Contents

[Introduction 1](#_Toc532662372)

[Create AOT Query 1](#_Toc532662373)

[AIF wizard 3](#_Toc532662374)

[Create job to generate XML in AX 6](#_Toc532662375)

[Update and Deploy Service 7](#_Toc532662376)

[Create .net application 11](#_Toc532662377)

# Introduction

Integrating AX with other application is the common requirement. This can be achieved using AIF in AX.

In this post, we have covered the functionality to export the customer’s basic information in XML format. This XML format can be generated either from AX or from .net application

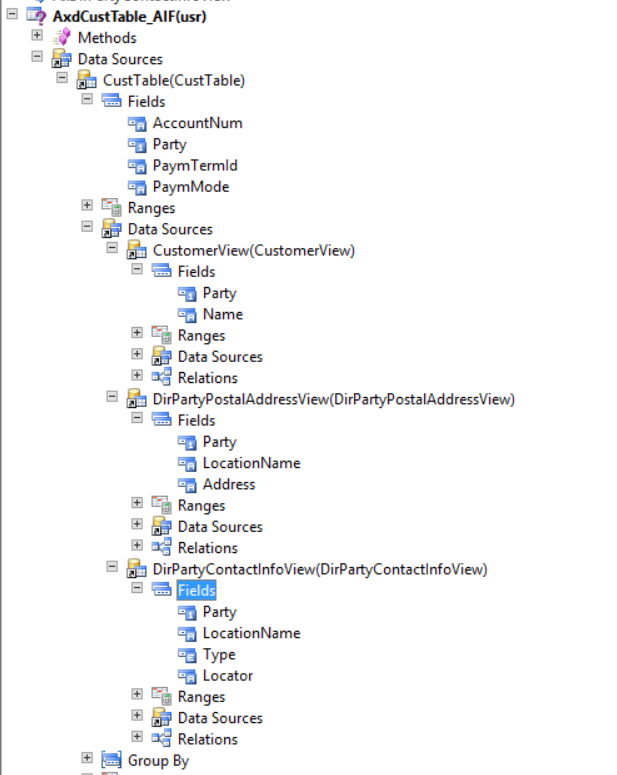
We will follow below steps to achieve the required functionality:

* Create AOT Query
* AIF wizard to generate project with AIF objects
* Create job to generate XML in AX
* Update and Deploy service for .net application
* Create .net application to generate the XML format of AX customers

# Create AOT Query

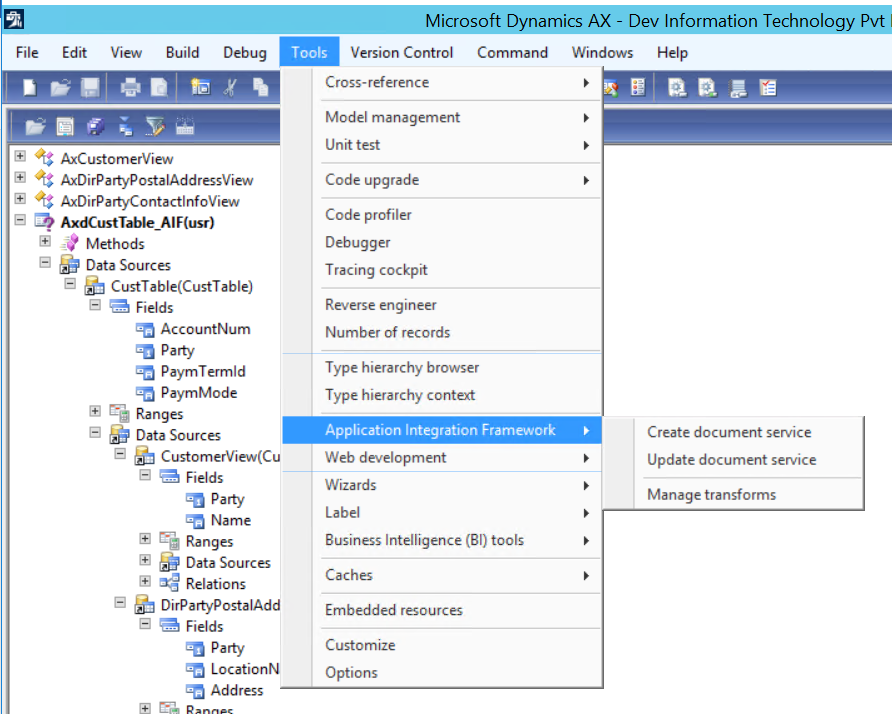
The customer’s basic information contains Account number, Name, Payment Term, Method of payment, Address and contact information.

