JDE Business Services Workshop

Creating a Table I/O based Business Service

Lab Guide

Created by:

Enterprise Services Group

Solution Engineering

Change Record

Date	Editor	Version	Change Reference
07/24/2008	Ghazi Mehmood	1.0	Initial Version

COPYRIGHT & TRADEMARKS

Copyright © 2003, 2007, Oracle. All rights reserved. Powered by OnDemand Software. Distributed by Oracle under license from Global Knowledge Software LLC. © 1998-2006. All rights reserved.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:
U.S. GOVERNMENT RIGHTS The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.
Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

1. Contents

1. Contents 3
2. Introduction 4
3. Prerequisites 4
4. Add a New OMW Project 4
5. Add a Published Business Service Object 4
6. Add a Business Service Object 6
7. Launch JDeveloper from OMW 7
8. Add JDeveloper Project for the Business Service 8
9. Add Value Object for Business Service (J55911Z1) 10
10. Add Input Value Object for Published Business Service (JP550009) 13
11. Add Output Value Object for Published Business Service JP550009 18
12. Add Business Service Class (J55911Z1) 22
13. Add Published Business Service Class (JP550010) 28
14. Creating a Test Class 31
15. Running the Test Class 33
16. Review and Next Steps 36

2. Introduction

JD Edwards EnterpriseOne provides interoperability with other Oracle applications and thirdparty systems by natively producing and consuming web services. Web services enable software applications written in various programming languages and running on various platforms to exchange information. JD Edwards EnterpriseOne exposes business services as web services . A web service is a standardized way of integrating web-based applications, and in JD Edwards EnterpriseOne, web services are referred to as published business services. Business services enable JD Edwards EnterpriseOne to expose transactions as a basic service that can expose an XML document-based interface.

In this lab, we will learn how to create a Table I/O Business Service.

3. Prerequisites

1. JDE Development Experience
2. Knowledge of Java.

4. Add a New OMW Project

In this section of the lab we will create a new OMW Project.

1. In OMW, click on Find to bring up all the current OMW projects
2. Click on the **Add** button. 

3. Select **OMW Project** and click **OK**.

4. In the Project Revisons screen, enter the following information and click on the **OK** button:
   Project ID: **CUSTOM01**
   Description: **BSSV Tutorial 1**
   Type: **02**
   Severity: **01**
   System Code: **55**
   Release: **E812**
5. The project is now added. The default project status is 21 (Programming) and User Role is Developer.

5. Add a Published Business Service Object

In this section of the lab we will add a Published Business Service object.

1. Select the OMW project that we created in the previous step and click **Add**.
2. Select the **Business Function** radio button and click **OK**.

3. On the *Add Object* form, complete the following fields and click the **OK** button:

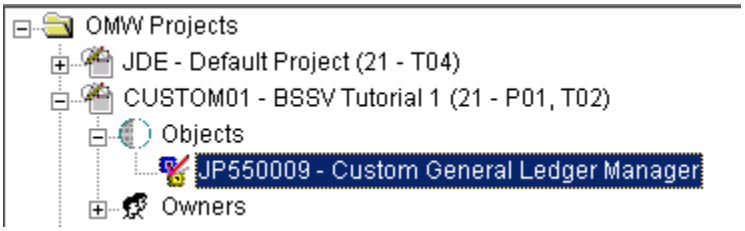
Object Name: **JP550010**
Description: **Custom General Ledger Manager**
Product Code: **55**
System Code: **55**
Object Use: **330**

Source Language: **BSSV**
Package Prefix: **oracle.e1.bssv** (**Note:** Use the flash light to select this value)

NOTE: ***The Package Prefix field is enabled when you select BSSV as the Source Language***

4. On the Business Function design form click **OK** to return to OMW.

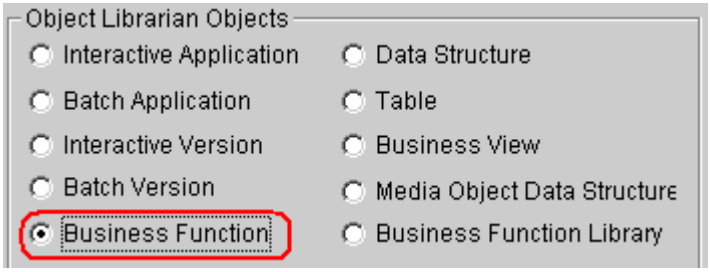
5. You should see the Published Business Service under the **Objects** node of your project.



6. Add a Business Service Object

In this section of the lab we will add a Business Service object in the same manner as the Published Business Service we created in the previous section.

1. Select the OMW project that we created in the previous step and click **Add**.
2. Select the **Business Function** radio button and click **OK**.



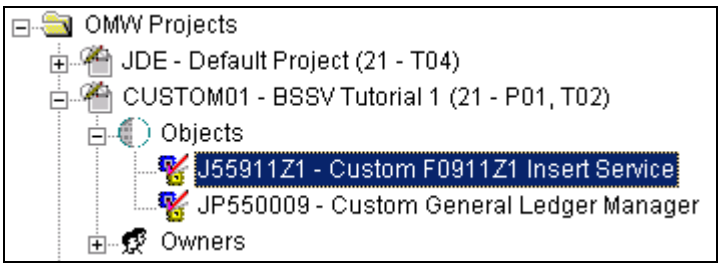
3. On the Add Object form, complete the following fields and click the **OK** button:

Object Name: **J55911Z1**
Description: **Custom F0911Z1 Insert Service**
Product Code: **55**
System Code: **55**
Object Use: **Undefined**

Source Language: **BSSV**
Package Prefix: **oracle.e1.bssv** (**Note:** Use the flash light to select this value)

NOTE: ***The Package Prefix field is enabled when you select BSSV as the Source Language***

4. On the Business Function design form click **OK** to return to OMW.
5. You should see the Business Service under the **Objects** node of your project.

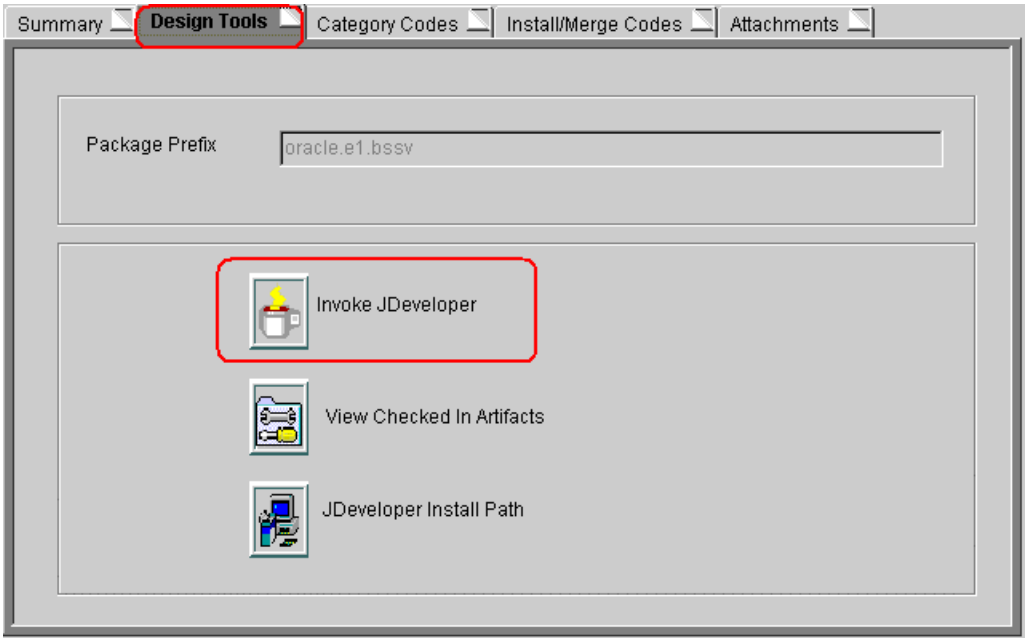


7. Launch JDeveloper from OMW

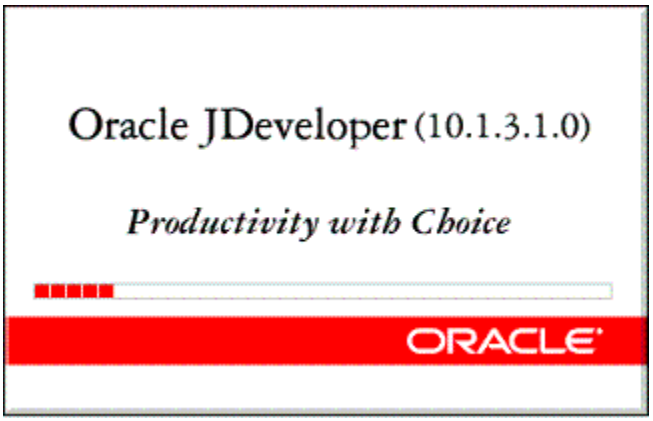
In this step we will launch the Oracle JDeveloper IDE from within Object Management Workbench (OMW).

1. In OMW, Select the Published Business Service (JP550009) and click on the **Design** button in the center column.

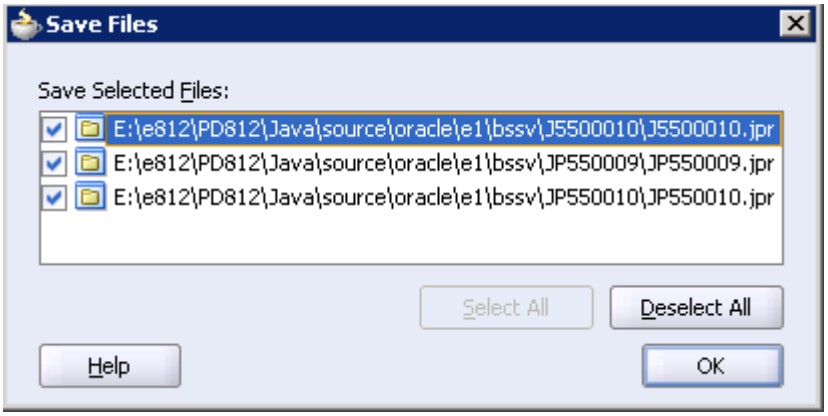
2. Click on the **Design Tools** tab and then click on **Invoke JDeveloper**.



3. This will launch Oracle JDeveloper



4. If you are prompted to save files, select **OK**.



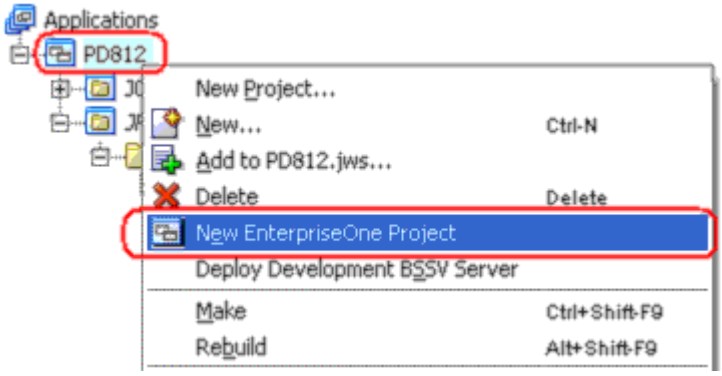
Note: Your screen may not be the same as the screenshot above.

