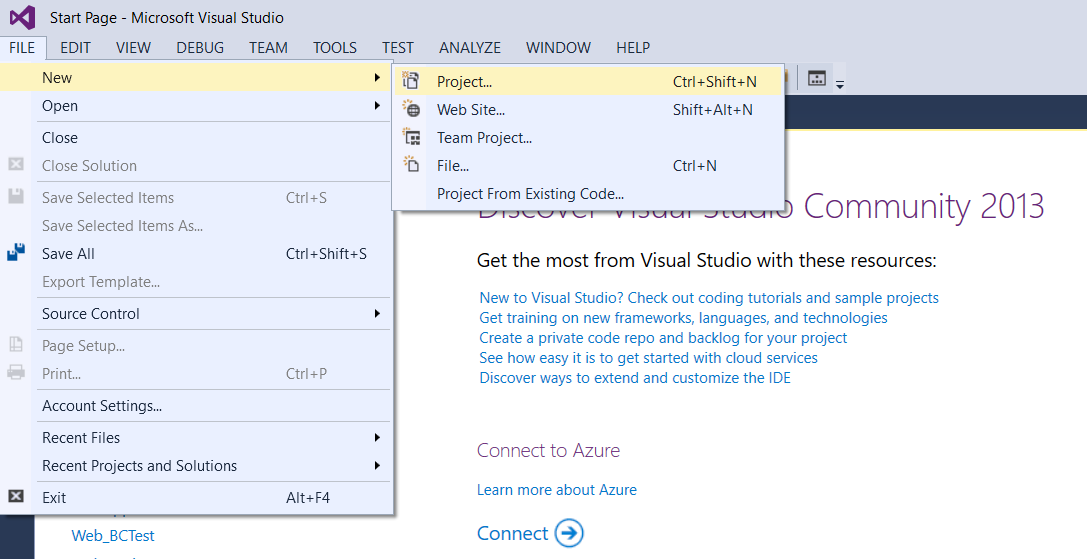
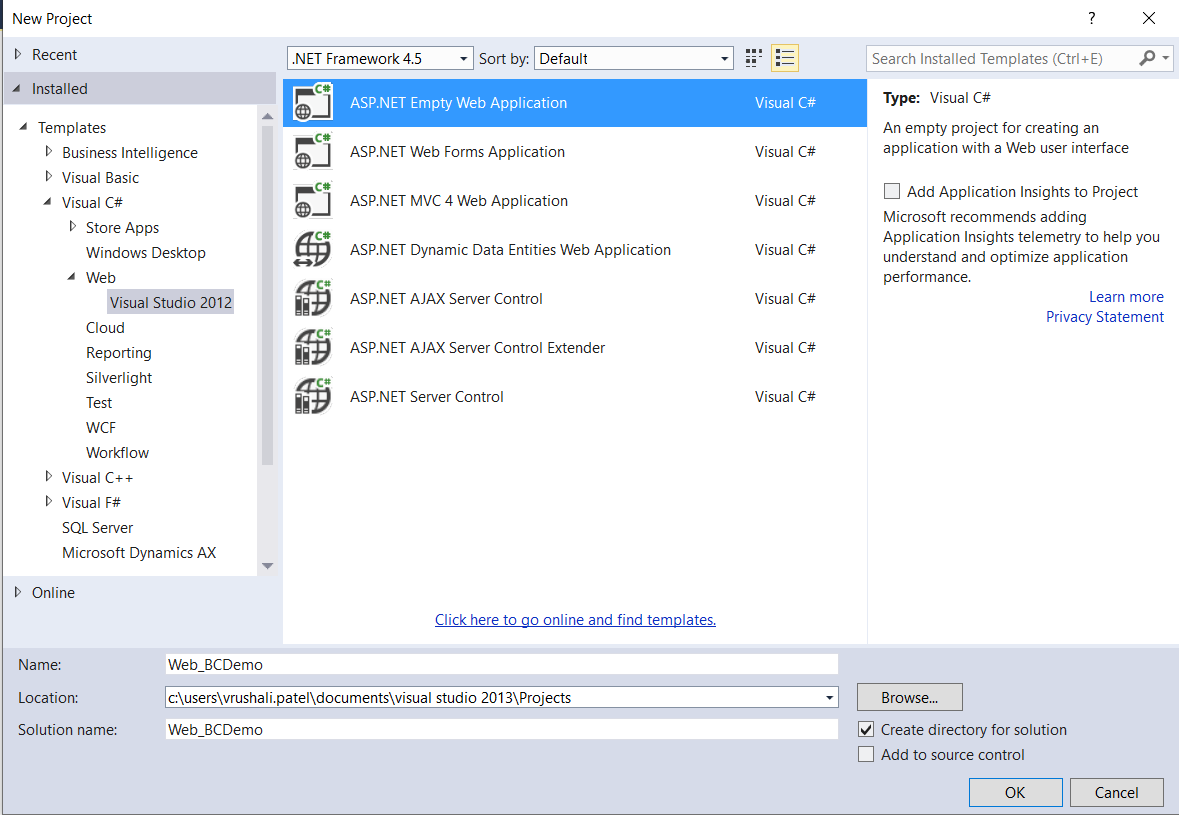
**Access data from AX using business connector in .Net Application**