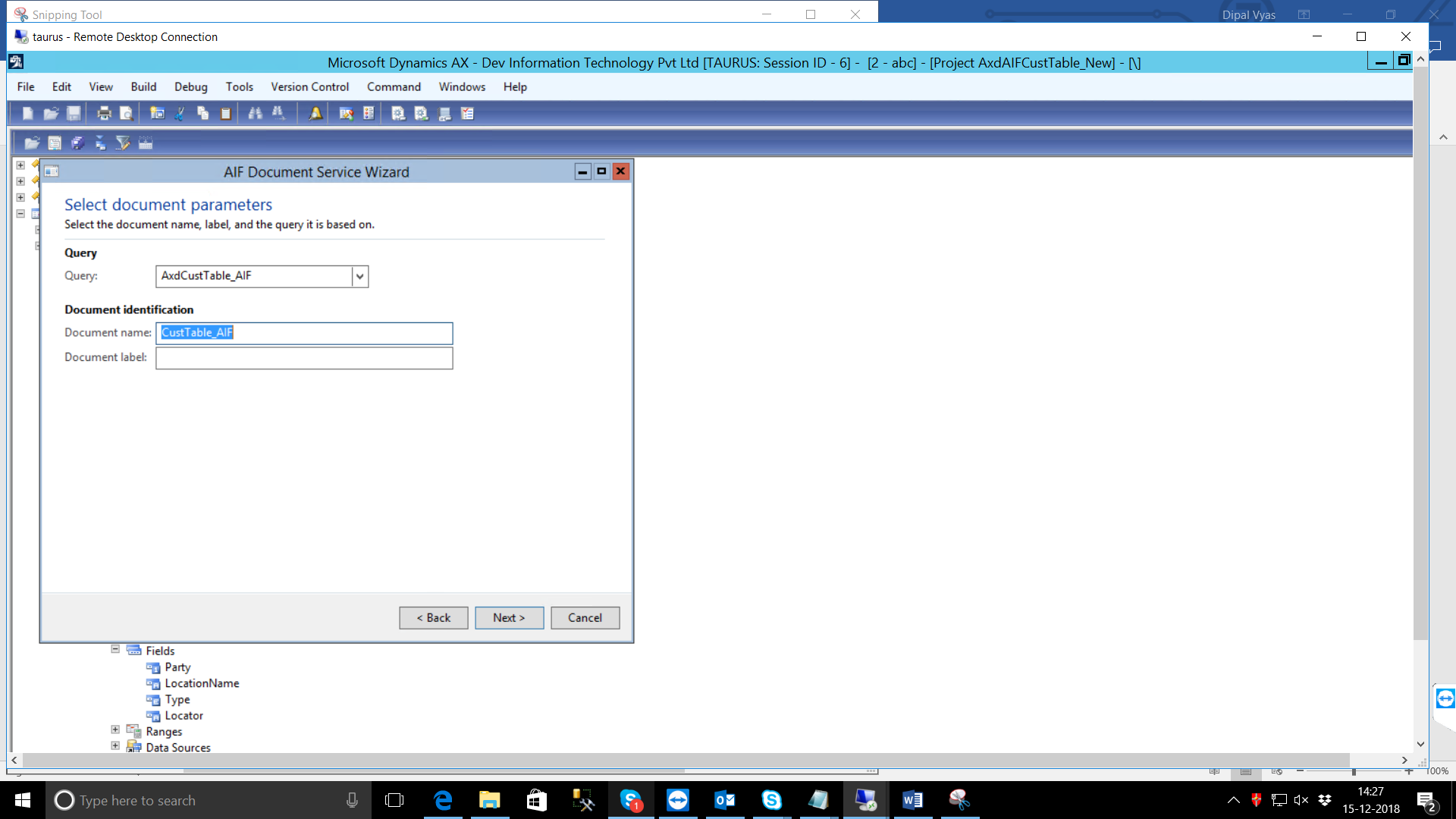
Create AOT query as below with all the required information. Add relation with field Party between all the tables/views.

