JDE EnterpriseOne Business Services

Consuming External Web Services Tutorial
VERSION HISTORY

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Updated by</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-Jul-2009</td>
<td>1.0</td>
<td>Dave Harrison</td>
<td>Document created</td>
</tr>
<tr>
<td>7-Dec-2009</td>
<td>2.0</td>
<td>Dave Harrison</td>
<td>Location of web service changed. Revised solution for handling situation where no HTTP proxy server is used. Changed Metalink3 links to Oracle Support.</td>
</tr>
</tbody>
</table>
1.0 Introduction

EnterpriseOne has had the ability to provide support for web services for a number of years now via the use of Web Services Gateway and XPI, but the process required to consume external web services was a tricky one and fraught with complications. Firstly, it was necessary to activate the Real Time Events (RTE) system in order to trigger either a standard JDE event, or a new one created by developers using calls to the RTE APIs. The RTE would send a JMS notification message to a Transaction Server running in Oracle Application Service or IBM Websphere, which would in turn forward the message onto the Web Services Gateway Integration Server, before finally consuming the external web service. If a response was required, then a different type of RTE could be used, called an XAPI event, whereby the calling business function would wait for the response to come back to EnterpriseOne via a different route (using the XML Dispatch and XML Service Kernels). This multi-step approach meant that any failure in the process would involve substantial investigation work to track down the cause.

Now, with Tools Release 8.97 and the ability for EnterpriseOne to natively support web services, this process is far more straightforward. It is now possible for a business function to directly call a business service (BSSV) that can then call out to the external web service. Furthermore, this process is fully synchronous, allowing the response to be sent directly back to the business function. An option for an asynchronous call is also available.

In this tutorial, we will be creating a Business Service which calls a simple 'Fortune' web service that returns an amusing or interesting random message based on a topic/category (such as 'computers' or 'sports') that is passed to the web service. There are a number of EnterpriseOne objects required for this process, viz.:
2.0 Prerequisites
Several steps need to be completed before you can create business service objects. These will normally be performed by your Systems Administrator. If you haven’t already done so, you should ensure that:

- You have an 8.11SP1, 8.12 or 9.0 Development Client installed on your workstation running at Tools Release 8.97 or above. This tutorial is based on an 8.12 environment, but can be adapted to other releases.
- Microsoft Visual Studio .NET has been installed on your workstation.
- JDeveloper 10.1.3.1 has been installed on your workstation.
- The Tools Baseline ESUs have been installed and deployed to your development client. These contain the enhancements to OMW and other applications to support business services.
- The Business Services Temporary tables F986020 and F986030 have been generated in your local MSDE/SQL2005 Express database.
- The configuration files (jdbj.ini, jdeinterop.ini and jdelog.properties) that exist in your C:\e812\<pathcode>\ini\sbf folder have been configured correctly. The tnsnames.ora file will also be required to communicate with an Oracle database.
- The appropriate JDBC driver jar files for your E1 databases exist in the c:\e812\MISC folder.
- The standard JDE Reference Implementations have been installed. Although it is not strictly necessary to install these to create business services, they provide an excellent model for the creation of your own business services.

3.0 Creating the OMW Project

**Summary:**
- Add a new Project in OMW to contain the objects required for this tutorial
- Type OMW in the Fastpath of the JD Edwards Solutions Explorer and press Enter to launch Object Management Workbench:
• In OMW, click on the **Add** button on the toolbar:

![OMW Add Button](image1.png)

• Select the **OMW Project** radio button on the Add EnterpriseOne Object to OMW Project screen and click **OK**:

![OMW Project Selection](image2.png)

• On the Project Revisions screen, enter the following information and click on the **OK** button:

<table>
<thead>
<tr>
<th>Project ID:</th>
<th>BSSVTEST01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>BSSV Tutorial</td>
</tr>
<tr>
<td>Type:</td>
<td>02</td>
</tr>
<tr>
<td>Severity:</td>
<td>01</td>
</tr>
<tr>
<td>System Code:</td>
<td>55</td>
</tr>
<tr>
<td>Release:</td>
<td>E812</td>
</tr>
</tbody>
</table>
• The project is now added. The default project status is 21 (Programming) and User Role is Developer. Press the Find button on the toolbar to see the newly added project:

![Image of Object Management Workbench](image)

4.0 Create the Business Function Data Structure

**Summary:**
- Create a Data Structure for the Business Function that will call the Business Service Class

We need to create the business function first, so that we can use its definition to create the Value Object class in the step 10.0. First of all create the Data Structure for the Business Function.