5. When JDeveloper opens, you will see a **JDeveloper project** for the Published Business Service (**JP550009**).

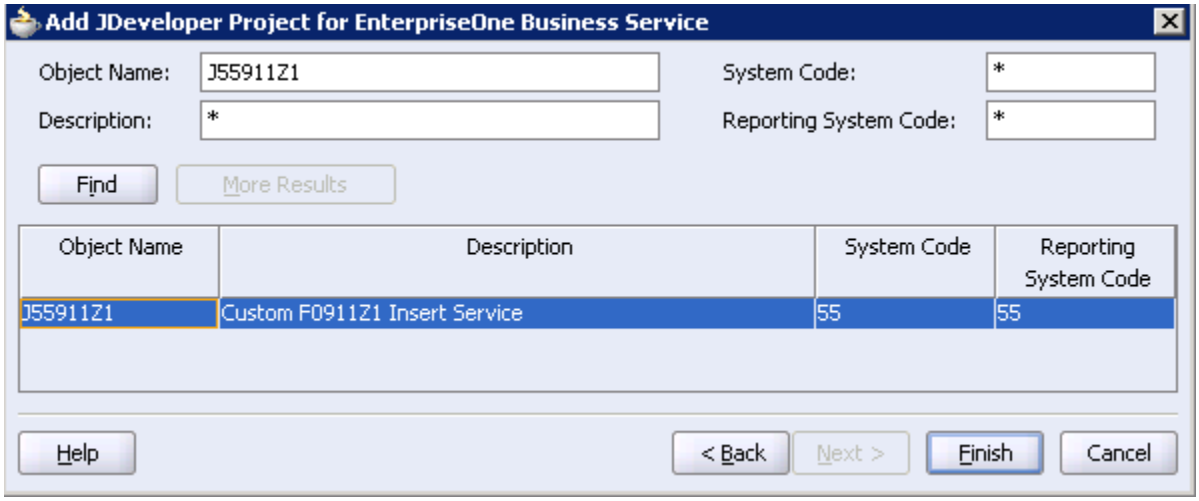
8. Add JDeveloper Project for the Business Service

In the previous section a JDeveloper project was automatically created when we launched JDeveloper from OMW. In this section we will manually add a JDeveloper project for the Business Service object (J55911Z1).

1. Select the application **PD812** and do a right click. Select **New EnterpriseOne Project** from the menu.

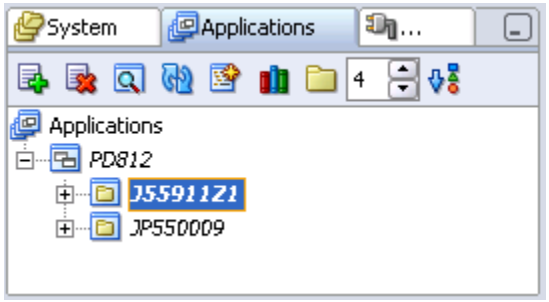


2. In the *Add JDeveloper Project for EnterpriseOne Business Service* window, search for the Business Service by entering the object name **J55911Z1** (You can search by entering J55*). Next, click on the **Find** button.



3. Select the Business Service and then click on the **Finish** button.

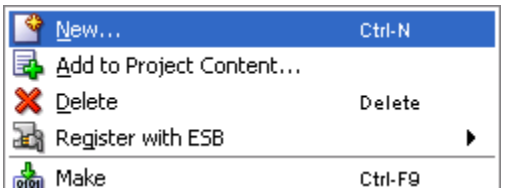
4. The J55911Z1 Business Service project will now be added to your JDeveloper Application.



9. Add Value Object for Business Service (J55911Z1)

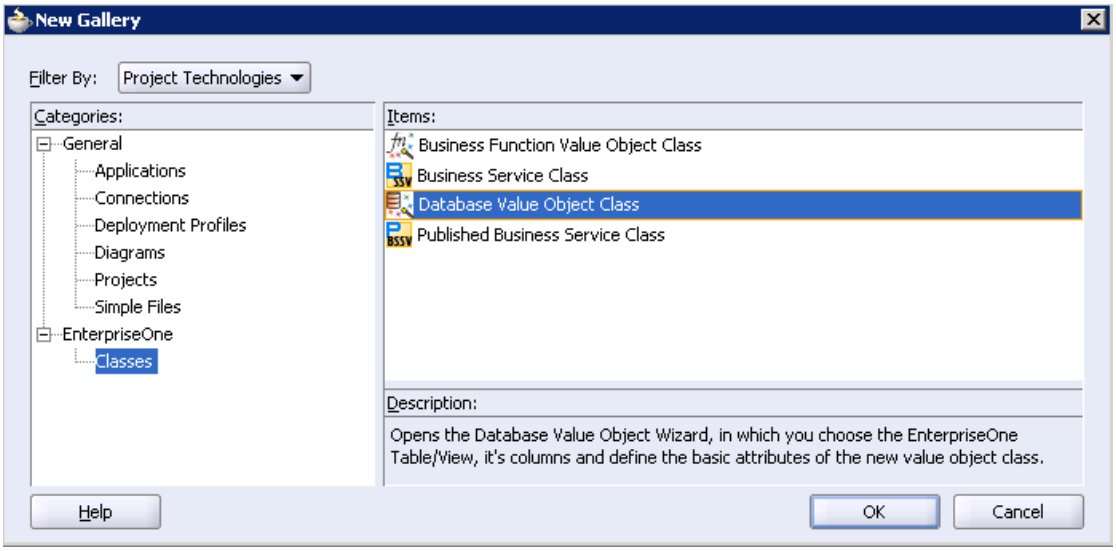
In this section of the lab we will add a Value Object for our Business Service. For a Businiess Service there is only one value object that is used for both input and output (similar to a business function data structure).

1. Select the Business Service (**J55911Z1**) and right click. Select **New** from the menu.

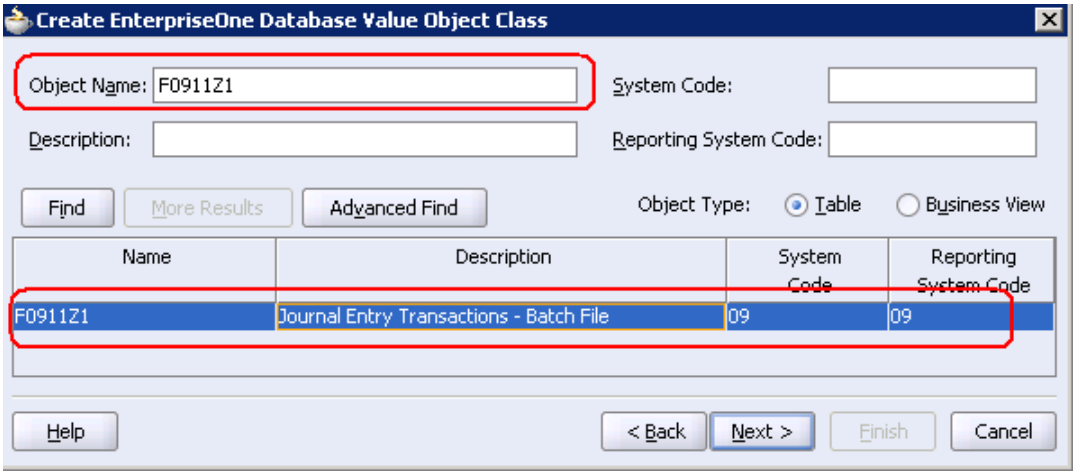


2. Under Categories, select **EnterpriseOne->Classes**.

   Select **Database Value Object Class** and press the **OK** button.

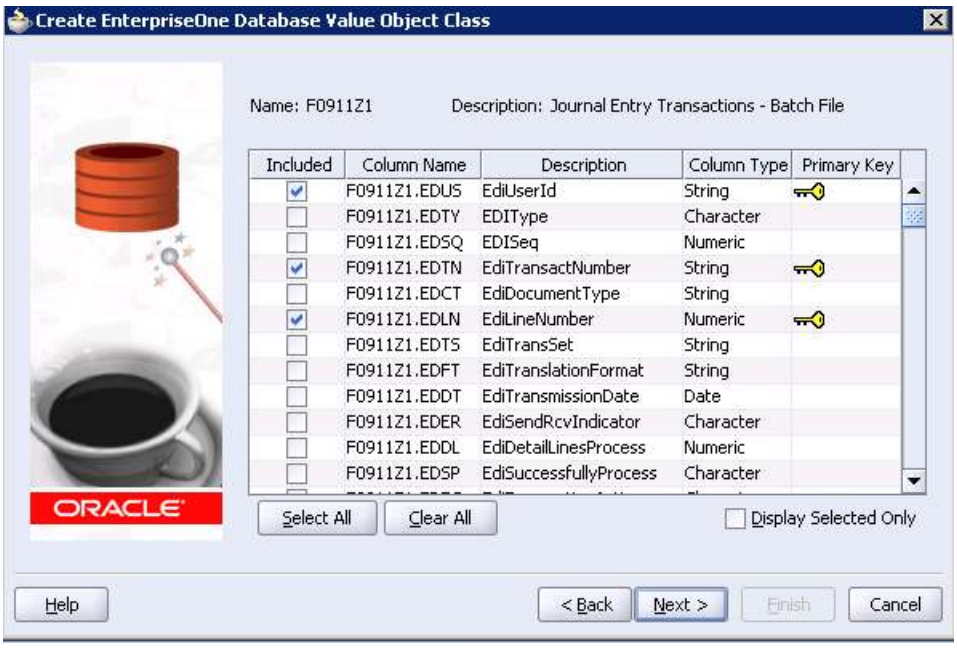


3. In the *Create EnterpriseOne Database Value Object* window, enter **F0911Z1** as the Object Name and press the **Find** button. Select the table from the result and press the **Next** button.



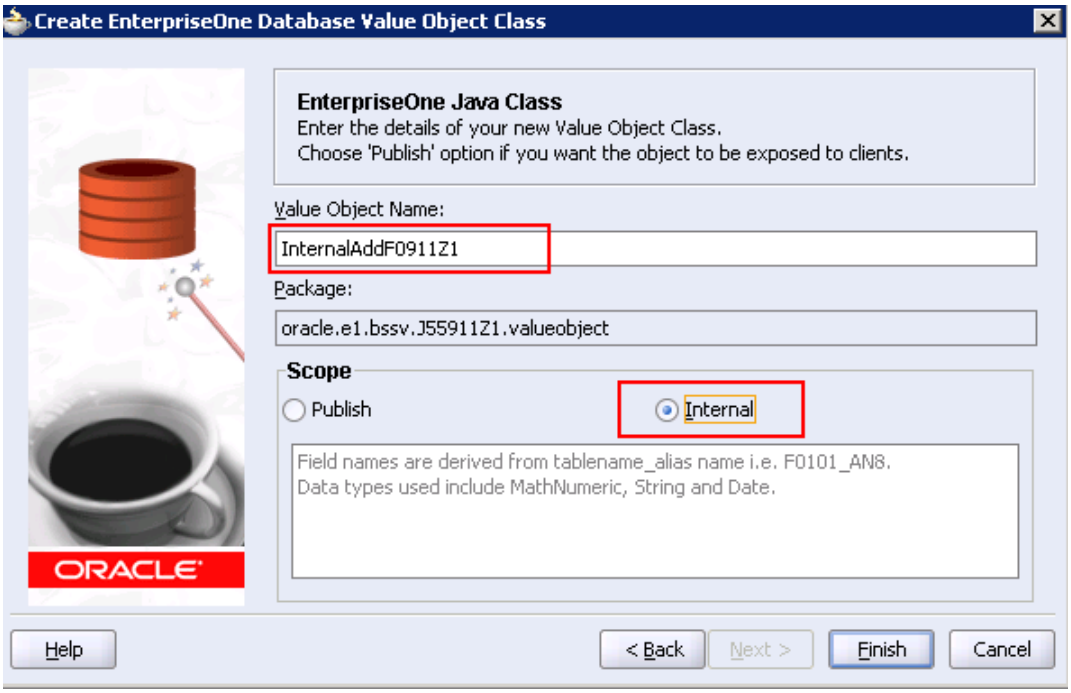
4. Select the following columns from the table and press the **Next** button.