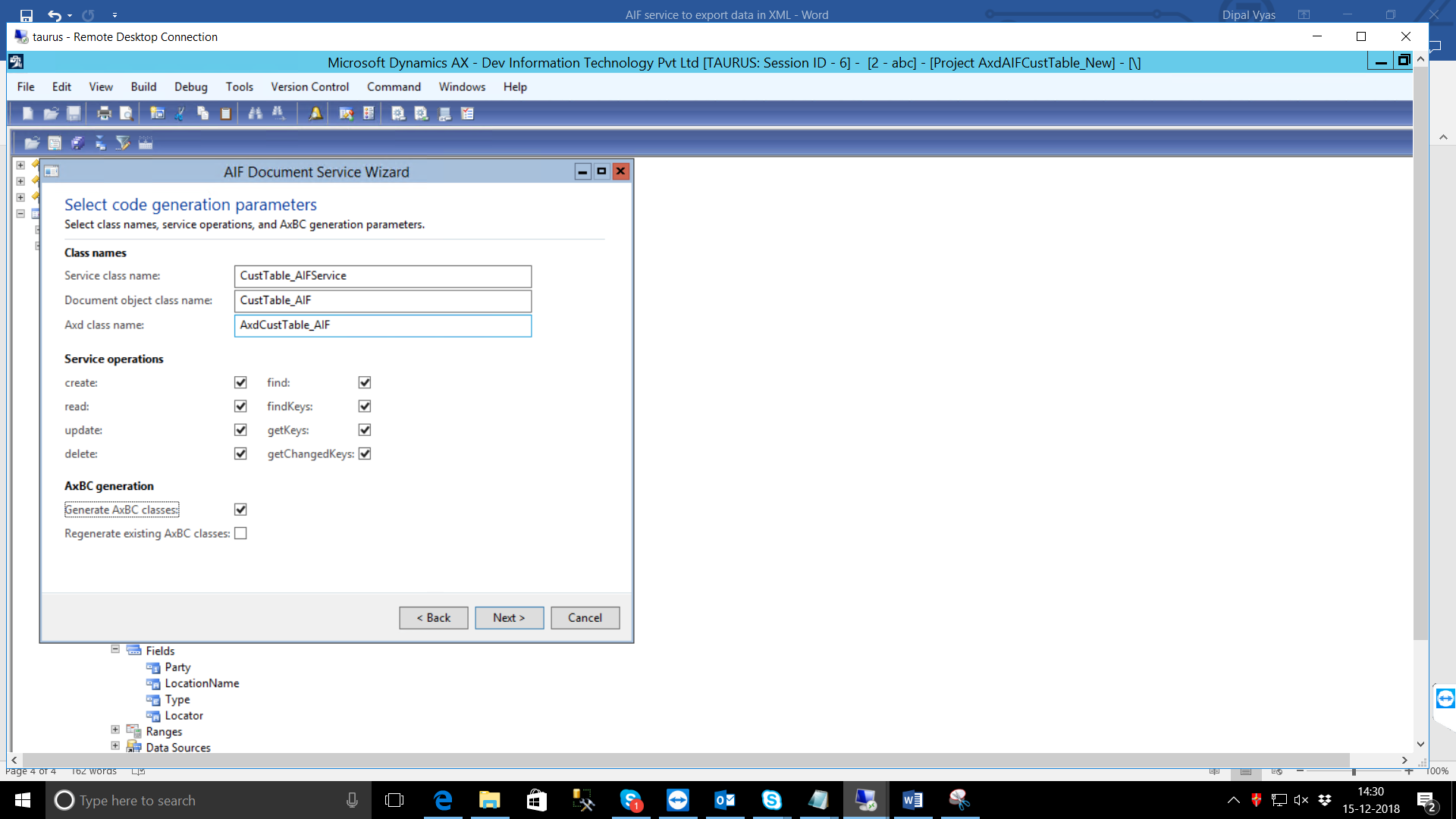