**Create .Net application and get data from AX using business connector**

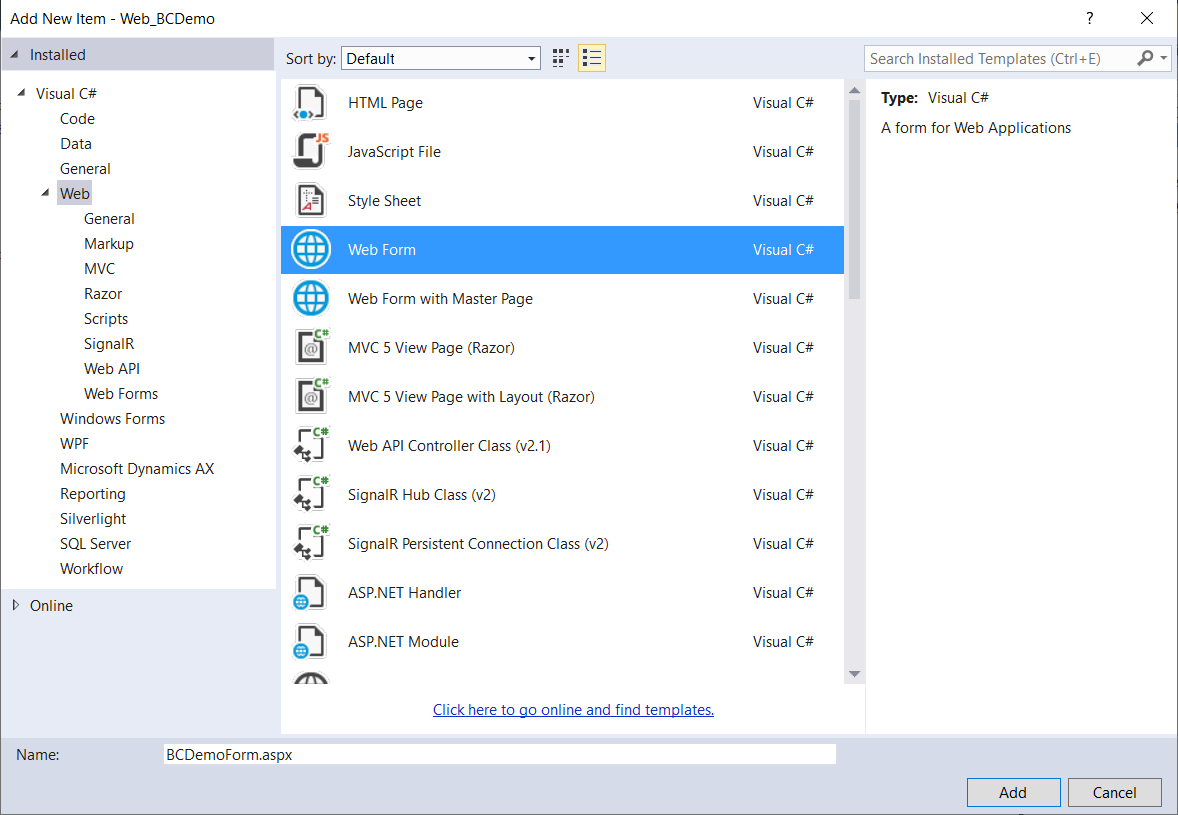
Open Visual studio and select new project option form File >> New menu.