NAME	ALIAS
EDI - User ID	EDUS
EDI - Transaction Number	EDTN
EDI - Line Number	EDLN
EDI - Batch Number	EDBT
Date - For G/L (and Voucher) - Julian	DGJ
Company	CO
Account Number - Input (Mode Unknown)	ANI
Subledger - G/L	SBL
Subledger Type	SBLT
Ledger Types	LT
Currency Code - From	CRCD
Currency Conversion Rate - Spot Rate	CRR
Amount	AA
Units	U
Unit of Measure	UM
Name - Alpha Explanation	EXA
Name - Remark Explanation	EXR
Reference 2	R2
Bill Code	BC
Date - Service/Tax	DSVJ



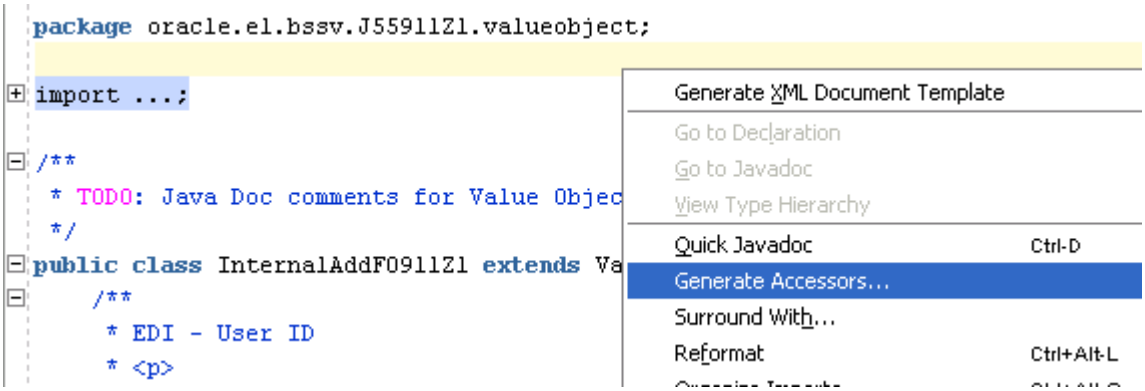
5. Specify the Value Object name and Scope

   Value Object Name: **InternalAddF0911Z1**
   Scope: **Internal**

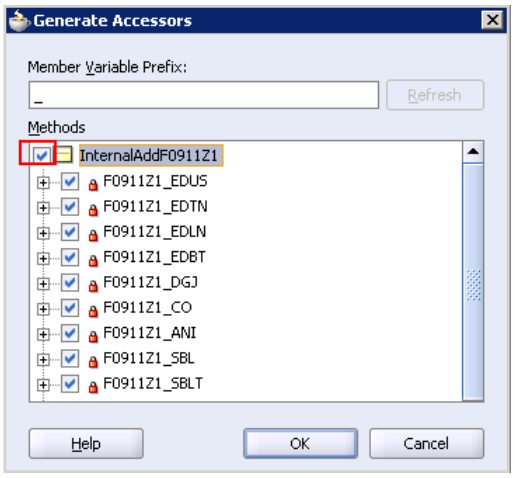


6. Press the **Finish** button. The Value Object class has been created and is displayed in JDeveloper.

7. Right-click on any whitespace in the value object class and click on **Generate Accessors**.



8. In the Generate Accessors window, select all fields and press the **OK** button.



9. The **get** and **set** methods for the value object elements have been created. You can scroll down to view these methods.

10. **Save** the value object class file.

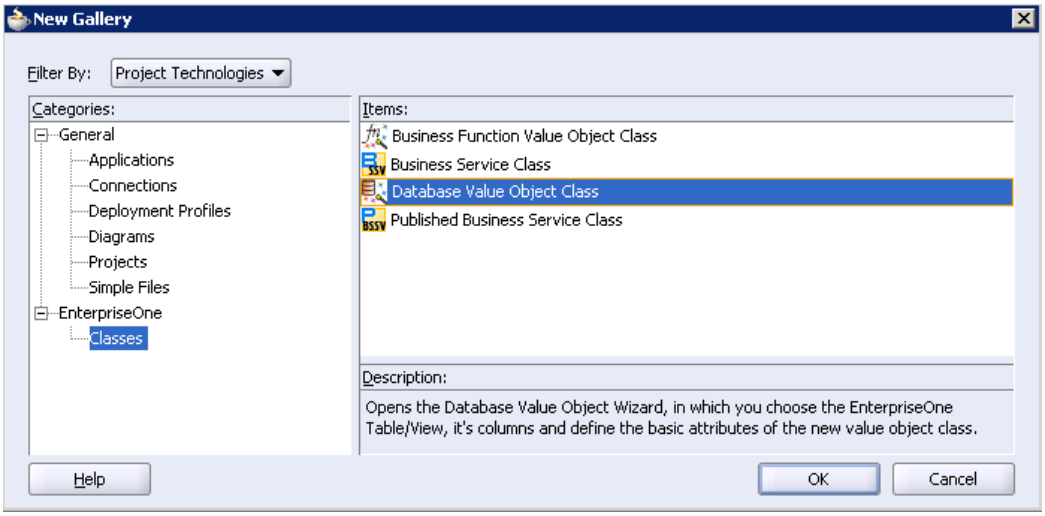
10. Add Input Value Object for Published Business Service (JP550009)

In the previous section we created one input/output Value Object for the Business Service JP550009. For a Published Business Service, the input and output value objects are separate. In this section we will create the Input Value Object.

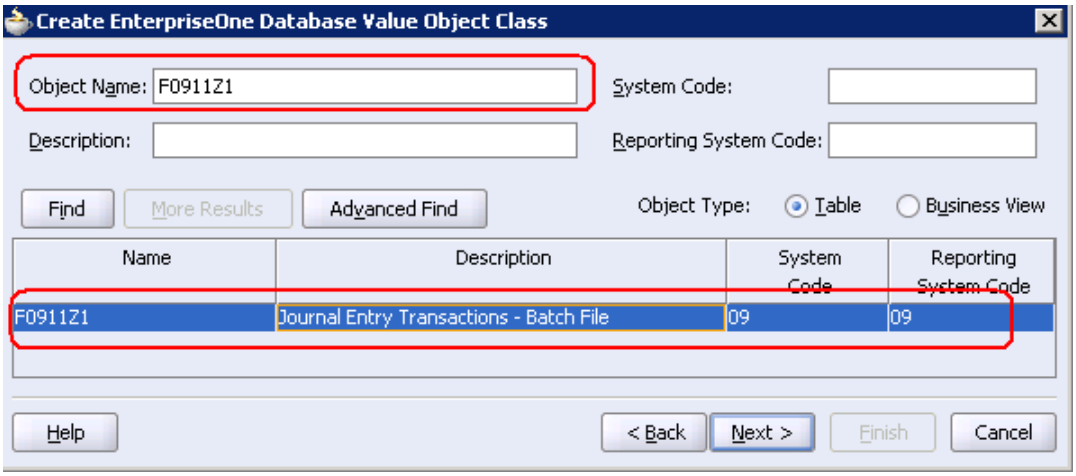
1. Select the Business Service (**JP550009**) and right click. Select **New** from the menu.

2. Under Categories, select **EnterpriseOne->Classes**.

   Select **Database Value Object Class** and press the **OK** button.

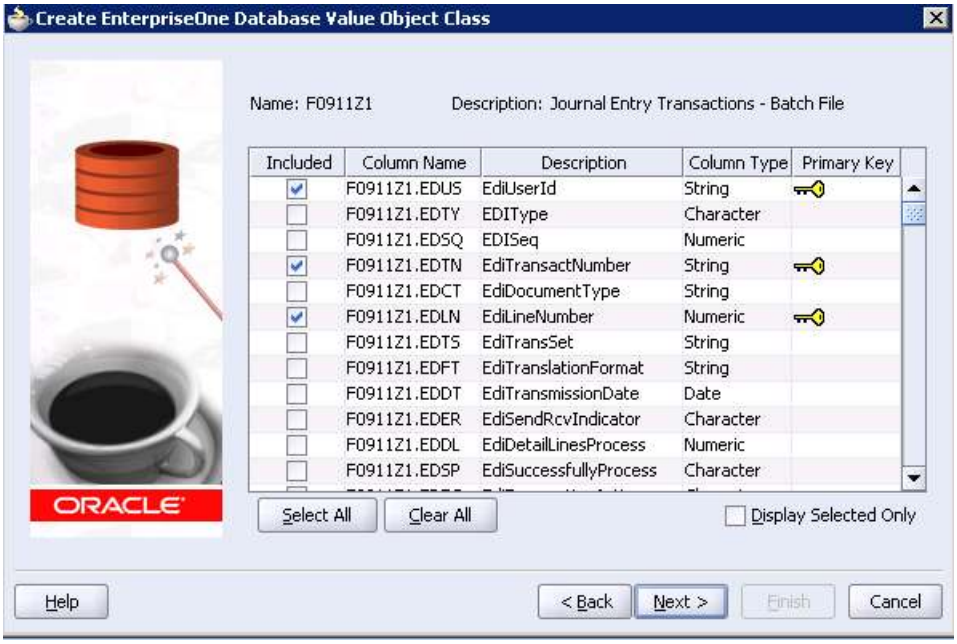


3. In the *Create EnterpriseOne Database Value Object* window, enter **F0911Z1** as the Object Name and press the **Find** button. Select the table from the result and press the **Next** button.

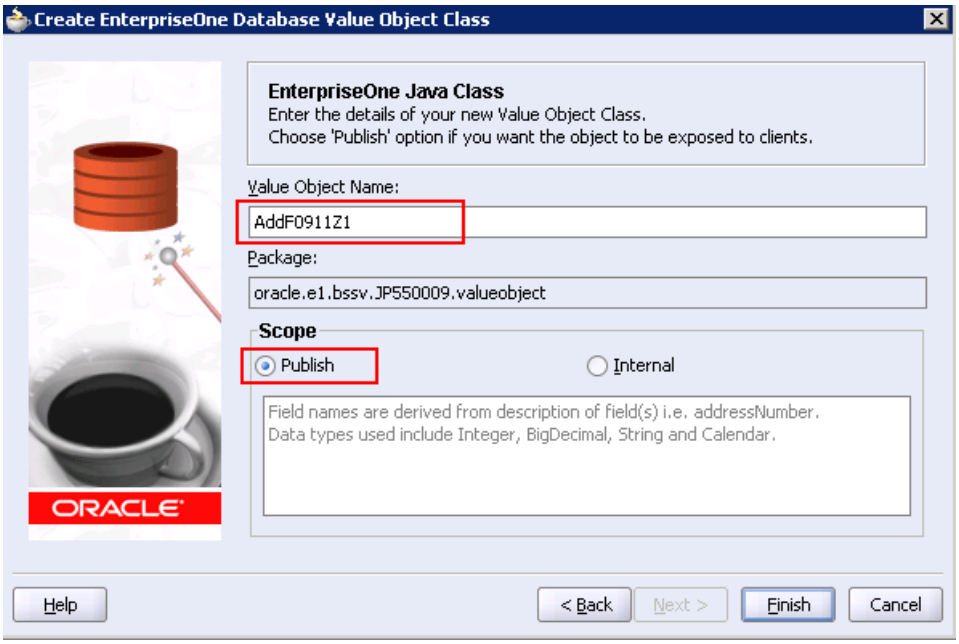


4. Select the following fields

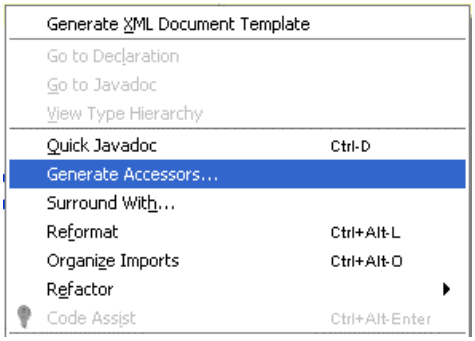
NAME	ALIAS
EDI - User ID	EDUS
EDI - Transaction Number	EDTN
EDI - Line Number	EDLN
EDI - Batch Number	EDBT
Date - For G/L (and Voucher) - Julian	DGJ
Company	CO
Account Number - Input (Mode Unknown)	ANI
Subledger - G/L	SBL
Subledger Type	SBLT
Ledger Types	LT
Currency Code - From	CRCD
Currency Conversion Rate - Spot Rate	CRR
Amount	AA
Units	U
Unit of Measure	UM
Name - Alpha Explanation	EXA
Name - Remark Explanation	EXR
Reference 2	R2
Bill Code	BC
Date - Service/Tax	DSVJ



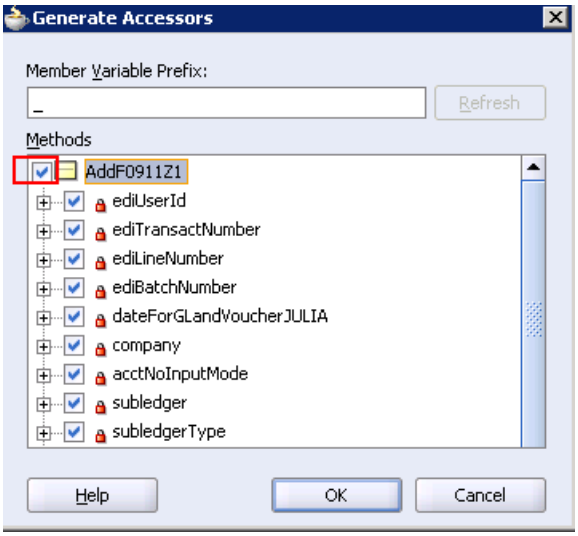
5. Set the *Value Object Name* to **AddF0911Z1** and the *Scope* as **Publish**. Click the **Finish** button to create the input Value Object.