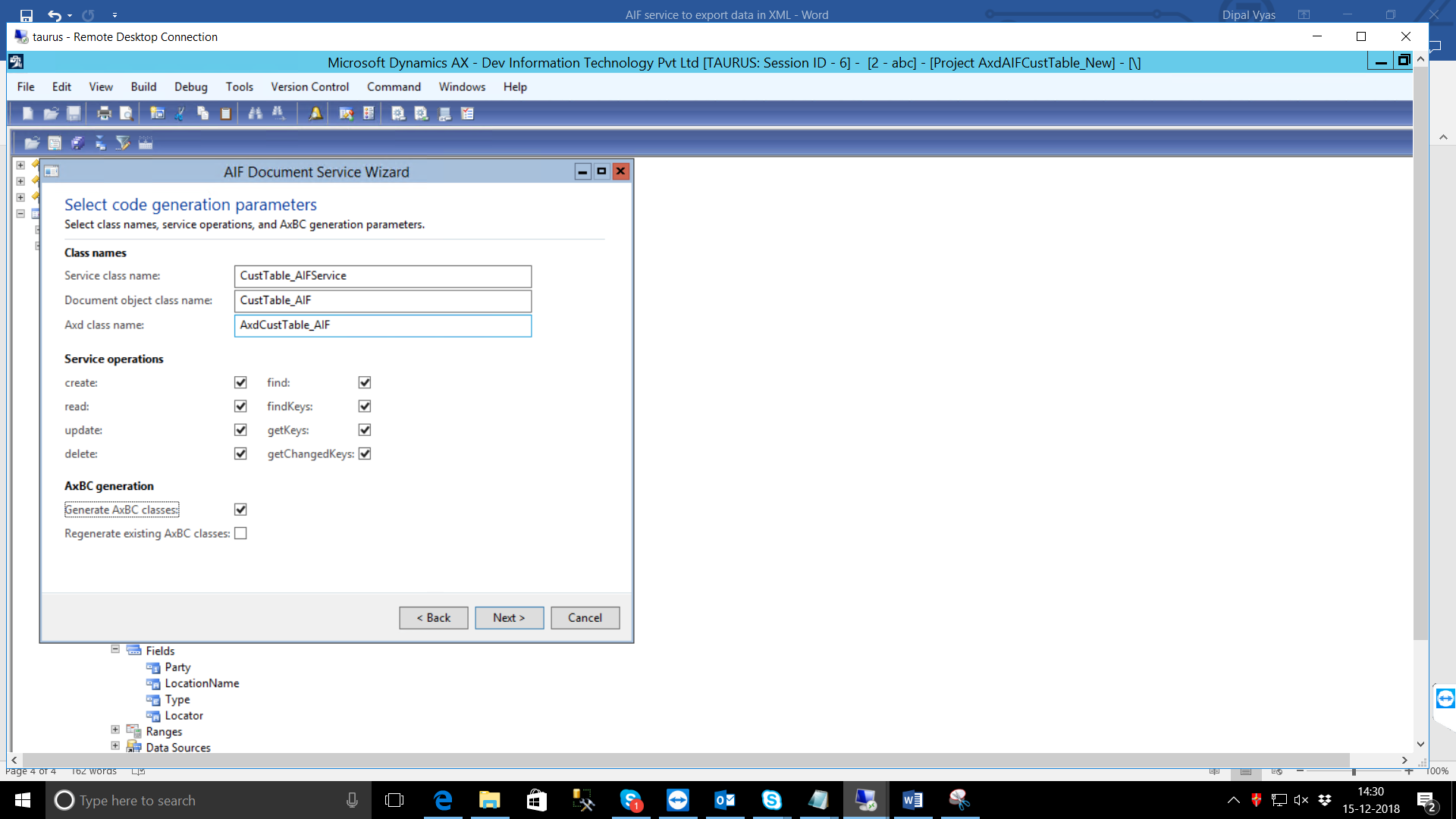