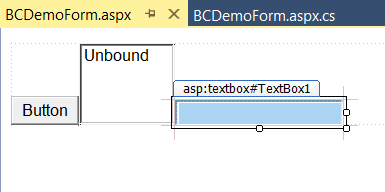
Select ASP.NET empty web application from the available option and save it.



Add a new Web form in your project and give it a name.

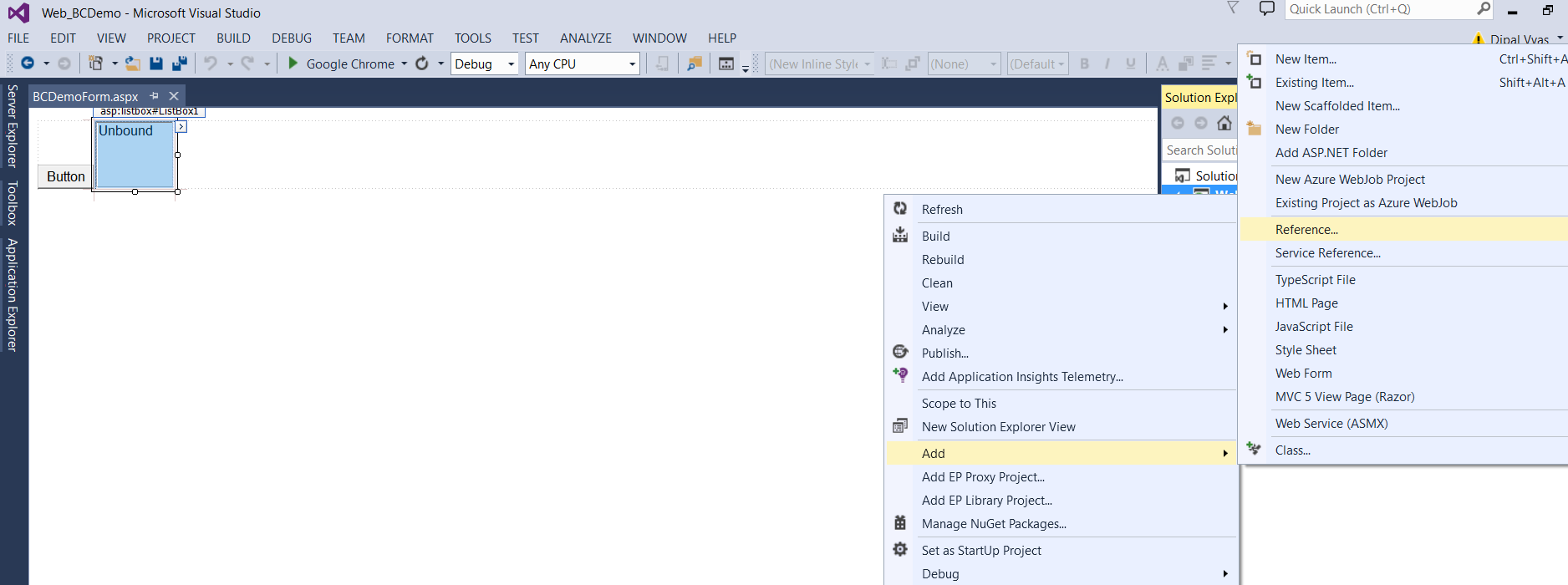


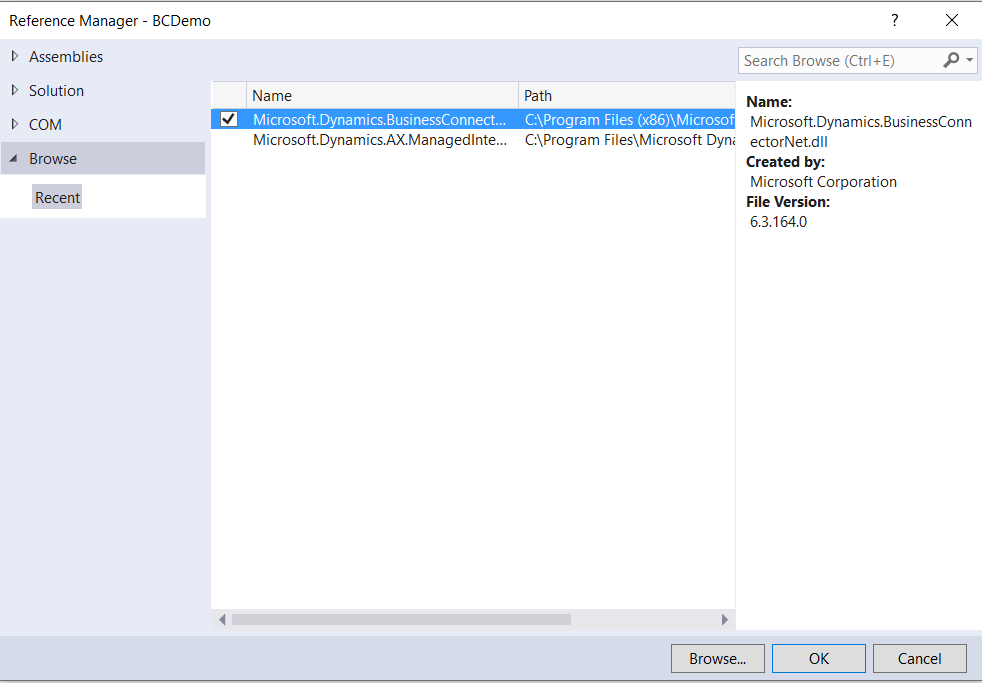
Add one Button, List box and Text box in the web form.



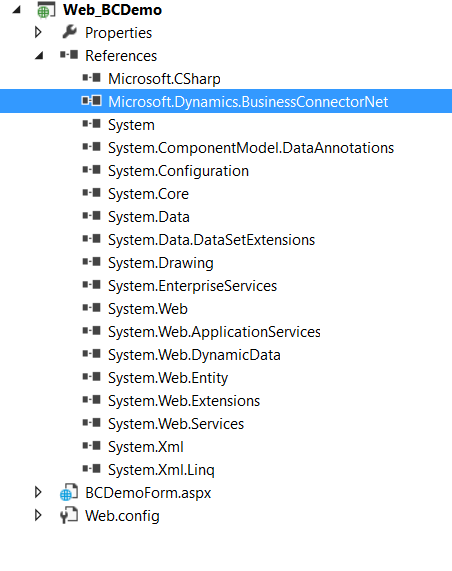
To use .Net business connector, we will have to add reference of business connectors’ dll file. This dll file can be added from the below path.

C:\Program Files (x86)\Microsoft Dynamics AX\60\Client\Bin





After adding the reference of the dll file, you can see it in your project as shown in below screen.