6. Right-click on any whitespace in the value object class and click on **Generate Accessors**.



7. Select all the fields and press the **OK** button.



8. The **get** and **set** methods for the value object elements have been created. You can scroll down to view these methods.

9. Add the Import statements highlighted in ***bold*** below:

```
package oracle.e1.bssv.JP550009.valueobject;
import java.io.Serializable;
import java.math.BigDecimal;
import java.util.Calendar;

import java.util.Date;
import oracle.e1.bssv.J55911Z1.valueobject.InternalAddF0911Z1;
import oracle.e1.bssvfoundation.base.IContext;
import oracle.e1.bssvfoundation.util.E1MessageList;
import oracle.e1.bssvfoundation.util.MathNumeric;
import oracle.e1.bssvfoundation.base.ValueObject;
```

10. Add the following function (highlighted in ***bold***). This goes towards the end of your code.

```
    public Calendar getDateServiceCurrency() {
        return dateServiceCurrency;
    }

    public E1MessageList mapFromPublished(IContext context, InternalAddF0911Z1 InternalVO){
        E1MessageList messages = new E1MessageList();
        //set all internal VO attributes based on external VO passed in
        InternalVO.setF0911Z1_EDUS(this.getEdiUserId());
        InternalVO.setF0911Z1_EDTN(this.getEdiTransactNumber());
        InternalVO.setF0911Z1_ANI(this.getAcctNoInputMode());
        if (this.getAmountField() != null)
            InternalVO.setF0911Z1_AA(new MathNumeric(this.getAmountField()));
        InternalVO.setF0911Z1_CO(this.getCompany());
        InternalVO.setF0911Z1_EXA(this.getNameAlphaExplanation());
        InternalVO.setF0911Z1_CRCD(this.getCurrencyCodeFrom());
        if (this.getCurrencyConverRateOv() != null)
            InternalVO.setF0911Z1_CRR(new MathNumeric(this.getCurrencyConverRateOv()));
        InternalVO.setF0911Z1_EXR(this.getNameRemarkExplanation());
        InternalVO.setF0911Z1_SBLT(this.getSubledgerType());
        InternalVO.setF0911Z1_LT(this.getLedgerType());
        InternalVO.setF0911Z1_SBL(this.getSubledger());
        InternalVO.setF0911Z1_BC(this.getBillCode());
        InternalVO.setF0911Z1_EDBT(this.getEdiBatchNumber());
        if (this.getDateServiceCurrency() != null)
            InternalVO.setF0911Z1_DSVJ(this.getDateServiceCurrency().getTime());
        if (this.getUnits() != null)
            InternalVO.setF0911Z1_U(new MathNumeric(this.getUnits()));
        InternalVO.setF0911Z1_UM(this.getUnitOfMeasure());
        InternalVO.setF0911Z1_R2(this.getReference2());
        if (this.getDateForGLandVoucherJULIA() != null)
            InternalVO.setF0911Z1_DGJ(this.getDateForGLandVoucherJULIA().getTime());

        return messages;
    }

}
```

11. Save the File.

11. Add Output Value Object for Published Business Service JP550009

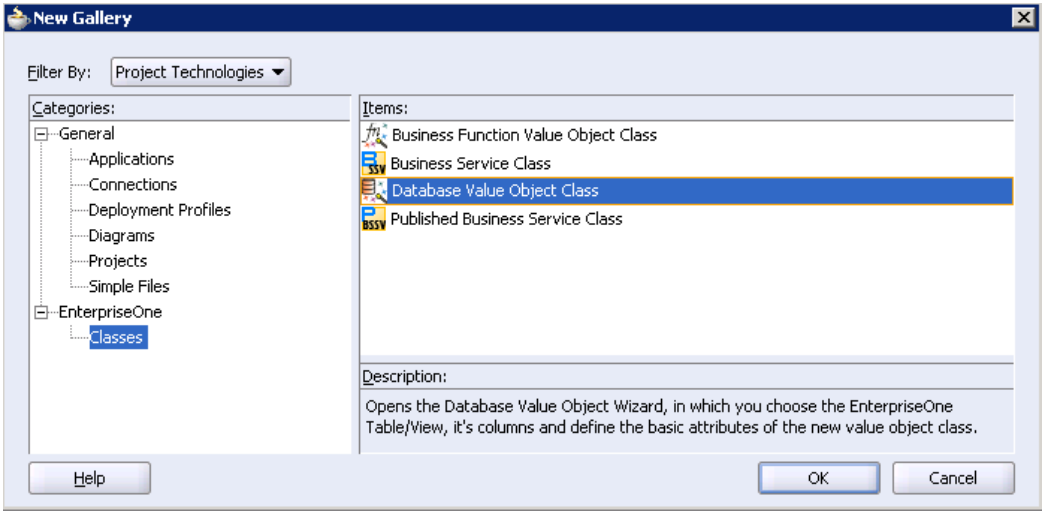
In the previous section we created an Input Value Object. Now we will create an Output Value Object.

NOTE: The process to create the output value object is the similar to the previous value objects. In the interest of time we will create a template for the value object and then copy/paste the code from the solutions directory.

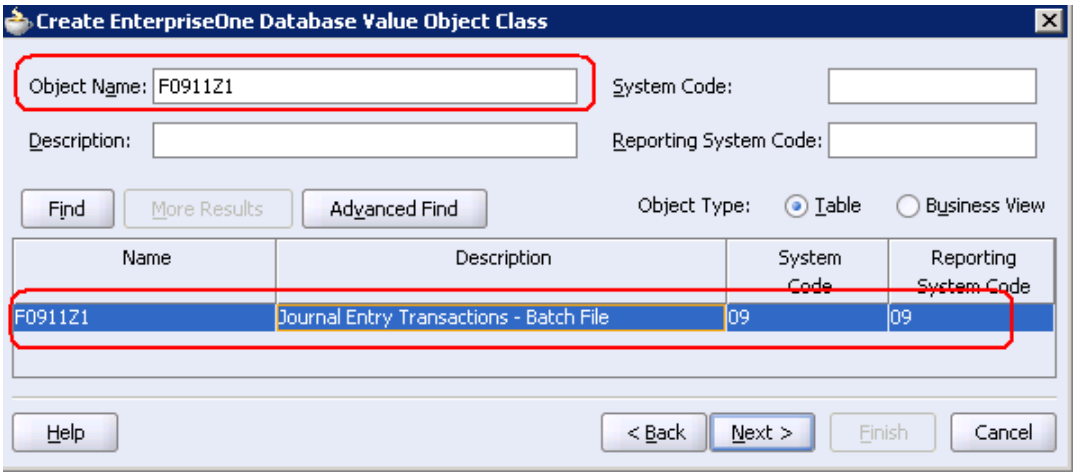
1. Select the Business Service (**JP550009**) and right click. Select **New** from the menu.

2. Under Categories, select **EnterpriseOne->Classes**.

   Select **Database Value Object Class** and press the **OK** button.

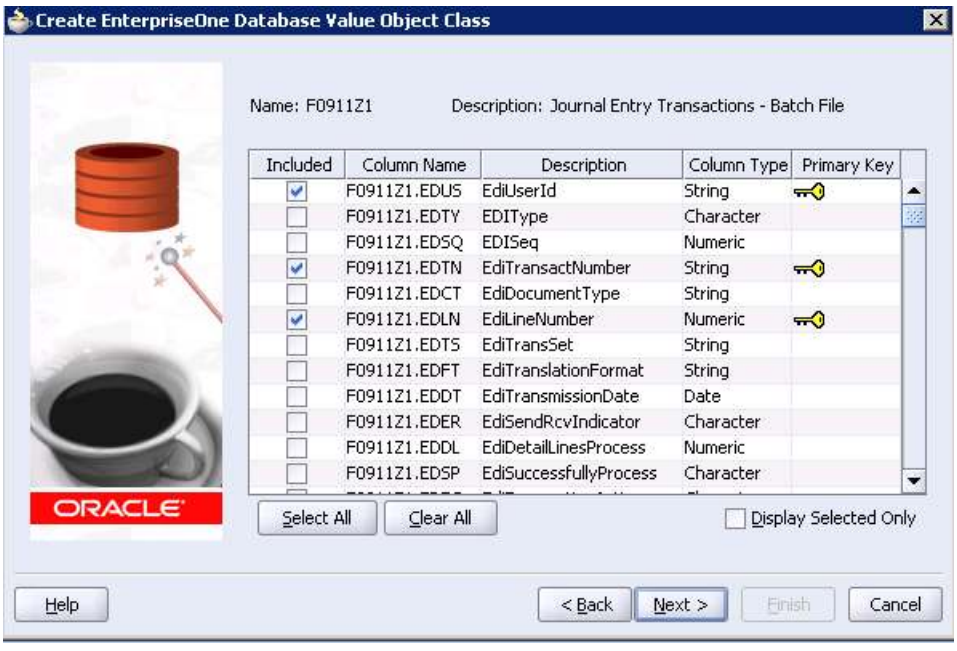


3. In the *Create EnterpriseOne Database Value Object* window, enter **F0911Z1** as the Object Name and press the **Find** button. Select the table from the result and press the **Next** button.

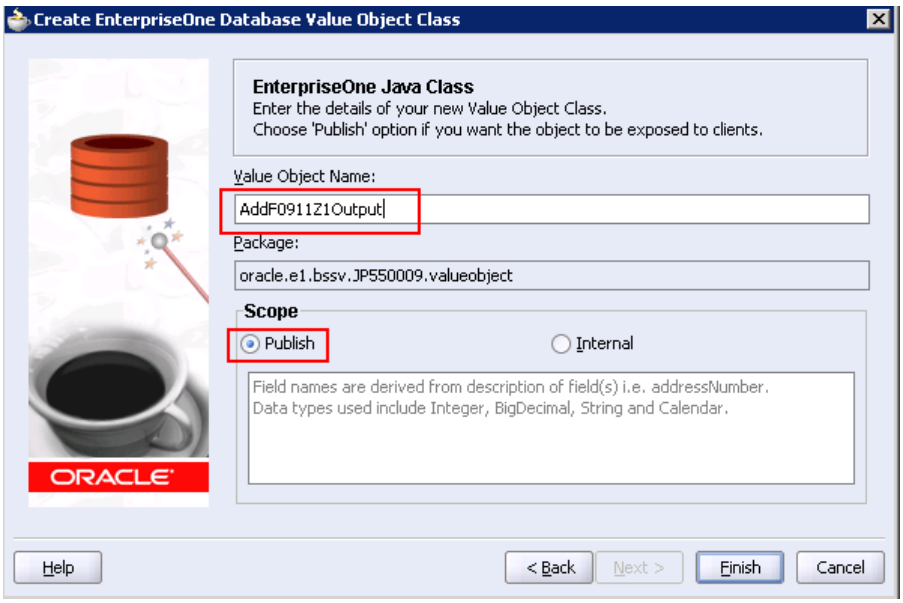


4. Select the following fields:

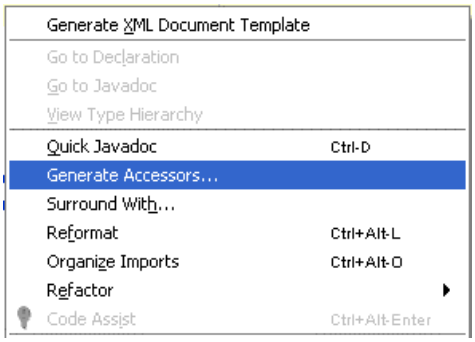
NAME	ALIAS
EDI - User ID	EDUS
EDI - Transaction Number	EDTN
EDI - Line Number	EDLN
EDI - Batch Number	EDBT
Date - For G/L (and Voucher) - Julian	DGJ
Company	CO
Account Number - Input (Mode Unknown)	ANI
Subledger - G/L	SBL
Subledger Type	SBLT
Ledger Types	LT
Currency Code - From	CRCD
Currency Conversion Rate - Spot Rate	CRR
Amount	AA
Units	U
Unit of Measure	UM
Name - Alpha Explanation	EXA
Name - Remark Explanation	EXR
Reference 2	R2
Bill Code	BC
Date - Service/Tax	DSVJ



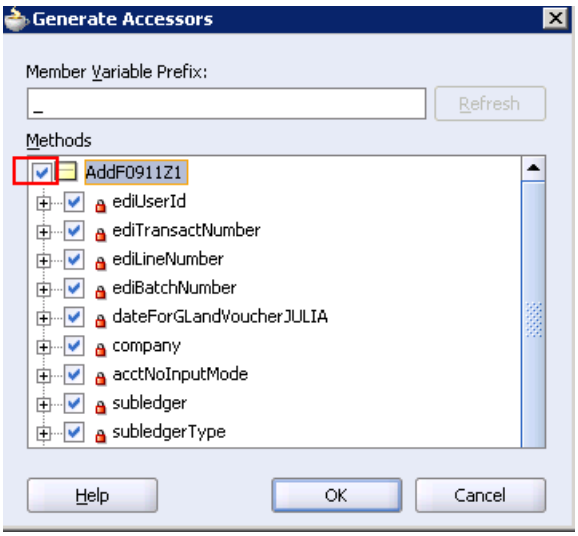
5. Set the *Value Object Name* to **AddF0911Z1Output** and the *Scope* as **Publish**. Click the **Finish** button to create the output Value Object.



