# Using a Configuration File to Define a Data Source

**Visual Studio 2015**

This walkthrough illustrates how to use a data source defined in an *app.config* file for unit testing. You will learn how to create an *app.config* file that defines a data source that can be used by the DataSourceAttribute class. Tasks presented in this walkthrough include the following:

- Creating an app.config file.
- Defining a custom configuration section.
- Defining connection strings.
- Defining the data sources.
- Accessing the data sources using the *DataSourceAttribute* class.

## Prerequisites

To complete this walkthrough, you will need:

- Visual Studio Premium or Visual Studio Ultimate
- Either Microsoft Access or Microsoft Excel to provide data for at least one of the test methods.
- A Visual Studio 2015 solution that contains a test project.

## Create the App.config File

**To add an *app.config* file to the project**

1. If your test project already has an *app.config* file, go to Define a Custom Configuration Section.
2. Right-click your test project in the Solution Explorer, point to Add, and then click New Item.

   The Add New Item window opens.

3. Select the *Application Configuration File* template and click Add.

## Define a Custom Configuration Section

Examine the *app.config* file. It contains at least the XML declaration and a root element.

**To add the custom configuration section to the *app.config* file**

1. The root element of *app.config* should be the configuration element. Create a configSections element within the configuration element. The *configSections* should be the first element in the *app.config* file.
2. Within the *configSections* element, create a section element.
3. In the section element, add an attribute called name and assign it a value equal microsoft.visualstudio.testtools.

   Add another attribute called type and assign it a value equal:

```
Microsoft.VisualStudio.TestTools.UnitTesting.TestConfigurationSection,
Microsoft.VisualStudio.QualityTools.UnitTestFramework, Version=10.0.0.0,
Culture=neutral
```

The section element should look similar to this:

```
<section name="microsoft.visualstudio.testtools"
type="Microsoft.VisualStudio.TestTools.UnitTesting.TestConfigurationSection,
Microsoft.VisualStudio.QualityTools.UnitTestFramework, Version=10.0.0.0,
Culture=neutral"/>
```

| **Note** |
| --- |
| The assembly name must match the Microsoft Visual Studio .NET Framework build that you are using. Set the Version to 9.0.0.0 if you are using the Visual Studio .NET Framework 3.5. If you are using the Visual Studio .NET Framework 2.0, set the Version to 8.0.0.0. |

## Define Connection Strings

The connection strings define provider specific information for accessing data sources. Connection strings defined in configuration files provide reusable data provider information across an application. In this section, you create two connection strings that will be used by data sources that are defined in the Custom Configuration Section.

**To define connection strings**

1. After the *configSections* element, create a *connectionStrings* element.
2. Within the *connectionStrings* element, create two add elements.
3. In the first add element, create the following attributes and values for a connection to a **Microsoft Access** database:

| Attribute | Values |
| --- | --- |
| name | "MyJetConn" |
| connectionString | "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=.\testdatasource.accdb; Persist Security Info=False;" |
| providerName | "System.Data.OleDb" |

In the second add element, create the following attributes and values for a connection to a **Microsoft Excel** spreadsheet:

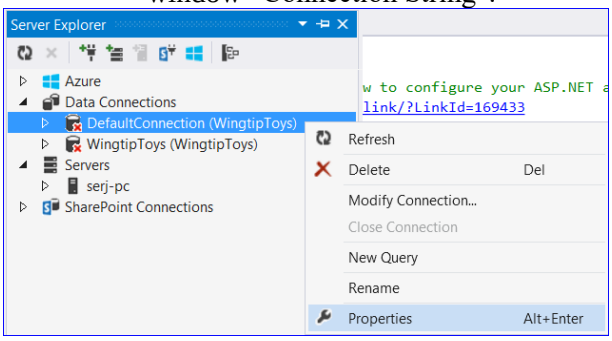
| | |
| --- | --- |
| name | "MyExcelConn" |
| connectionString | "Dsn=Excel Files;dbq=.\data.xlsx;defaultdir=.;driverid=1046;maxbuffersize=2048;pagetimeout=5" |
| providerName | "System.Data.Odbc" |

The ***connectionStrings*** element should look similar to this:

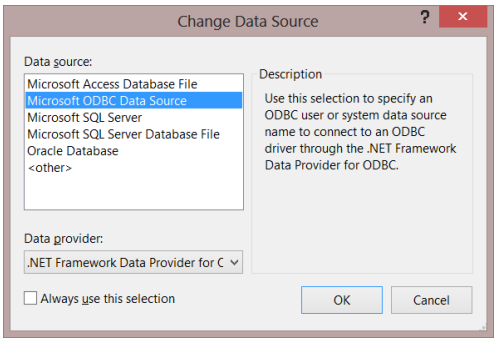
```
<connectionStrings>
    <add name="MyJetConn" connectionString="Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=.\testdatasource.accdb; Persist Security Info=False;"
providerName="System.Data.OleDb" />
    <add name="MyExcelConn" connectionString="Dsn=Excel
Files;dbq=.\data.xlsx;defaultdir=.;driverid=1046;maxbuffersize=2048;pagetimeout=5"
providerName="System.Data.Odbc" />
</connectionStrings>
```

**Notes: To see *connectionString* perform the following:**

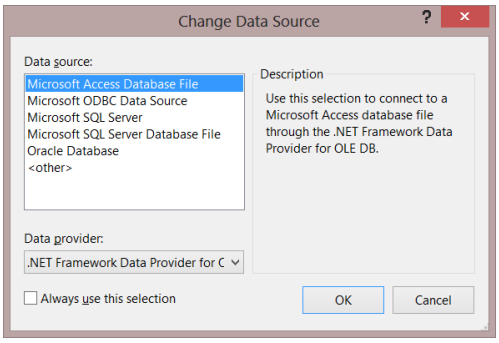
- In Server Explorer -> right click on Data Connection -> Add Connection…
- Define in new window:
  - First option (see the picture below):
    - Server Explorer -> Data Connection -> right click on the DB -> Properties. See in Properties window “Connection String”.



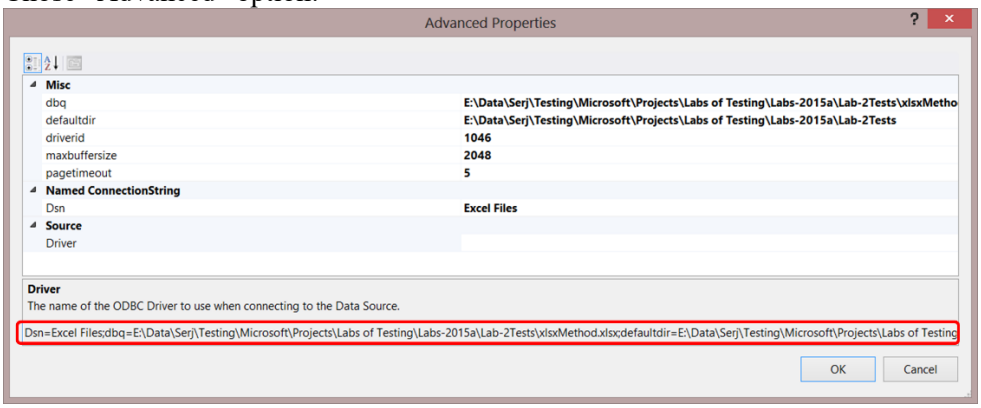

  - Option two:
    - For Excel:




For Access:




- Define other parameters;
- Check “Test Connection”;
- Chose “Advanced” option:




## Define Data Sources

The data sources section contains four attributes that are used by the test engine to retrieve data from a data source.

- name defines the identity used by the DataSourceAttribute to specify which data source to use.
- connectionString identifies the connection string created in the previous Define Connection Strings section.

- dataTableName defines the table or sheet that holds the data to use in the test.
- dataAccessMethod defines the technique for accessing data values in the data source.

In this section, you will define two data sources to use in a unit test.

**To define data sources**

1. After the *connectionStrings* element, create a *microsoft.visualstudio.testtools* element. This section was created in Define a Custom Configuration Section.
2. Within the microsoft.visualstudio.testtools element, create a *dataSources* element.
3. Within the *dataSources* element, create two add elements.
4. In the first add element, create the following attributes and values for a **Microsoft Access** data source:

| Attribute | Values |
|---|---|
| name | "MyJetDataSource" |
| connectionString | "MyJetConn" |
| dataTableName | "MyDataTable" |
| dataAccessMethod | "Sequential" |

In the second add element, create the following attributes and values for a **Microsoft Excel** data source:

| | |
|---|---|
| Name | "MyExcelDataSource" |
| connectionString | "MyExcelConn" |
| dataTableName | "Sheet1$" |
| dataAccessMethod | "Sequential" |