Now you have to add below two lines your code to use business connector.

using Microsoft.Dynamics.BusinessConnectorNet;

using System.Data;

Create a method and write code as below.

public void callmethod()

{

Axapta ax;

AxaptaRecord axRecord;

string tableName = "CustTable";

string strNameField = "AccountNum";

string strCustGroupField = "CustGroup";

object fieldName,fieldCustGroup;

try

{

ax = new Axapta();

ax.Logon("USMF", "en-us", "TAURUS", null);

using (axRecord = ax.CreateAxaptaRecord(tableName))

{

axRecord.ExecuteStmt("select \* from %1 ");

ListBox1.Items.Clear();

while (axRecord.Found)

{

fieldName = axRecord.get\_Field(strNameField);

fieldCustGroup = axRecord.get\_Field(strCustGroupField);

ListBox1.Items.Add(fieldName.ToString()+","+fieldCustGroup);

axRecord.Next();

}

}

}

catch (Exception e)

{

TextBox1.Text = e.Message;

}

}

**Call this method in the Click event of the button.**

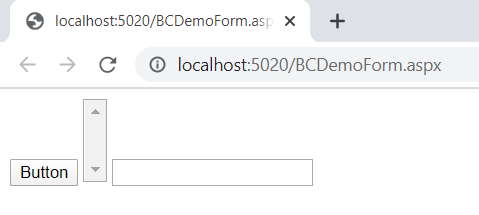
protected void Button1\_Click(object sender, EventArgs e)

{

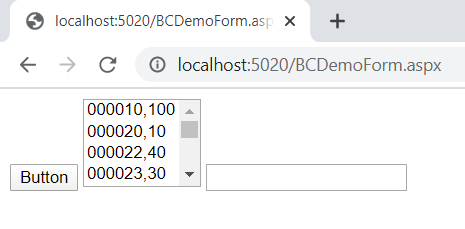
this.callmethod();

}

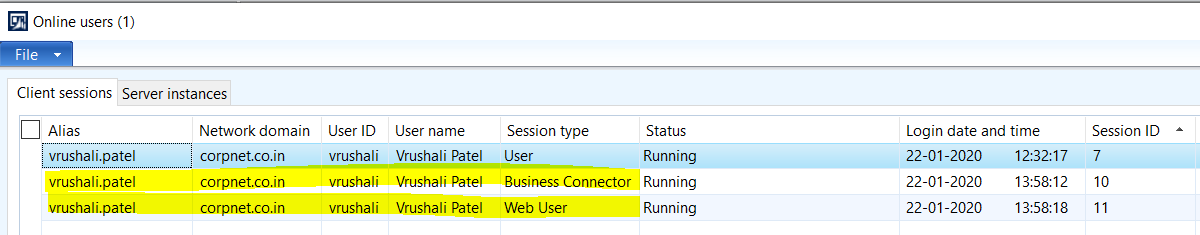
Run the application.



Click on the button. After clicking the button, the data will be retrieved from AX and displayed in the list box as shown in below screen.



After running the application, one session of “Web User” and “other session of “Business Connector” have been added in AX. You can see it from the Online Users form in AX.



We cannot leave the Web user session open after completing the task. So we will need to close the “Web User” session via coding.

Code to Close the Web User session

ax.Logoff();

ax.Dispose();

After adding the code of session close, the method will look like as below.

public void callmethod()

{

Axapta ax;

AxaptaRecord axRecord;

string tableName = "CustTable";

string strNameField = "AccountNum";

string strCustGroupField = "CustGroup";

object fieldName,fieldCustGroup;

try

{

ax = new Axapta();

ax.Logon("USMF", "en-us", "TAURUS", null);

using (axRecord = ax.CreateAxaptaRecord(tableName))

{

axRecord.ExecuteStmt("select \* from %1 ");

ListBox1.Items.Clear();

while (axRecord.Found)

{

fieldName = axRecord.get\_Field(strNameField);

fieldCustGroup = axRecord.get\_Field(strCustGroupField);

ListBox1.Items.Add(fieldName.ToString()+","+fieldCustGroup);

axRecord.Next();

}

}

ax.Logoff();

ax.Dispose();

}

catch (Exception e)