6. Right-click on any whitespace in the value object class and click on **Generate Accessors**.



7. Select all the fields and press the **OK** button.



8. The **get** and **set** methods for the value object elements have been created. You can scroll down to view these methods.

9. Add the Import statements highlighted in ***bold*** below:

```
import java.io.Serializable;
import java.math.BigDecimal;
import java.util.Calendar;
import oracle.e1.bssv.J55911Z1.valueobject.InternalAddF0911Z1;
import oracle.e1.bssvfoundation.base.MessageValueObject;
import oracle.e1.bssvfoundation.util.E1MessageList;
import oracle.e1.bssvfoundation.base.ValueObject;

/**
 * TODO: Java Doc comments for Value Object here
 */
```

10. Make the following change:

public class AddF0911Z1Output extends *ValueObject* implements Serializable {
- to -
public class AddF0911Z1Output extends ***MessageValueObject*** implements Serializable {

11. Add the following function (highlighted in **bold**).

```
  /**
   * TODO: Default public constructor for instantiating: AddF0911Z1Output
   */
  public AddF0911Z1Output() {
  }

  public AddF0911Z1Output(InternalAddF0911Z1 internalVO) {
    /* In this function we map the internal VO return values to the Output VO of the Publishsed biz
service */
      this.setEdiUserId(internalVO.getF0911Z1_EDUS());
      this.setEdiTransactNumber(internalVO.getF0911Z1_EDTN());
      this.setAcctNoInputMode(internalVO.getF0911Z1_ANI());
      this.setAmountField(internalVO.getF0911Z1_AA().asBigDecimal());
      this.setCompany(internalVO.getF0911Z1_CO());
      this.setNameAlphaExplanation(internalVO.getF0911Z1_EXA());
      this.setCurrencyCodeFrom(internalVO.getF0911Z1_CRCD());
      if ( internalVO.getF0911Z1_CRR()!= null){
         this.setCurrencyConverRateOv(internalVO.getF0911Z1_CRR().asBigDecimal());}
      this.setNameRemarkExplanation(internalVO.getF0911Z1_EXR());
      this.setSubledgerType(internalVO.getF0911Z1_SBLT());
      this.setLedgerType(internalVO.getF0911Z1_LT());
      this.setSubledger(internalVO.getF0911Z1_SBL());
      this.setBillCode(internalVO.getF0911Z1_BC());
      this.setEdiBatchNumber(internalVO.getF0911Z1_EDBT());
      if ( internalVO.getF0911Z1_U()!= null){
         this.setUnits(internalVO.getF0911Z1_U().asBigDecimal());}
      this.setUnitOfMeasure(internalVO.getF0911Z1_UM());
      this.setReference2(internalVO.getF0911Z1_R2());
```

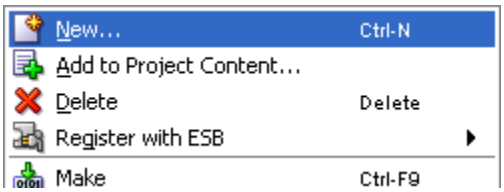
```
    this.setDateForGLandVoucherJULIA(Calendar.getInstance());
    if (internalVO.getF0911Z1_DGJ() != null)
       this.getDateForGLandVoucherJULIA().setTime(internalVO.getF0911Z1_DGJ());

    this.setDateServiceCurrency(Calendar.getInstance());
    if (internalVO.getF0911Z1_DSVJ() != null)
       this.getDateServiceCurrency().setTime(internalVO.getF0911Z1_DSVJ());

}


public void setEdiUserId(String ediUserId) {
   this.ediUserId = ediUserId;
}
```

12. Save all files in JDeveloper.

12. Add Business Service Class (J55911Z1)

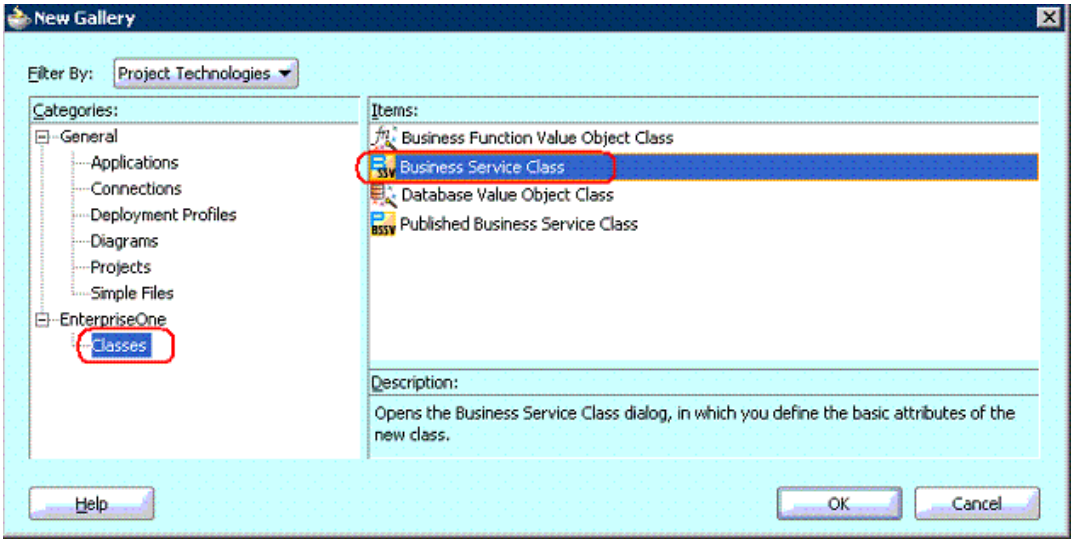
So far we have created Value Objects for both the Business Service and the Published Business Service. In this section we will create a Business Service Class.

1. Select the Business Service (**J55911Z1**) and right click. Select **New** from the menu.



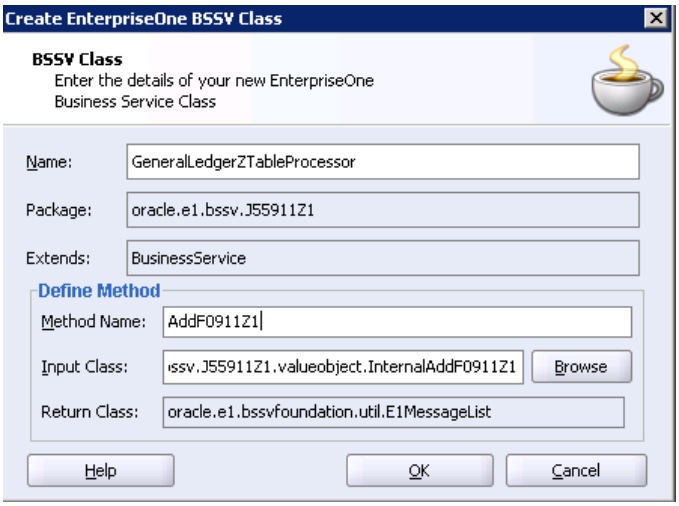
2. Under Categories, select **EnterpriseOne->Classes**.

   Select **Business Service Class** and press the **OK** button


New Gallery
Filter By:
Project Technologies
Categories:
General
Applications
Connections
Deployment Profiles
Diagrams
Projects
Simple Files
EnterpriseOne
Classes
Items:
Business Function Value Object Class
Business Service Class
Database Value Object Class
Published Business Service Class
Description:
Opens the Business Service Class dialog, in which you define the basic attributes of the new class.
Help
OK
Cancel

3. In the *Create EnterpriseOne BSSV Class* window, enter the following information

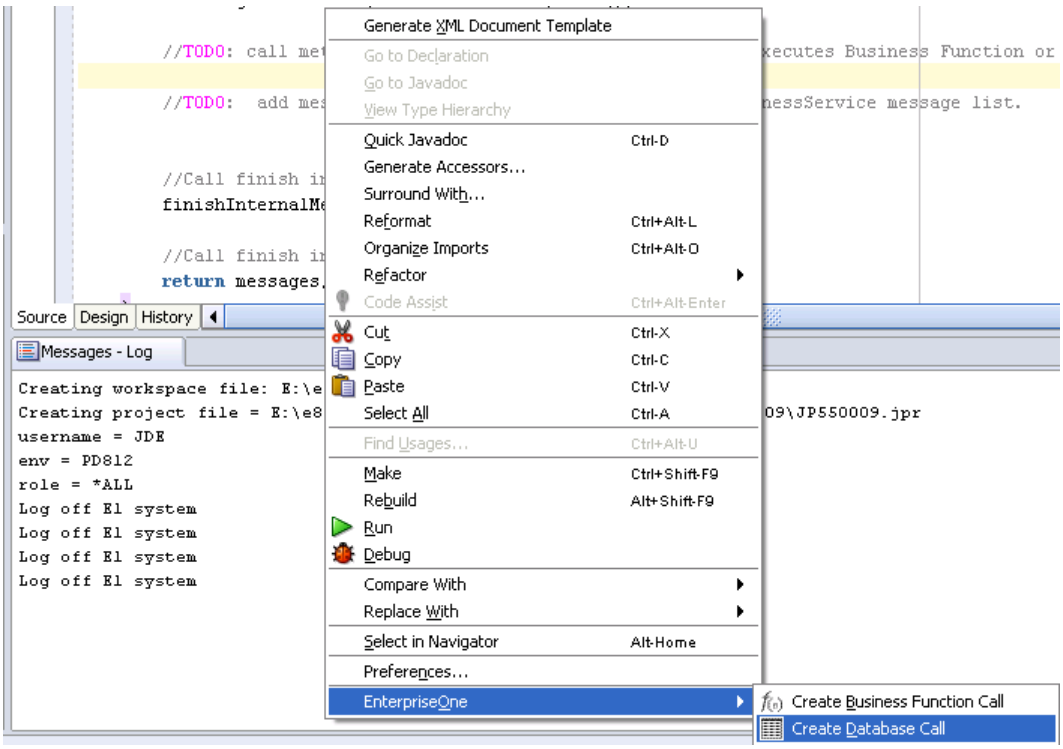
   Name: **GeneralLedgerZTableProcessor**
   Method Name: **AddF0911Z1**
   Input Class: **oracle.e1.bssv.J55911Z1.valueobject.InternalAddF0911Z1**
   (**Note**: Use the Browse button to select the Input Class)



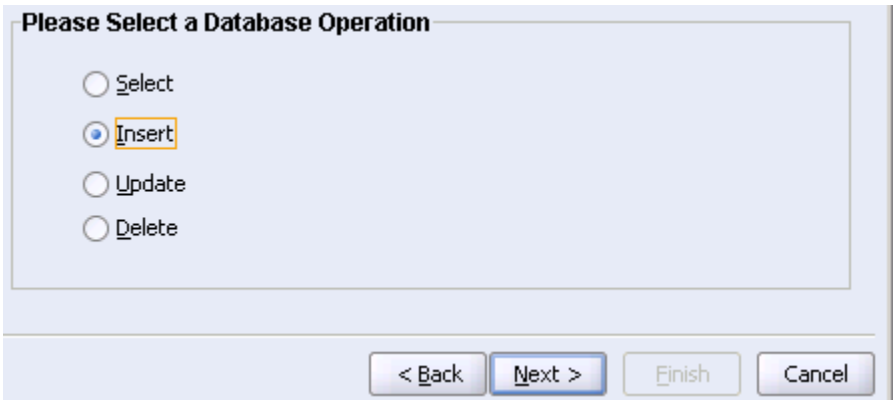
4. Press the **OK** button. The business service class template is created.