# AIF wizard

Create document service using AIF wizard from AOT >> Tools >> AIF >> Create document services

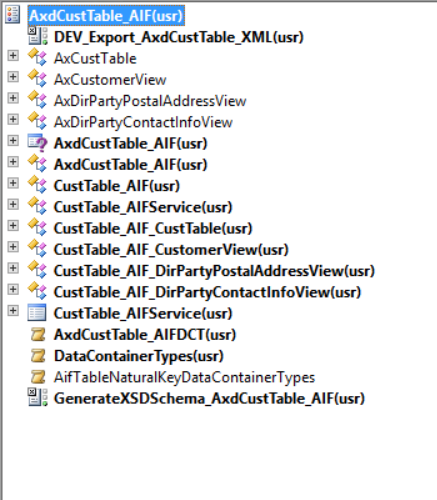






Click Next >> Finish.

AIF project with below object will be created.



# Create job to generate XML in AX

Below is the job to generate XML file of customer’s information.

**static** **void** DEV\_Export\_AxdCustTable\_XML(Args \_args)

{

CustTable custTable;

*//VendTable\_AIFService VendTable\_AIFService = AxdBase::newClassId(classnum(VendTable\_AIFService));*

AxdCustTable\_AIF AxdCustTable\_AIF = **new** AxdCustTable\_AIF(); *// Axd Class*

AifEntityKeyList aifEntiryKeyList = **new** AifEntityKeyList();

AifDocumentXml aifDocumentXml;

XmlDocument xmlDocument;

*// convert a record to the entity key*

AifEntityKey getEntityKey(Common \_record)

{

AifEntityKey aifEntityKey = AifEntityKey::construct();

List keyFields = SysDictTable::getUniqueIndexFields(\_record.TableId);

Map keyMap = SysDictTable::mapFieldIds2Values(keyFields,\_record);

;

aifEntityKey.parmKeyDataMap(keyMap);

aifEntityKey.parmTableId(\_record.TableId);

aifEntityKey.parmRecId(\_record.RecId);

**return** aifEntityKey;

}

;

*// add all records to be exported into the entity key list*

**while** **select** custTable

{

aifEntiryKeyList.addEntityKey(getEntityKey(custTable));

}

*// create an xml document containg the records*

aifDocumentXml = AxdCustTable\_AIF.readList(aifEntiryKeyList,

**null**,

**new** AifEndpointActionPolicyInfo(),

**new** AifConstraintListCollection(),

**connull**());

xmlDocument = XMLDocument::newXML(aifDocumentXML);

*// save the xml document on the disk*

xmlDocument.save("c:\\AIF\_XML\\AxdCustTable\_AIF.xml");

}

# Update and Deploy Service

Below are the steps to be followed to have customer information in SML format in .net application:

* Create method return XML format in string
* Add operation in the service
* Create and deploy service group

Create method as below in the service class “CustTable\_AIFService”

[AifCollectionTypeAttribute('return', Types::String), SysEntryPointAttribute(**true**)]

**public** **str** Get\_CustInfor\_XML()

{

**str** strXML;

CustTable custTable;

*//VendTable\_AIFService VendTable\_AIFService = AxdBase::newClassId(classnum(VendTable\_AIFService));*

AxdCustTable\_AIF AxdCustTable\_AIF = **new** AxdCustTable\_AIF(); *// Axd Class*

AifEntityKeyList aifEntiryKeyList = **new** AifEntityKeyList();

AifDocumentXml aifDocumentXml;

XmlDocument xmlDocument;

*// convert a record to the entity key*

AifEntityKey getEntityKey(Common \_record)

{

AifEntityKey aifEntityKey = AifEntityKey::construct();

List keyFields = SysDictTable::getUniqueIndexFields(\_record.TableId);

Map keyMap = SysDictTable::mapFieldIds2Values(keyFields,\_record);

;

aifEntityKey.parmKeyDataMap(keyMap);

aifEntityKey.parmTableId(\_record.TableId);

aifEntityKey.parmRecId(\_record.RecId);

**return** aifEntityKey;

}

;

*// add all records to be exported into the entity key list*

**while** **select** custTable

{

aifEntiryKeyList.addEntityKey(getEntityKey(custTable));

}

strXML = AxdCustTable\_AIF.readList(aifEntiryKeyList, **null**, **new** AifEndpointActionPolicyInfo(), **new** AifConstraintListCollection(), **connull**());

