Deploy an application with a database connection string – Using the **App.config** file

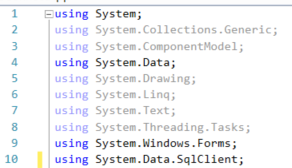
Open the Application **DBConnectApp.** Copy the two databases fie into the application folder.

In the **LoginForm.cs**

Add the following:

using System.Data;

using System.Data.SqlClient;



Add the following code to the **btnConnect\_Click** event button.

private void btnConnect\_Click(object sender, EventArgs e)

Change the path to suit your .mdf located in your applications location.

{

string username, password;

username = txtUserName.Text;

password = txtPassword.Text;

string tableName = "[userlogin]";

SqlConnection connection = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = D:\MyTafe\2018\_S1\5TSD\Resources\Session13\Student\_Files\DBConnectApp\DBConnectApp\DBConnectApp\LoginDB.mdf; Integrated Security = True; Connect Timeout = 30");

string sql = "SELECT \* FROM " + tableName + " WHERE user\_Name= @username AND user\_password = @password";

try

{

SqlCommand sqlCommand = new SqlCommand(sql, connection);

sqlCommand.CommandText = sql;

SqlParameter UsernameParametar = new SqlParameter("@username", SqlDbType.VarChar);

SqlParameter PassParametar = new SqlParameter("@password", SqlDbType.VarChar);

sqlCommand.Parameters.Add(UsernameParametar);

sqlCommand.Parameters.Add(PassParametar);

UsernameParametar.Value = @username;

PassParametar.Value = @password;

connection.Open();

SqlDataReader sqlReader = sqlCommand.ExecuteReader();

MessageBox.Show("Login Successful");

if (sqlReader.HasRows)

connection.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error - try again" + ex.Message);

txtUserName.Focus();

return;

}

}

Run the application.

This way of putting the database connection string is not good. It should be put at the **App.confg** file. The program should reference with the database parameters.

Let’s modify the program.

Open the **App.config file**. It should look like the following:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

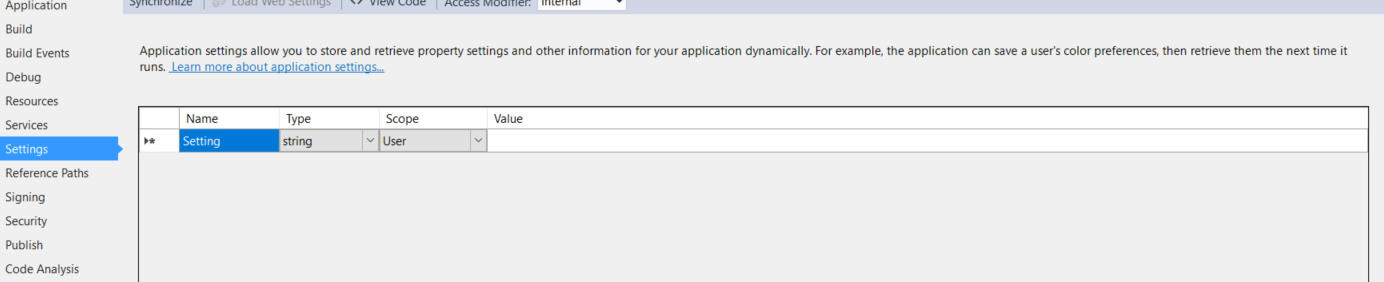
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />

