public class YourDataService : IMyDataService

{

public YourDataType GetData(int idValue)

{

YourDataType result = new YourDataType();

using(SqlConnection \_con = new SqlConnection("server=(local);database=test;integrated security=SSPI;"))

{

using(SqlCommand \_cmd = new SqlCommand("YourStoredProcName", \_con))

{

\_cmd.Parameters.AddWithValue("@ID", idValue);

\_cmd.Parameters.Add("@OutStringValue", SqlDbType.VarChar, 20)

.Direction = ParameterDirection.Output;

\_cmd.Parameters.Add("@OutBoolValue", SqlDbType.Bit)

.Direction = ParameterDirection.Output;

\_cmd.ExecuteNonQuery();

result.StringValue = \_cmd.Parameters["@OutStringValue"].Value.ToString();

result.BoolValue = Convert.ToBoolean(\_cmd.Parameters["@OutBoolValue"].Value);

}

}

return result;

}

}

Using entities

|  |  |
| --- | --- |
| Connect | Check stored procedures |
|  |  |

public static List<ATT\_Tip> GetTips()

{

List<ATT\_Tip> lista = new List<ATT\_Tip>();

using (var entities = new RPAEntities())

{

var settings = entities.spGet\_ATT\_TIPS().ToList();

foreach (var item in settings)

{

ATT\_Tip ss = new ATT\_Tip();

ss.TIP = item.TIP;

ss.LINK = item.LINK;

lista.Add(ss);

}

}

return lista;

//using (var entities = new RPAEntities())

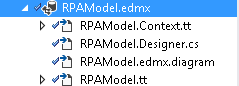
//{

// lista= entities.spGet\_ATT\_TIPS().Cast<ATT\_Tip>().ToList();

//}

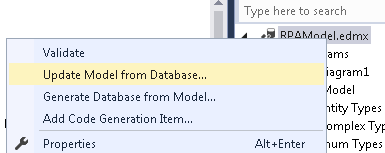
return lista;

}

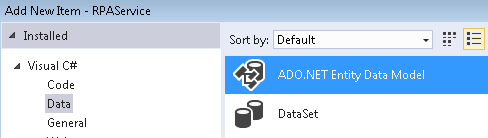


**Update the model**

* Double click file edmx
* Right click



1. Right-click the Models folder in the Solution Explorer window and select the menu option **Add, New Item**.
2. In the **Add New Item** dialog, select the Data category , select **ADO.NET Entity Data Model** template, give a name MoviesDBModel.edmx, and click Add



1. In the **Choose Model Contents** step, choose the **Generate from a database** option and click the **Next** button

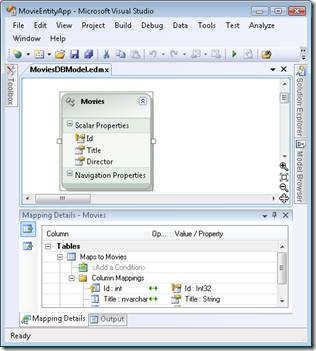
|  |  |
| --- | --- |
| 1. In the **Choose Your Data Connection** step, select the MoviesDB.mdf database connection, enter the entities connection settings name MoviesDBEntities, and click the **Next** button (see Figure 3). | clip_image006 |
|  |  |

1. In the **Choose Your Database Objects** step, select the Movie database table and click the **Finish** button

Modifying the ADO.NET Entity Data Model

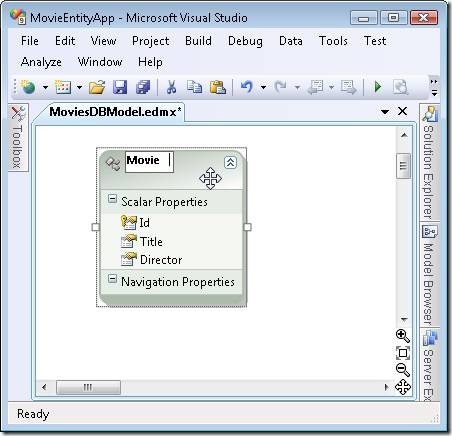
After you create an Entity Data Model, you can modify the model by taking advantage of the Entity Designer (see Figure 5). You can open the Entity Designer at any time by double-clicking the MoviesDBModel.edmx file contained in the Models folder within the Solution Explorer window.

**Figure 5 – The ADO.NET Entity Data Model Designer**



If you want to rename an entity class, you can double-click on the class name in the Entity Designer and enter a new name (see Figure 6). Alternatively, you can change the name of an entity in the Properties window after selecting an entity in the Entity Designer.

**Figure 6 – Changing an entity name**



Remember to save your Entity Data Model after making a modification by clicking the Save button (the icon of the floppy disk). Behind the scenes, the Entity Designer generates a set of C# classes. You can view these classes by opening the MoviesDBModel.Designer.cs file from the Solution Explorer window.

Don't modify the code in the Designer.cs file since your changes will be overwritten the next time you use the Entity Designer. If you want to extend the functionality of the entity classes defined in the Designer.cs file then you can create *partial classes* in separate files.