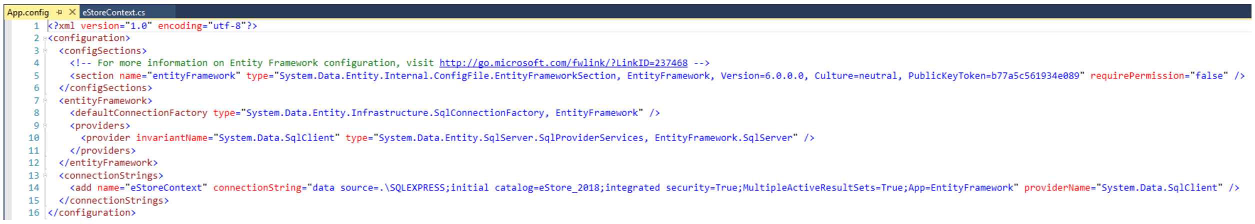
3.2.0 – Database Connectivity

Introduction

The **eStoreSystem** and the **eStoreData** projects have the basic setup but there is still one more thing to do – setup the connection to the database for the web site, **eStoreWeb**. In Lesson 3.1.0, we reverse engineered the database. In doing this, Visual Studio generated a database connection string and stored in the **App.config** file in the **eStoreData** project. This file looks like:



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
  <connectionStrings>
    <add name="eStoreContext" connectionString="data source=.\SQLEXPRESS;initial catalog=eStore_2018;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Figure 1: App.config

Database Connection String – App.config

The connection string for our system is:

```
<connectionStrings>
    <add name="eStoreContext" connectionString="data source=.\SQLEXPRESS;initial
catalog=eStore_2018;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

There are important pieces of this:

- name: this points to the **eStoreContext.cs** class file (in the **DAL** folder of the **eStoreSystem** project.
- connectionString: there are 2 important parts of this connection string:
    - data source: points to the database server we are using
    - initial catalog: points to the database we are using
- provider name: standard for a connection to a Microsoft SQL Server

Database Connection – Web site

When we created the **eStoreWeb** web site, a file called **`Web.config`** was created:

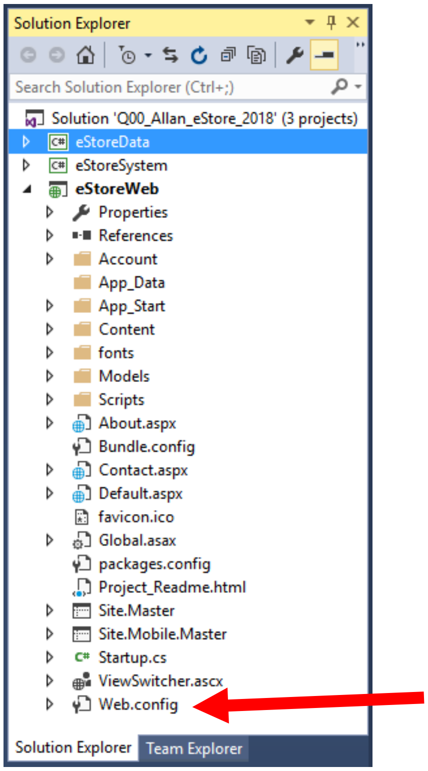
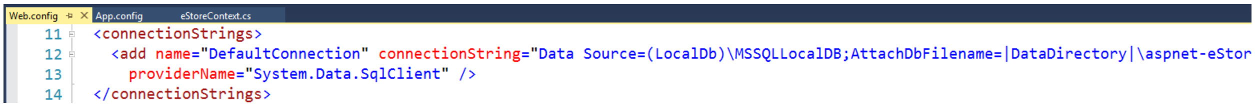


Figure 2: eStoreWeb Files

Open the **Web.config** file and find the database connection information:

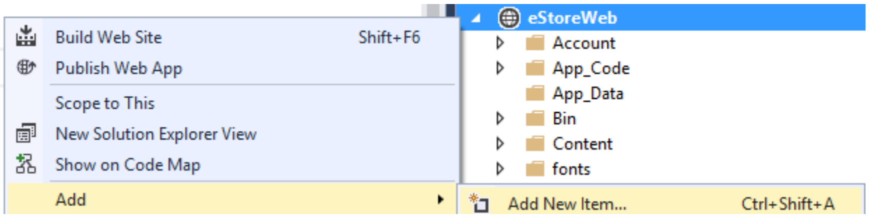