5. Look for the comment "***TODO: call method (created by the wizard), which then executes Business Function or Database operation.***"

   Place the cursor under this comment and **right-click** on your mouse. Select **EnterpriseOne -> Create Database Call** (see screenprint below).

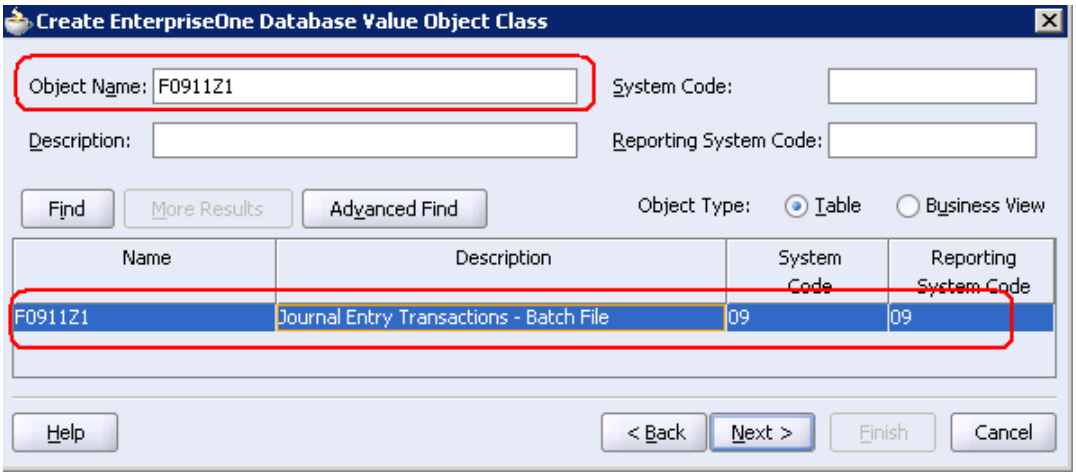


6. In the *Create EnterpriseOne Database Call* window, select **Insert** as the ***Operation***. Click the **Next** button.



7. In the *Create EnterpriseOne Database Call* window, enter the Object Name (**F0911Z1**) and press the **Find** button.

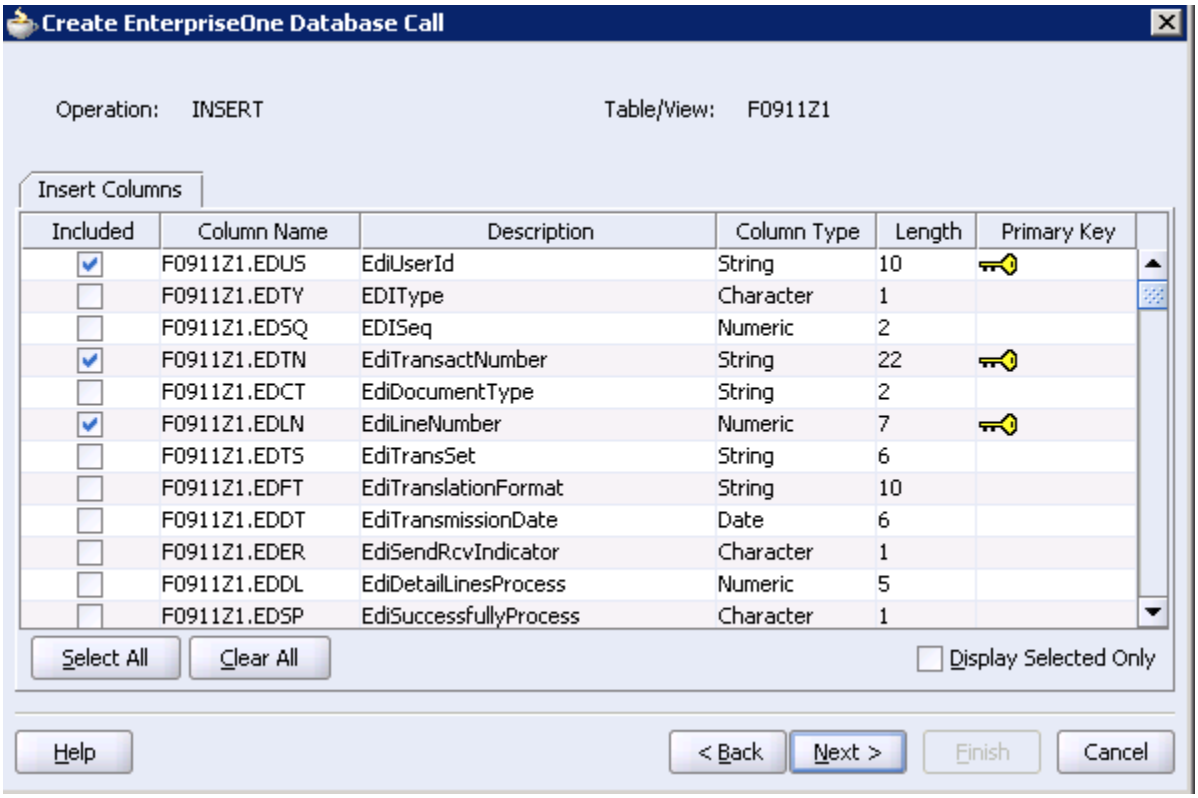
   Select the table from the results panel and press the **Next** button.



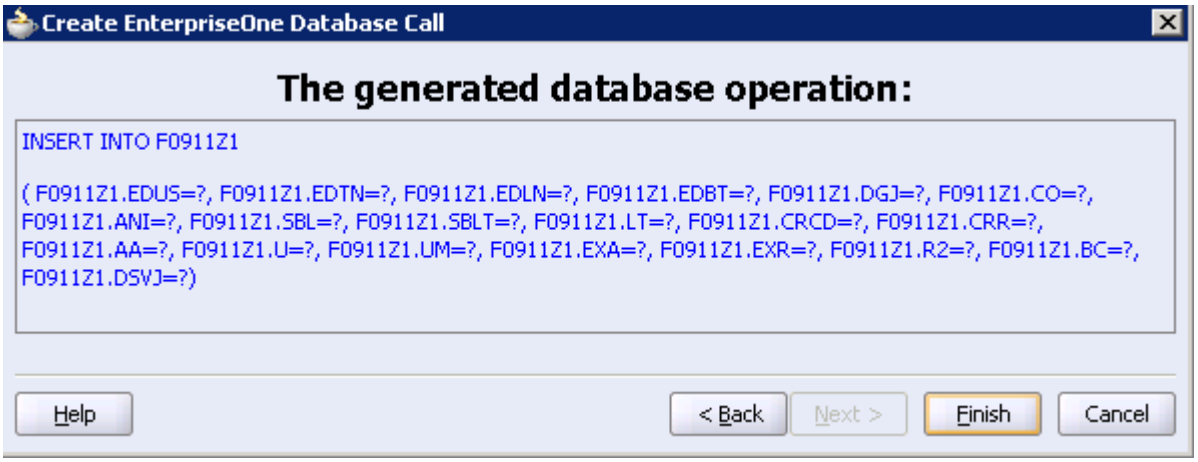
8. Select the following columns to be used in the **Insert** operation.

NAME	ALIAS
EDI - User ID	EDUS
EDI - Transaction Number	EDTN
EDI - Line Number	EDLN
EDI - Batch Number	EDBT
Date - For G/L (and Voucher) - Julian	DGJ
Company	CO
Account Number - Input (Mode Unknown)	ANI
Subledger - G/L	SBL
Subledger Type	SBLT
Ledger Types	LT
Currency Code - From	CRCD
Currency Conversion Rate - Spot Rate	CRR
Amount	AA
Units	U

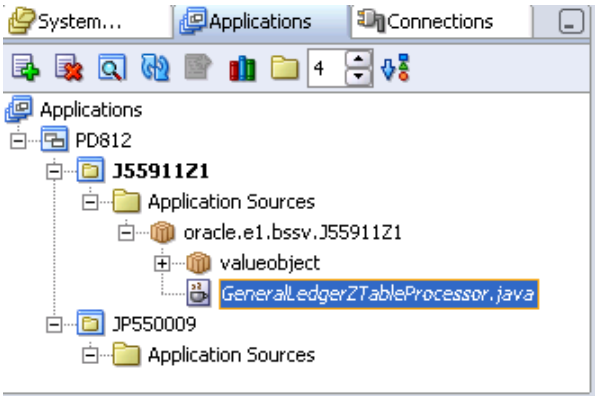
Unit of Measure	UM
Name - Alpha Explanation	EXA
Name - Remark Explanation	EXR
Reference 2	R2
Bill Code	BC
Date - Service/Tax	DSVJ



9. Click on the **Next** button. The generated SQL code will be displayed. Review it and click on the **Finish** button.



10. You should see the Business Service under the **Objects** node of your project



11. Change the following:

private static E1MessageList insertToF0911Z1(IContext context, IConnection connection, *InputVOType* internalVO) {
//create return object
E1MessageList returnMessages = new E1MessageList();

- to -

private static E1MessageList insertToF0911Z1(IContext context, IConnection connection, ***InternalAddF0911Z1*** internalVO) {
//create return object
E1MessageList returnMessages = new E1MessageList();

12. ***Save*** your JDeveloper files.

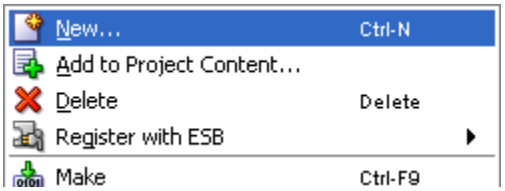
13. ***Compile*** the code by going to **Run -> Make J55911Z1.jpr** or by pressing the **Make** icon 

    You will receive a “Successful Compilation” message as shown below

13. Add Published Business Service Class (JP550010)

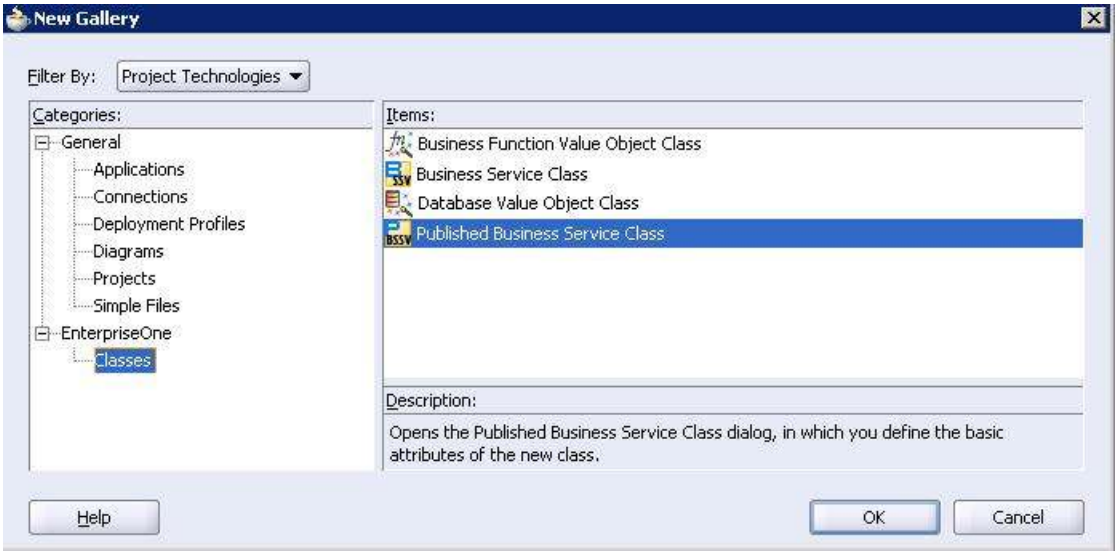
In this section of the lab we will add a Published Business Service class.