- Select the OMW project **BSSVTEST01** that we've just created and click the Add button on the toolbar.
- Select the Data Structure radio button from the Add EnterpriseOne Object to the Project screen and press OK:

![Screenshot of Add Object screen](image)

- Enter the following information on the Add Object screen and press OK:

<table>
<thead>
<tr>
<th>Object Name:</th>
<th>D5500001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Call Get Fortune Business Service</td>
</tr>
<tr>
<td>Product Code:</td>
<td>55</td>
</tr>
<tr>
<td>Product System Code:</td>
<td>55</td>
</tr>
<tr>
<td>Object Use:</td>
<td>360 (Data Structures)</td>
</tr>
<tr>
<td>Type:</td>
<td>Regular Data Structure</td>
</tr>
</tbody>
</table>
• On the Design Tools tab of the Data Structure Design screen, select the **Data Structure Design** icon to launch the Data Structure Design tool:

![Data Structure Design Icon]

- Event Definition Workbench
- Create a type definition
- Named Mapping

• We want to pass a Topic name to the Business Function so that this can be passed on to the Web Service. Enter ‘**AA30**’ in the Alias column QBE field and press **Enter**:  

![Data Structure: Bus. Function DS DS500001]

- Structure Members
- Dictionary Items

Aliases: AA30 |
<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha Spec Data 30</td>
<td>Alpha Spec Data 30</td>
</tr>
</tbody>
</table>
Select the row shown and drag it to the left hand Structure Members pane:

- Select the ‘szAlphaSpecData30’ text in the Structure Member Name grid cell and rename it to ‘szTopic’. Also click on the Input/Output/IO icon to change it to ➩ (Input):

- Repeat the above process for the following Data Structure Members:

<table>
<thead>
<tr>
<th>Alias</th>
<th>Structure Member Name</th>
<th>Input/Output/IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC2000</td>
<td>szFortune</td>
<td>➩</td>
</tr>
<tr>
<td>SUPPS</td>
<td>cSuppressErrorMessage</td>
<td>➩</td>
</tr>
<tr>
<td>DTAI</td>
<td>szErrorMessageId</td>
<td>➩</td>
</tr>
<tr>
<td>MSGTEXT</td>
<td>szErrorMessageText</td>
<td>➩</td>
</tr>
</tbody>
</table>
• The screen should now look like this:

![Graphical representation of the screen showing data structure and dictionary items]

• Press OK to return to the OMW Data Structure Design screen.
• Press OK to return to the main OMW screen.

5.0 Create the Business Function Object

Summary:
Create the Business Function Object that will format the XML document to send to the Business Service

Normally, the easiest way to create this function would be to base it on the sample Reference Implementation function provided by Oracle, B953002. However, for the purposes of this tutorial, the code will be provided. The business function uses system APIs jdeCallBusinessService or jdeCallBusinessServiceNoResponse to call the Business Service class. It also uses the Xerces APIs to parse the data into XML for passing to the class via the Value Object.

Note that if it is likely that a lot of data will need to be passed via the Value Object, then this could place large memory requirements on the Enterprise Server where this function is running. An alternative solution may need to be considered that will pass the data to the Business Service via Table I/O or a Business Function call from the Business Service.

• Select the OMW Project BSSVTEST01 and press the Add button. Select the Business Function radio button from the Add EnterpriseOne Object to the Project screen and press OK:
• Enter the following information on the Add Object screen and press OK:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Name</td>
<td>B5500001</td>
</tr>
<tr>
<td>Description</td>
<td>Call Get Fortune Business Service</td>
</tr>
<tr>
<td>Product Code</td>
<td>55</td>
</tr>
<tr>
<td>Product System Code</td>
<td>55</td>
</tr>
<tr>
<td>Object Use</td>
<td>194 (Common Subroutine)</td>
</tr>
<tr>
<td>Source Language</td>
<td>C</td>
</tr>
<tr>
<td>Function Location</td>
<td>Client/Server</td>
</tr>
</tbody>
</table>

• On the Design Tools tab of the Business Function Design screen, select the **Start Business Function Design Aid** icon:
• Enter ‘callGetFortuneBSSV’ in the Function Name and ‘Call Get Fortune Business Service’ in the Description:

• Select the row and press the Parameters row exit
• Enter **D5500001** in the Template Name column of the QBE line, and press the **Find** button:

![Object Management Workbench - [Select Business Function Parameters]](object_management_workbench.png)

• Select the row and press the **Select** button to return to the Business Function Design screen.