```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-eStor
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

Figure 3: Location of the Database Connection Strings

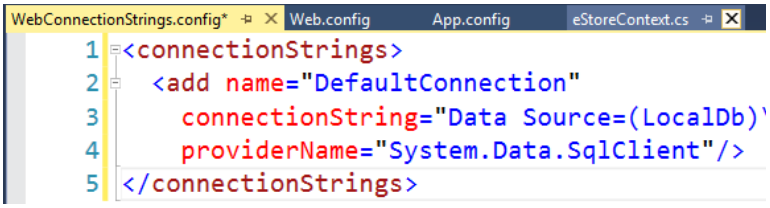
It is possible to add more database connection strings to this section of the `Web.config` file, but a better approach is to put ALL database connection strings in a separate file and then link this file to `Web.config`. Use the following steps to do this:

1. Right-click the eStoreWeb project **Add New Item…**:



2. Scroll the Add New Item wizard until you find **Web Configuration File**. Name the file `WebConnectionStrings.config` and press **Add**.
3. Delete all the code that is generated; we will add our own code to this new file.

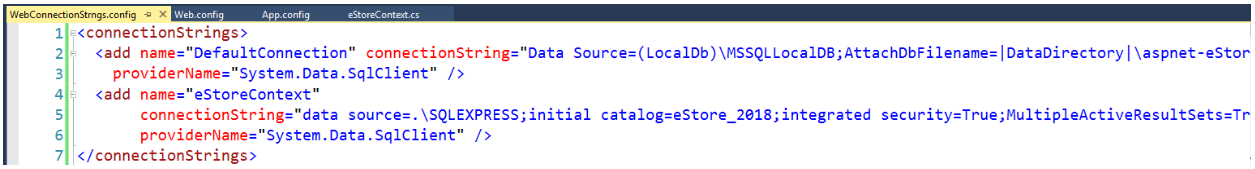
4. Copy the `<connectionStrings>` … `</connectionStrings>` block of code from the **`Web.config`** file and add it to the **`WebConnectionStrings.config`** file.



```
<connectionStrings>
  <add name="DefaultConnection"
    connectionString="Data Source=(LocalDb)'
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Figure 4: Default Connection Strings Added

5. Copy the `add name="eStoreContext"` line of code from the `<connectionStrings>` … `</connectionStrings>` block from the **`App.config`** file and add it to the **`WebConnectionStrings.config`** file. When done the **`WebConnectionStrings.config`** file should look like:



```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-eStor
    providerName="System.Data.SqlClient" />
  <add name="eStoreContext"
      connectionString="data source=.\SQLEXPRESS;initial catalog=eStore_2018;integrated security=True;MultipleActiveResultSets=Tr
      providerName="System.Data.SqlClient" />
</connectionStrings>
```

Figure 5:WebConnectionStrings.config Completed

6. Since we have a separate file, we no longer need the full connection string block in the **`Web.config`** file. We can replace the `<connectionStrings>` … `</connectionStrings>` block with the following line of code:

```
<connectionStrings configSource="WebConnectionStrings.config" />
```

7. Next, we need to tell the web site, **eStoreWeb**, about the Class Library projects. This will be done by adding a reference to the projects (see Lesson 2.3.0 on how to add a reference to a code project). We will add a reference to both Class Library projects (**eStoreSystem** and **eStoreData**). We may get a warning that one of the references will not be added; **eStoreSystem** already has a reference to **eStoreData**. When added, you should see the following in the Solution Explorer:

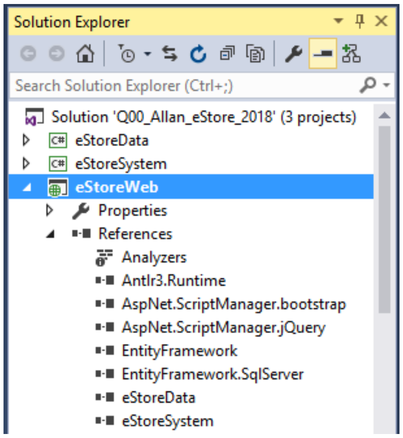
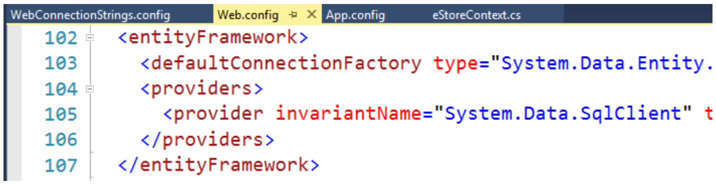

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Q00_Allan_eStore_2018' (3 projects)
eStoreData
eStoreSystem
eStoreWeb
Properties
References
Analyzers
Antlr3.Runtime
AspNet.ScriptManager.bootstrap
AspNet.ScriptManager.jQuery
EntityFramework
EntityFramework.SqlServer
eStoreData
eStoreSystem

Figure 6: Class Library References Added

8. Although the modification above to **`Web.config`** will generally be good enough we should also make one further modification. We need to disable the dynamic creation of the database. Scroll down in the **`Web.config`** file to find the `<entityFramework>` section.