1. Select the Business Service (**JP550009**) and right click. Select **New** from the menu.



2. Under Categories, select **EnterpriseOne->Classes**.

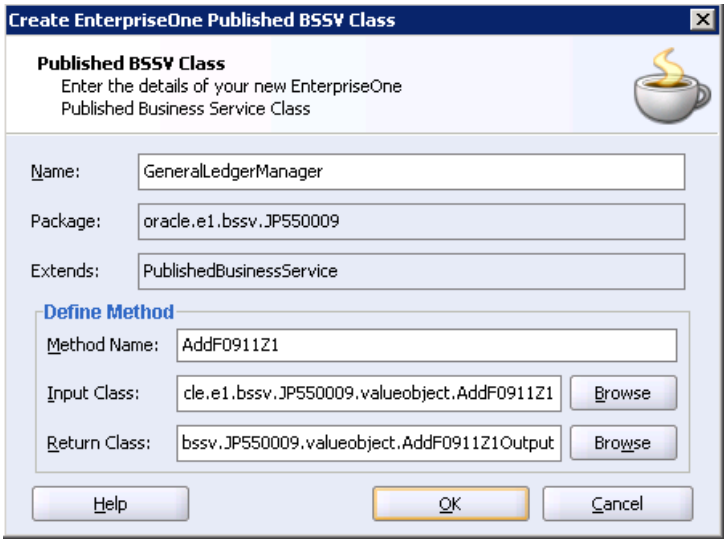
   Select **Published Business Service Class** and press the **OK** button. This will create the template for the Published Business Service class.



3. In the *Create EnterpriseOne Published BSSV Class* window, enter the following information

   Name: **GeneralLedgerManager**
   Method Name: **AddF0911Z1**
   Input Class: **oracle.e1.bssv.JP550009.valueobject.AddF0911Z1**
   Output Class: **oracle.e1.bssv.JP550009.valueobject.AddF0911Z1Output**

   (**Note**: Use the Browse button to select the Input and Output Class)



4. Press the **OK** button. The Publlished Business Service class template is created.

5. Add the following two import statements:

```
package oracle.e1.bssv.JP550009;

import oracle.e1.bssv.J55911Z1.valueobject.InternalAddF0911Z1;
import oracle.e1.bssv.J55911Z1.*;
import oracle.e1.bssv.JP550009.valueobject.AddF0911Z1;
import oracle.e1.bssv.JP550009.valueobject.AddF0911Z1Output;
import oracle.e1.bssvfoundation.base.IContext;
import oracle.e1.bssvfoundation.base.PublishedBusinessService;
import oracle.e1.bssvfoundation.connection.IConnection;
import oracle.e1.bssvfoundation.exception.BusinessServiceException;
import oracle.e1.bssvfoundation.util.E1MessageList;
```

6. The next step is to invoke the Business Service (J55911Z1) from the Published Business Service,.

   Add the following code highlighted in ***bold***.

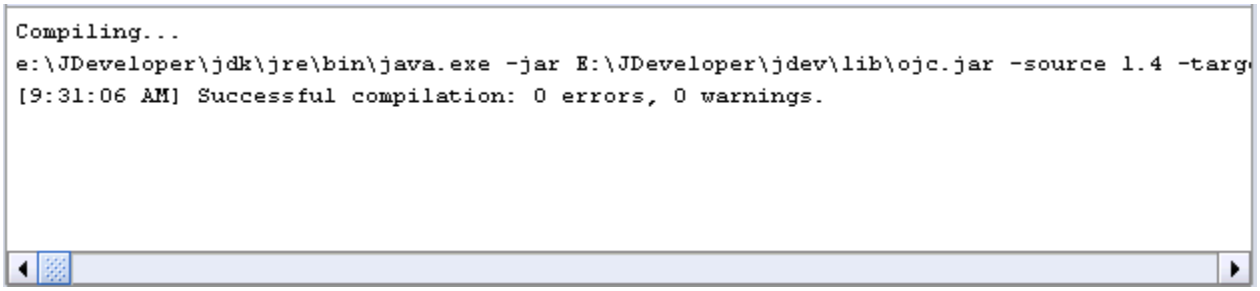
```
    protected AddF0911Z1Output AddF0911Z1(IContext context, IConnection connection, AddF0911Z1 vo)
throws BusinessServiceException {
        //perform all work within try block, finally will clean up any connections.
       InternalAddF0911Z1 internalVO = null;
        try {
            //Call start published method, passing context of null
            //will return context object so BSFN or DB operation can be called later.
            //Context will be used to indicate default transaction boundary, as well as access
            //to formatting and logging operations.
            context = startPublishedMethod(context, "AddF0911Z1", vo);

            //Create new PublishedBusinessService messages object for holding errors and warnings that occur
during processing.
            E1MessageList messages = new E1MessageList();
            //TODO: Create a new internal value object.
```

```
            internalVO = new InternalAddF0911Z1();

            vo.mapFromPublished(context, internalVO);

            //TODO: Call BusinessService passing context, connection and internal VO
            E1MessageList bssvMessages = GeneralLedgerZTableProcessor.AddF0911Z1(context,
connection, internalVO);

            //TODO: Add messages returned from BusinessService to message list for PublishedBusinessService.
           messages.addMessages(bssvMessages);

            //PublishedBusinessService will send either warnings in the Confirm Value Object or throw a
BusinessServiceException.
            //If messages contains errors, throw the exception

            if (messages.hasErrors()) {
               //Get the string representation of all the messages.
               String error = messages.getMessagesAsString();
               //Throw new BusinessServiceException
               throw new BusinessServiceException(error, context);
            }

            //Exception was not thrown, so create the confirm VO from internal VO
            AddF0911Z1Output confirmVO = new AddF0911Z1Output(internalVO);
            confirmVO.setE1MessageList(messages);
            finishPublishedMethod(context, "AddF0911Z1");
            //return outVO, filled with return values and messages
            return confirmVO;
         } finally {
            //Call close to clean up all remaining connections and resources.
            close(context, "AddF0911Z1");
         }
      }
```

7. Save your JDeveloper files.

8. Compile the code by going to **Run -> Make JP550009.jpr** or by pressing the **Make** icon 

   You will receive a “Successful Compilation” message as shown below

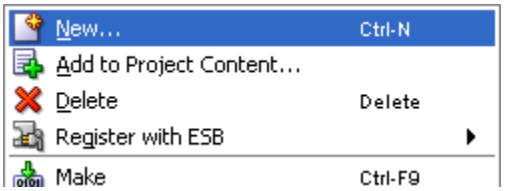


14. Creating a Test Class

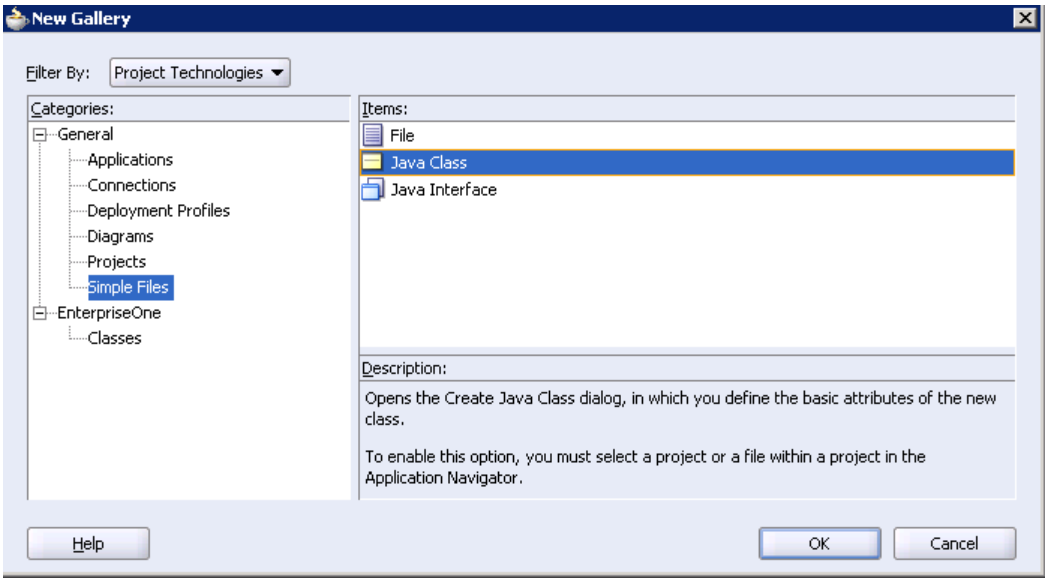
In this step we create a new java class to test our **Published Business Service**. Testing the business service from JDeveloper allows us to validate the accuracy of our code prior to checking it in.

1. Select the Published Business Service project (**JP550009**) and *right click* on it.

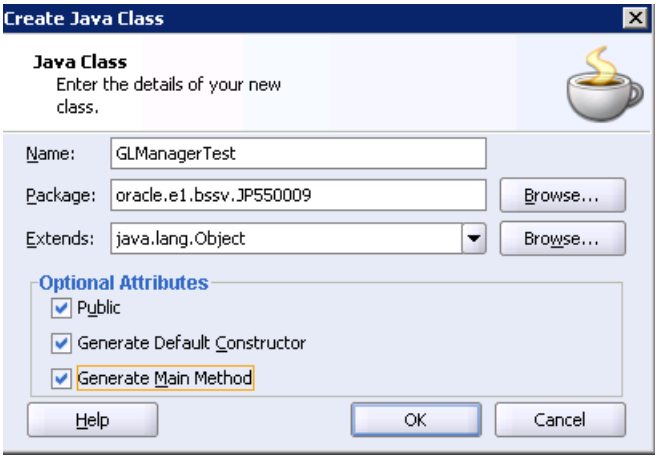
2. Select **New** from the menu.



3. In the Categories panel, select **General -> Simple Files**. Then select **Java Class** from the Items panel. Click **OK**.



4. In the *Create Java Class window*, enter the class name as **GLManagerTest**. Select all **three optional attributes** (i.e. *Public, Generate Default Constructor and Generate Main Method*). Leave the Package and Extends to the default values. Click on the **OK** button to create the class.



5. Add the following **import** statements highlighted in bold.

```
package oracle.e1.bssv.JP550009;

import java.math.BigDecimal;
import java.util.Calendar;
import oracle.e1.bssv.JP550009.valueobject.AddF0911Z1;
import oracle.e1.bssv.JP550009.valueobject.AddF0911Z1Output;
import oracle.e1.bssvfoundation.exception.BusinessServiceException;
import oracle.e1.bssvfoundation.util.E1MessageList;

public class GLManagerTest {
```

6. Remove or disable the following code in the **main** method.

```
/*  GLManagerTest gLManagerTest = new GLManagerTest(); */
```

7. Add the following code inside the **main** method

```
public static void main(String[] args) {
   /* GLManagerTest gLManagerTest = new GLManagerTest(); */
    /* add code here to map output from internal vo to the output-pub-biz-service-vo */
     AddF0911Z1 p = new AddF0911Z1();
     p.setEdiUserId("JDE");
     p.setAmountField(new BigDecimal(100.22));
     Calendar c = Calendar.getInstance();
     c.set(2008,04,28);
     p.setDateForGLandVoucherJULIA(c);
     p.setEdiTransactNumber("111");
     p.setEdiLineNumber(new BigDecimal(1));
     p.setEdiBatchNumber("101");

     // p.setEdiBatch("*");
     GeneralLedgerManager glm = new GeneralLedgerManager();
     E1MessageList el;
     AddF0911Z1Output s = null;
```

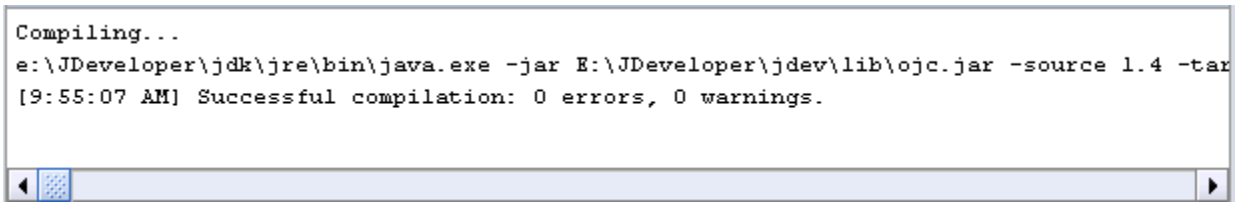
```
        try {
            s = glm.AddF0911Z1(p);
        } catch (BusinessServiceException e) {
            e.printStackTrace(System.out);
        }
        finally {
            System.out.println(s.toString());
        }
}
```

In the above code we are initializing a few input values and invoking the Published Business Service. The output of the service will be printed to the screen.