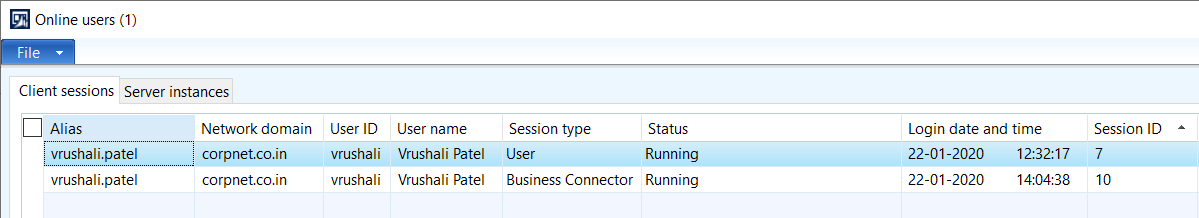
{

TextBox1.Text = e.Message;

}

}

Now you can see in the below screen, that the “Web User” session has been closed after running the form.



**Access AX with different user other than the logged in user in .net application.**

System.Net.NetworkCredential credential = new System.Net.NetworkCredential("UserID", "Password", "corpnet");

ax.LogonAs("UserID", "NetworkDomain", credential, "AXCompany", "en-us", "AXServer", null);

**Create a method in AX class and return data from the method.**

**Create class in AX.**

**class** Dev\_WebServiceBCdemo

{

}

Add a new method as below in this class. We will pass the parameter from the .net application and get data from AX using this parameter.

**public** **static** dev\_WebServiceDemo GetVendorDetailsFromEmail(**str** **100** strVendEmailId)

{

VendTable objVendTable1;

LogisticsElectronicAddress objLogisticsElectronicAddress1;

DirPartyLocation objDirPartyLocation1;

dev\_WebServiceDemo objdev\_WebServiceDemo\_Insert;

**while** **select** Locator **from** objLogisticsElectronicAddress1 **join** objDirPartyLocation1

**where** objLogisticsElectronicAddress1.Location == objDirPartyLocation1.Location

&& objLogisticsElectronicAddress1.Type == LogisticsElectronicAddressMethodType::Email

&& objLogisticsElectronicAddress1.Locator == strVendEmailId

**join** AccountNum **from** objVendTable1 **order** **by** objvendTable1.AccountNum

**where** objDirPartyLocation1.Party == objVendTable1.Party

{

objdev\_WebServiceDemo\_Insert.clear();

objdev\_WebServiceDemo\_Insert.AccountNum=objVendTable1.AccountNum;

objdev\_WebServiceDemo\_Insert.EmailID=objLogisticsElectronicAddress1.Locator;

objdev\_WebServiceDemo\_Insert.insert();

}

**return** objdev\_WebServiceDemo\_Insert;

}

Create below method in .Net application.

public void callAXClassMethod(string \_VendorEmail)

{

Axapta ax1;

AxaptaRecord axRecord;

object fieldName, fieldName1;

ax1 = new Axapta();

ax.LogonAs("UserID", "NetworkDomain", credential, "AXCompany", "en-us", "AXServer", null);

axRecord = (AxaptaRecord)ax1.CallStaticClassMethod("Dev\_WebServiceBCdemo", "GetVendorDetailsFromEmail", \_VendorEmail);

axRecord.ExecuteStmt("select \* from %1 ");

ListBox2.Items.Clear();

while (axRecord.Found)

{

fieldName = axRecord.get\_Field("AccountNum");

fieldName1 = axRecord.get\_Field("EmailID");

ListBox2.Items.Add(fieldName.ToString() + "-" + fieldName1.ToString());

axRecord.Next();

}

ax1.Logoff();

ax1.Dispose();

}

Write below code in the click event of Button2.

protected void Button2\_Click(object sender, EventArgs e)

{

this.callAXClassMethod("kelly@test.com");

}

After running the application, you can see the output in the listbox2.