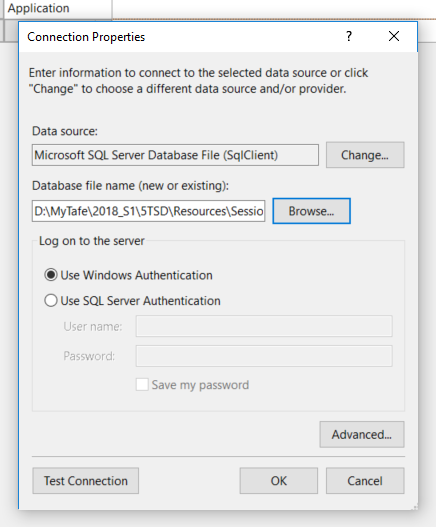
</startup>

</configuration>

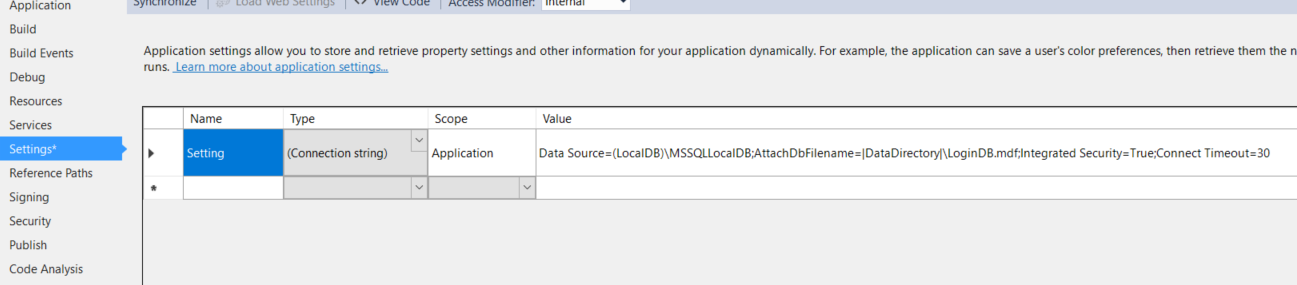
There is no database connection string in it yet.

Open the property of the project. Select the

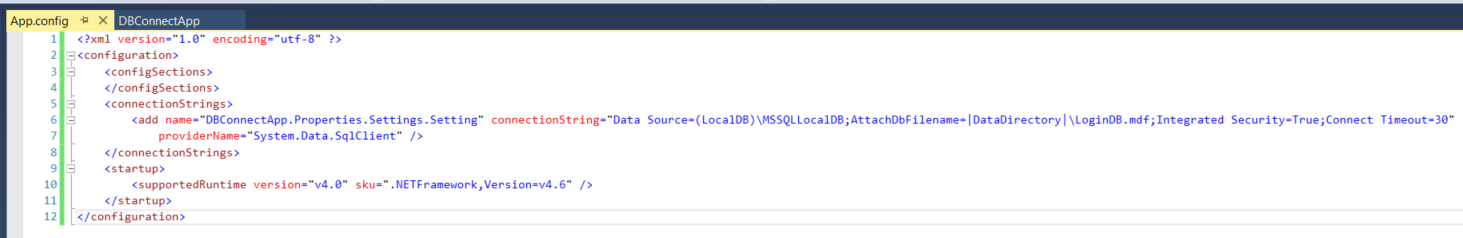




Browse into the LoginDB.mdf file.



Open the **App.config** file again, you should be able to see the following. It generated the connection string into the **App.config** file.

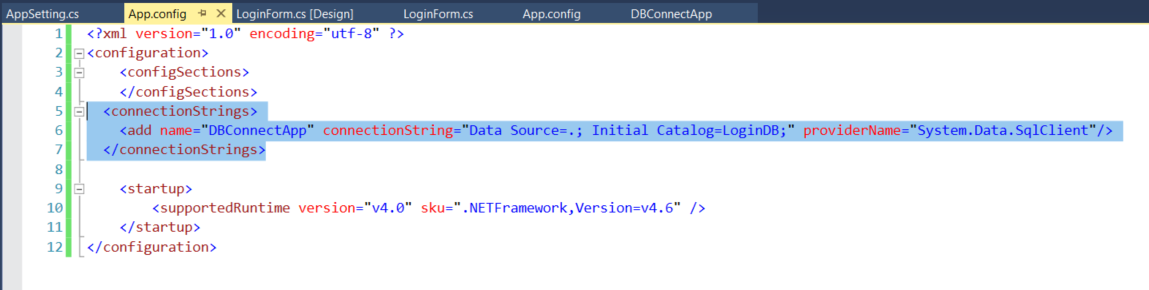


Add this to the App.config file:

<connectionStrings>

<add name="DBConnectApp" connectionString="Data Source=.; Initial Catalog=LoginDB;" providerName="System.Data.SqlClient"/>

</connectionStrings>



using System.Configuration;

Add reference to System.Configuration

In the btnConnect\_Click event, incorporate the following statements:

public bool verifier(string tableName, string username, string password)

{

bool success = false;

ConnectionStringSettings connSettings = ConfigurationManager.ConnectionStrings["DBConnectApp"];

string connectionString = connSettings.ConnectionString;

string sql = "SELECT \* FROM " + tableName + " WHERE user\_name=@userName AND pass\_word=@password";

try

{

conn = new SqlConnection(connectionString);

conn.Open();

cmd = new SqlCommand(sql, conn);

cmd.CommandText = sql;

SqlParameter UsernameParametar = new SqlParameter("@username", SqlDbType.VarChar);

SqlParameter PassParametar = new SqlParameter("@password", SqlDbType.VarChar);

cmd.Parameters.Add(UsernameParametar);

cmd.Parameters.Add(PassParametar);

UsernameParametar.Value = username;

PassParametar.Value = password;

dr = cmd.ExecuteReader();

if (dr.HasRows)

success = true;

conn.Close();

}

catch (Exception e)

{

MetroFramework.MetroMessageBox.Show(this, e.ToString(), "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

}

finally

{

conn.Close();

}

return success;

}