8. Save your test file.

9. Compile the code by going to **Run -> Make GLManagerTest.jpr** or by pressing the **Make** icon 

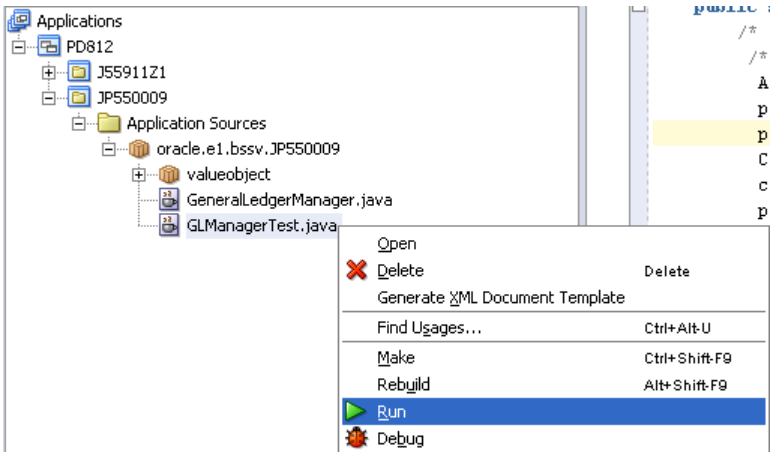
   You will receive a "Successful Compilation" message as shown below



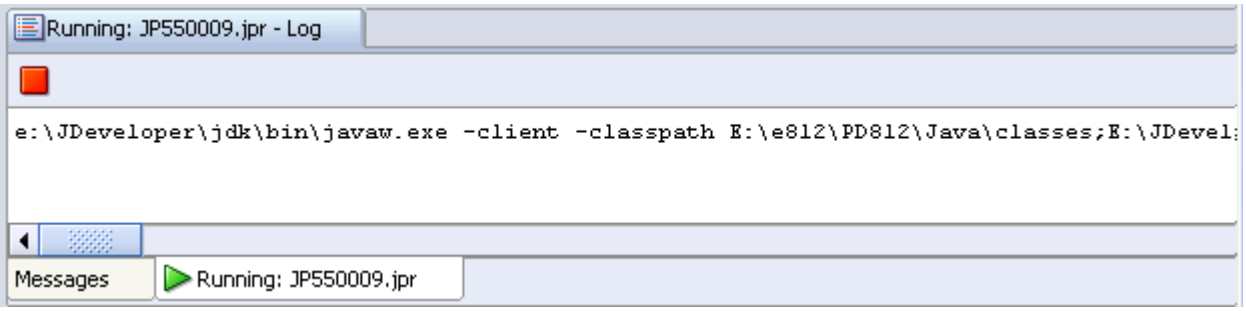
15. Running the Test Class

Now that we have created the test class and successfully compiled it, we can now test it.

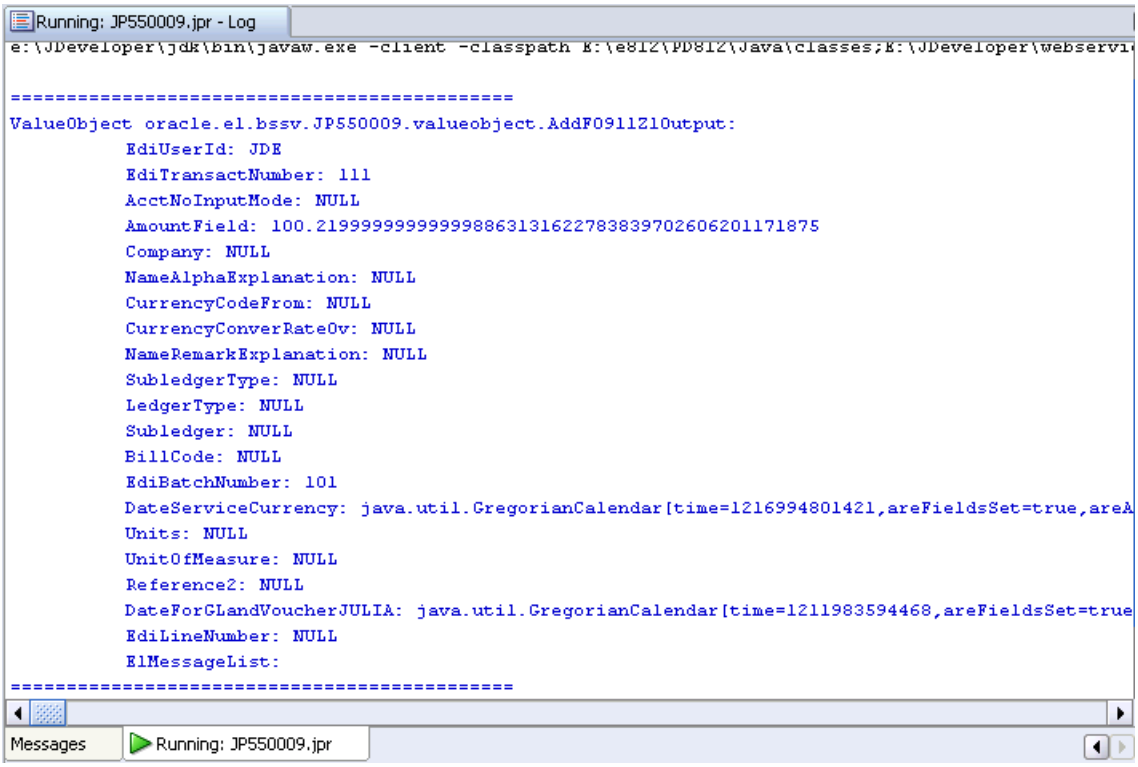
1. Select **CreditCheckTest.java** and *right-click* on it. Select **Run** from the menu.



2. You will notice a **new tab (Running: JP550009.jpr)** that shows the details of the execution of the program.

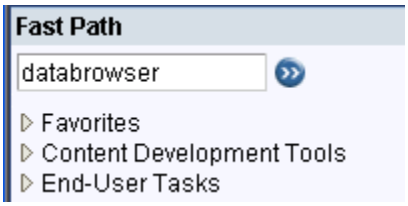


3. When the execution has completed you will see the results in the same tab. The result will look similar to the screenshot below.

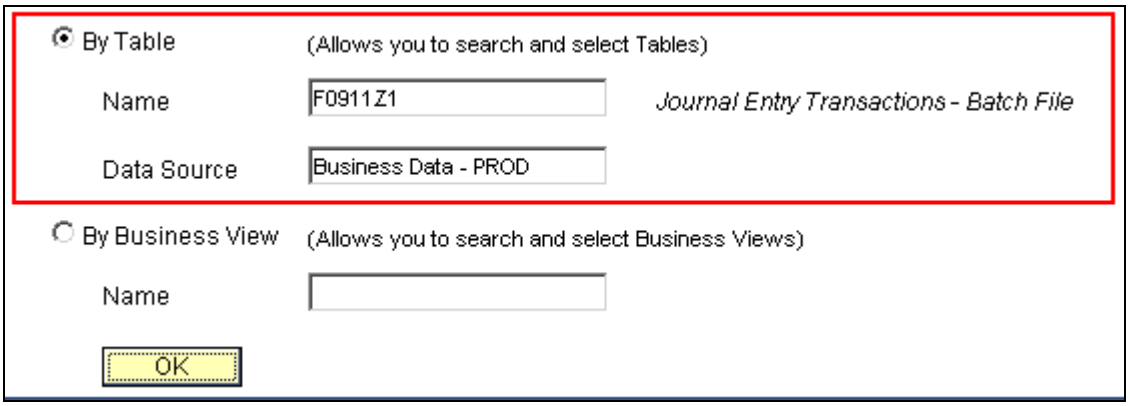


4. You can confirm these results by logging into JDE E1.

5. Enter **databrowser** in the Fast Path and press **Enter**.

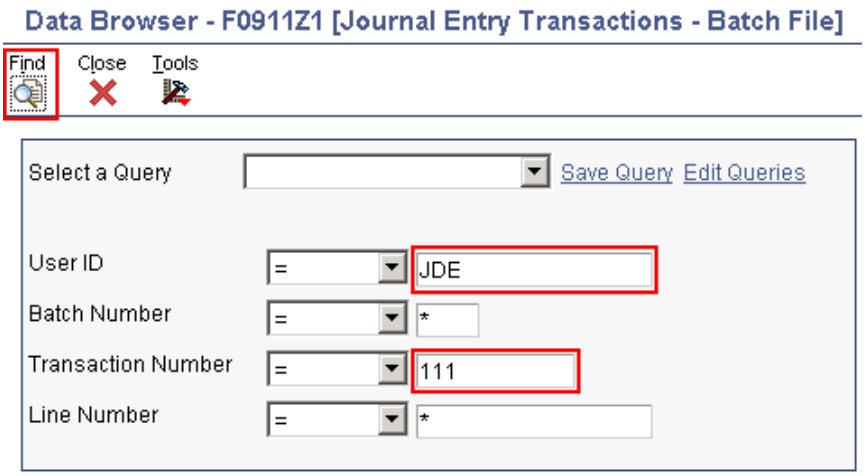


6. In the *Query Selector*, enter table as **F0911Z1.** The data source will be automatically selected. Click on the **OK** button.

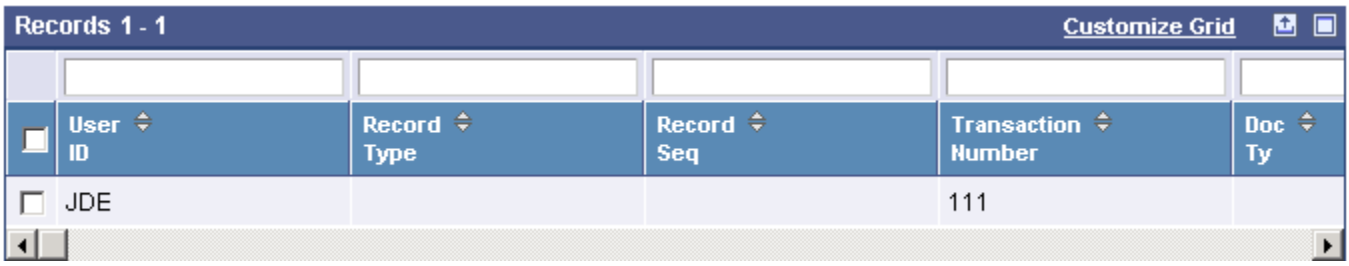


7. In the Data Browser for F0911Z1, enter the following search values and click the **Find** button.

   User ID = JDE
   Transaction Number = 111
   Note: These are the values that we hardcoded in our test class.



8. The results grid will show one record with the values that we passed in from our test java class.



9. [***Optional***] You can modify the test java class by setting the inputs to a different set of values. Compile the test class and then Run it. You can then look up the results in the F0911Z1 table as outlined in the steps above.

10. The Business Service has been successfully tested and ready for check-in to OMW.

16. Review and Next Steps

In this lab we have learnt to create:

- OMW Projects, Published Business Service Object and Business Service Object in OMW
- Accessing Oracle JDeveloper from OMW
- Using the JDE EnterpriseOne plug-in for JDeveloper
- Creating Value Objects (Database Value Object)
- Creating Business Service and Published Business Service Classes
- Test Class to validate the business service and validating the results in J.D. Edwards E1.

Now that we have successfully created the Business Service, here are the next steps:

- **Check-In**: From OMW, you can now check-in the two objects (J55911Z1 and J5500009). The check-in process is the same as any other object. However in case of a business service, OMW will check-in not only the object but all the related artifacts related to the business service. OMW will also do a validation during the check-in process. The Business Service source code is located in: \E1_Home_Dir\Path_Code\java\source\oracle\e1\bssv (e.g. E:\E812\PD812\java\source\oracle\e1\bssv)
- **Package Build**: After the Business Service is checked in, the next step is to build and deploy the package. The package build creates the necessary .ear and .wsdl files. They are then deployed on the J.D. Edwards Business Services Server.
- **Look up the WSDL**: To be able to consume the business service the first step is to locate the WSDL for the particular business service. You can do this by logging into the OAS Enterprise Manager.
- **Building a BPEL/ESB Process**: You can now create a BPEL/ESB process that will consume the J.D. Edwards Business Service.