The microsoft.visualstudio.testtools element should look similar to this:

```
<microsoft.visualstudio.testtools>
    <dataSources>
        <add name="MyJetDataSource" connectionString="MyJetConn"
dataTableName="MyDataTable" dataAccessMethod="Sequential"/>
        <add name="MyExcelDataSource" connectionString="MyExcelConn"
dataTableName="Sheet1$" dataAccessMethod="Sequential"/>
    </dataSources>
</microsoft.visualstudio.testtools>
```

The final *app.config* file should look similar to this:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="microsoft.visualstudio.testtools"
type="Microsoft.VisualStudio.TestTools.UnitTesting.TestConfigurationSection,
Microsoft.VisualStudio.QualityTools.UnitTestFramework, Version=10.0.0.0,
Culture=neutral"/>
  </configSections>
  <connectionStrings>
    <add name="MyJetConn" connectionString="Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=.\Testdatasource.accdb; Persist Security Info=False;"
providerName="System.Data.OleDb" />
```

```
    <add name="MyExcelConn" connectionString="Dsn=Excel
Files;dbq=.\data.xlsx;defaultdir=.;driverid=1046;maxbuffersize=2048;pagetimeout=5"
providerName="System.Data.Odbc" />
  </connectionStrings>
  <microsoft.visualstudio.testtools>
    <dataSources>
      <add name="MyJetDataSource" connectionString="MyJetConn"
dataTableName="MyDataTable" dataAccessMethod="Sequential"/>
      <add name="MyExcelDataSource" connectionString="MyExcelConn"
dataTableName="Sheet1$" dataAccessMethod="Sequential"/>
    </dataSources>
  </microsoft.visualstudio.testtools>
</configuration>
```

## Create a Unit Test Using Data Sources Defined in *app.config*

Now that an *app.config* file has been defined, you will create a unit test that uses data located in the data sources that are defined in the app.config file. In this section, we will:

- Create the data sources found in the app.config file.
- Use the data sources in two test methods that compare the values in each data source.

**To create a Microsoft Access data source**

1. Create a Microsoft Access database named testdatasource.accdb.
2. Create a table and name it MyDataTable in testdatasource.accdb.
3. Create two fields in MyDataTable named Arg1 and Arg2 using the Number data type.
4. Add five entities to MyDataTable with the following values for Arg1 and Arg2, respectively: (10,50), (3,2), (6,0), (0,8) and (12312,1000).
5. Save and close the database.
6. Change the connection string to point to the location of the database. Change the value of Data Source to reflect the location of the database.

**To create a Microsoft Excel data source**

1. Create a Microsoft Excel spreadsheet named data.xlsx.
2. Create a sheet named *Sheet1* if it does not already exist in data.xlsx.
3. Create two column headers and name them Val1 and Val2 in Sheet1.
4. Add five entities to Sheet1 with the following values for Val1 and Val2, respectively: (1,1), (2,2), (3,3), (4,4) and (5,0).
5. Save and close the spreadsheet.
6. Change the connection string to point to the location of the spreadsheet. Change the value of *dbq* to reflect the location of the spreadsheet.

**To create a unit test using the *app.config* data sources**

1. Add a unit test to the test project.

   For more information, see Creating and Running Unit Tests for Existing Code.

2. Replace the auto-generated contents of the unit test with the following code:

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace TestProject1
{
     [TestClass]
    public class UnitTest1
    {
        private TestContext context;

        public TestContext TestContext
        {
            get { return context; }
            set { context = value; }
        }

        [TestMethod()]
        [DeploymentItem("MyTestProject\\testdatasource.accdb")]
        [DataSource("MyJetDataSource")]
        public void MyTestMethod()
        {
            int a = Int32.Parse(context.DataRow["Arg1"].ToString());
            int b = Int32.Parse(context.DataRow["Arg2"].ToString());
            Assert.AreNotEqual(a, b, "A value was equal.");
        }

        [TestMethod()]
        [DeploymentItem("MyTestProject\\data.xlsx")]
        [DataSource("MyExcelDataSource")]
        public void MyTestMethod2()
        {
            Assert.AreEqual(context.DataRow["Val1"], context.DataRow["Val2"]);
        }
    }
}
```

3. Examine the *DataSource* attributes. Notice the setting names from the *app.config* file.
4. Build your solution and run *MyTestMethod* and *MyTestMethod2* tests.

| **Important** |
| --- |
| Deploy items like data sources so that they are accessible to the test in the deployment directory. |