**return** strXML;

*// create an xml document containg the records*

*/\*aifDocumentXml = AxdCustTable\_AIF.readList(aifEntiryKeyList,*

*null,*

*new AifEndpointActionPolicyInfo(),*

*new AifConstraintListCollection(),*

*connull());*

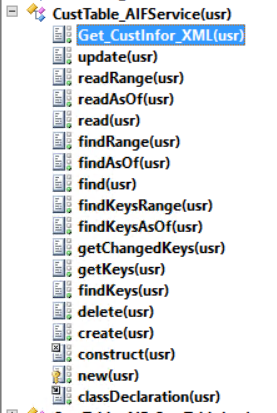
*xmlDocument = XMLDocument::newXML(aifDocumentXML);*

*// save the xml document on the disk*

*xmlDocument.save("c:\\AIF\_XML\\AxdCustTable\_AIF.xml");\*/*

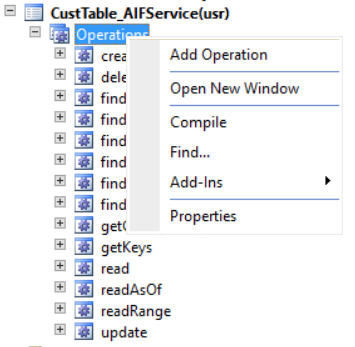
}

After adding above method in service class, the class will be viewed like below:

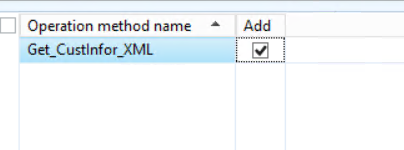


Add operations in service to have the access of this method in .net application.

As shown in the screen below, select service >> operations >> Add operation



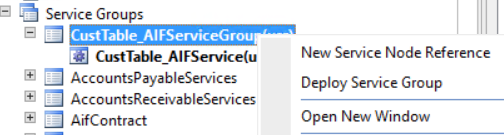
Select operation and click ok



Create servicegroup “CustTable\_AIFServiceGroup” , set autodeploy property to “Yes” and add service node in it.

