 6.0 MVC Professional.

Note: Some features shown here are not available in the Express Edition.

End of tutorial.