```
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.
  <providers>
    <provider invariantName="System.Data.SqlClient" t
  </providers>
</entityFramework>
```

Figure 7: Entity Framework Section in Web.config

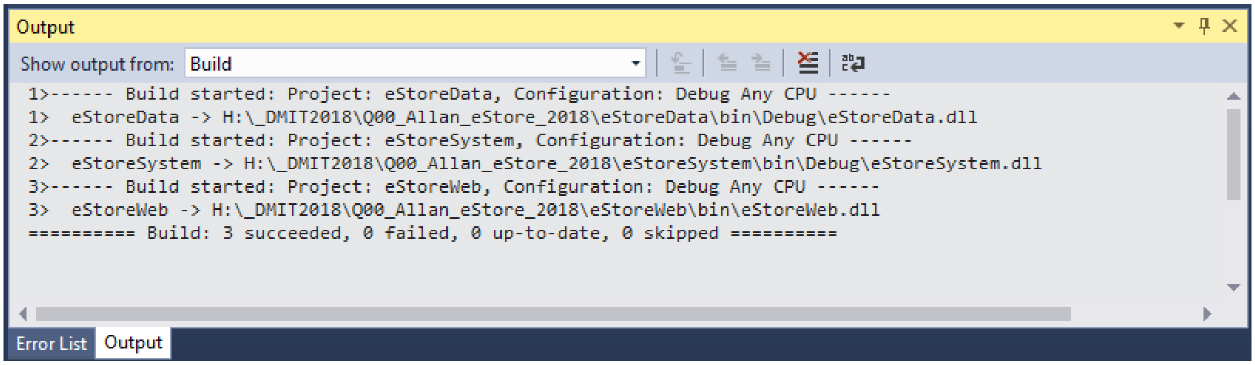
9. Add the code, below, on a new line below the `<entityFramework>` line:

```
<!-- To prevent the dynamic creation of a database -->
<contexts>
  <context type="eStoreSystem.DAL.eStoreContext, eStoreSystem"
disableDatabaseInitialization="true" />
</contexts>
```

10. **If you make a mistake in any of the names of the `context`, Visual Studio will connect to the database server, look for the database but not find it, and will create a new database for you.**

Build Solution

With all this work completed, it is time to build your solution. Your Output window should show something like that shown below. It is only important that you get `0 failed`.



```
1>------ Build started: Project: eStoreData, Configuration: Debug Any CPU ------
1>  eStoreData -> H:\_DMIT2018\Q00_Allan_eStore_2018\eStoreData\bin\Debug\eStoreData.dll
2>------ Build started: Project: eStoreSystem, Configuration: Debug Any CPU ------
2>  eStoreSystem -> H:\_DMIT2018\Q00_Allan_eStore_2018\eStoreSystem\bin\Debug\eStoreSystem.dll
3>------ Build started: Project: eStoreWeb, Configuration: Debug Any CPU ------
3>  eStoreWeb -> H:\_DMIT2018\Q00_Allan_eStore_2018\eStoreWeb\bin\eStoreWeb.dll
========== Build: 3 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Figure 8: Successful Build

If you have any errors, they need to be fixed before continuing with the next lesson.

Exercise

Complete Exercise 3.2.1.