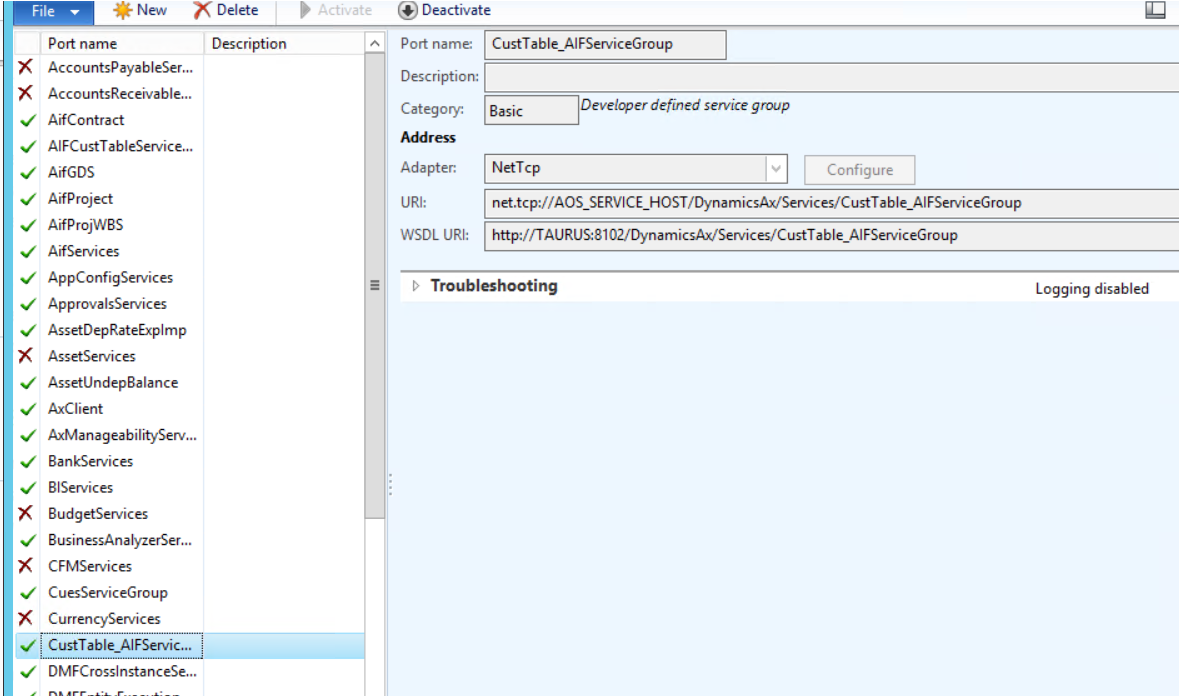


Right click on service group and deploy the service group



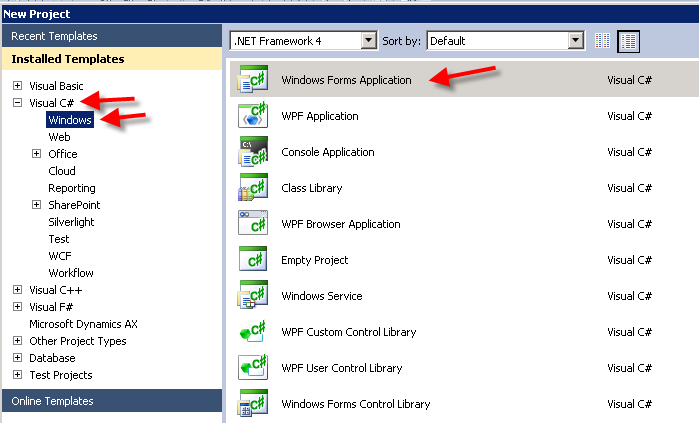
The deployed service group can be viewed from System administration >> Setup >> Services and application integration framework >> Inbound port

It contains the WSDL link which can be consumed from .net application



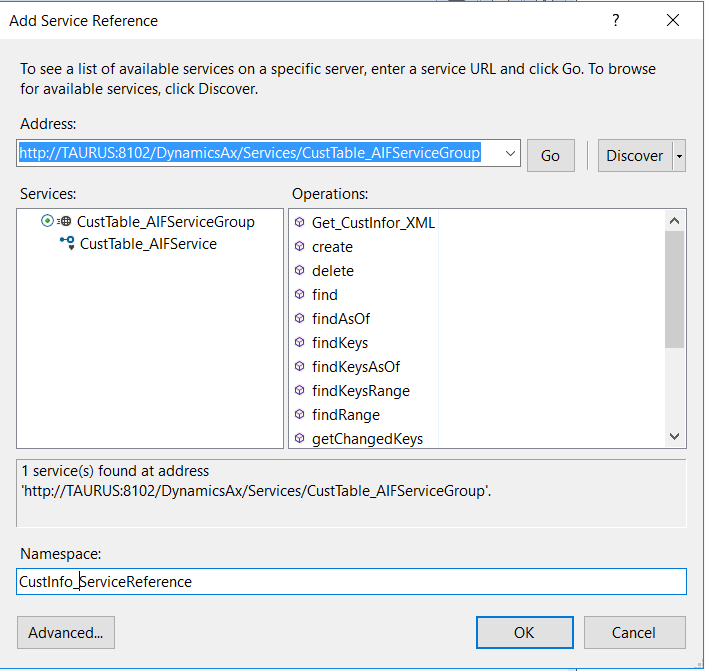
# Create .net application

Create new project of Type “Windows Application” for language Visual C#



In solution explorer >> Service References >> Right click and add service reference.

Copy the WSDL link from AX inbound port and add as reference as shown in the screen below:



All the operations of the service will be shown as in the screen above

Add button and textbox in the form and add below code on click event of the button.

private void button1\_Click(object sender, EventArgs e)

{

CustInfo\_ServiceReference.CustTable\_AIFServiceClient servClient = new CustInfo\_ServiceReference.CustTable\_AIFServiceClient();

string custinfoXML = servClient.Get\_CustInfor\_XML(new CustInfo\_ServiceReference.CallContext());

textBox1.Text = custinfoXML;

}

Now run the application and click on the button to view the customer information in textbox with XML format. It will show you the data of your default AX company in XML format.