• Press the **Create** form exit, and answer **Yes** to the following prompts:

![Business Function Librarian](business_function_librarian1.png)

• Click **OK** on the following prompt:

![Business Function Librarian](business_function_librarian2.png)

• Ensure that the row is still selected and press the **Typedef** row exit. The message ‘Your typedef is in the clipboard’ will be displayed in the status area of the window. This copies the data structure definition into the clipboard which we will then paste into the business function source.
• Press the **Edit** form exit to launch the Microsoft Development Environment [design] screen. The generated C source code will be displayed:

```c
#define b5500001_c

/*****************************************************************************/
* Source File:  B5500001
* Description:  Call Get Fortune Business Service Source File
* History:  
<table>
<thead>
<tr>
<th>Date</th>
<th>Programmer</th>
<th>SAM# - Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
* Author 01/07/2009  Unknown  Unknown - Created
* Copyright © J.D. Edwards World Source Company, 1996
* This unpublished material is proprietary to J.D. Edwards World Source Company.
* All rights reserved. The methods and techniques described herein are
* considered trade secrets and/or confidential. Reproduction or
* distribution, in whole or in part, is forbidden except by express
* written permission of J.D. Edwards World Source Company.
/*****************************************************************************/
* Notes:

#include <b5500001.h>

/*****************************************************************************/
* DS Template Type Definitions
*****************************************************************************/
```

• Select the B5500001.h tab to view the include file for the function. Paste the typedef that was copied into the clipboard into the include file after the DS Template Type Definitions heading:
• Your code will look like this:
• We also need to add the Typedef for data structure DE954000 to the .h file. This is used to send any errors to the 007FIS error message. Copy the following code and paste after the D5500001 typedef (after the #endif):

```c
/**
 *_TYPEDEF for Data Structure
 *    Template Name: Error Text For WS Reference Implementation
 *    Template ID:   DE954000
 * *
 * DO NOT EDIT THE FOLLOWING TYPEDEF
 * To make modifications, use the OneWorld Data Structure Tool to Generate a revised version, and paste from the clipboard.
 */

#ifndef DATASTRUCTURE_DE954000
#define DATASTRUCTURE_DE954000

typedef struct tagDSDE954000
{
    JCHAR szWSCallExceptionInfo[1025];
} DSDE954000, *LPDSDE954000;
#endif

#define IDERRszWSCallExceptionInfo_2              2L

/**
 * Add an include statement for the Xerces API prototypes after the External Business Function Header Inclusions heading:
 */

#include <xerceswrapper.h>
```
#include <jde.h>

define b5500001_c

/**************************************************************************
* Source File:  b5500001
* Description:  Call Get Fortune Business Service Source File
* History:
* Date        Programmer  SAR# - Description
* -----------  ----------  -------------------------------------------
* Author 01/07/2009  DaveH       Oracle Fusion Middleware for Apps Best Practice Centre Tutorial
* Copyright (c) J.D. Edwards World Source Company, 1996
* This unpublished material is proprietary to J.D. Edwards World Source Company.
* All rights reserved. The methods and techniques described herein are
* considered trade secrets and/or confidential. Reproduction or
* distribution, in whole or in part, is forbidden except by express
* written permission of J.D. Edwards World Source Company.
**************************************************************************/

#include <b5500001.h>

/**************************************************************************
* Business Function:  callGetFortuneBSSV
* Description:  Call Get Fortune Business Service
* Parameters:
* LPBHVRCOM        lpBhvrCom    Business Function Communications
* LPVOID           lpVoid       Void Parameter - DO NOT USE!
* LPDS5500001      lpDS        Parameter Data Structure Pointer
**************************************************************************

JDEBFRTN (ID) JDEBFWINAPI callGetFortuneBSSV (LPBHVRCOM lpBhvrCom, LPVOID lpVoid, LPDS5500001 lpDS)
{
    #ifdef SOA_ENABLED
        /**************************************************************************
        * Variable declarations
        **************************************************************************

        XRCS_hParser      hParser   = NULL;
        XRCS_Status       XRCSStatus = XRCS_SUCCESS;
        XRCS_hDocument    hDoc      = NULL;
        XRCS_hDocument    hBSSVDoc  = NULL;
        XRCS_hElement     hRootElm  = NULL;
    */
XRCS_hElement *hElm = NULL;
unsigned int nElmCount = 0;
XRCS_hElement *hChildElms = NULL;
unsigned int nChildCount = 0;
JCHAR *elmName = NULL;
JCHAR *elmValue = NULL;
unsigned int i = 0;
ID idReturnValue = ER_SUCCESS;
JCHAR *bssvPayloadReturn = NULL;
JCHAR *newXMLString = NULL;
JCHAR *errorText = NULL;
JCHAR szSuppressErrorMessage[2];
JCHAR *xmlString =
 J("<?xml version="1.0"?>")
 J("<get-fortune-vO>")
 J("<sz-fortune></sz-fortune>")
 J("<sz-topic></sz-topic>")
 J("<CSuppressErrorMessage></CSuppressErrorMessage>")
 J("<sz-error-message-id></sz-error-message-id>")
 J("<sz-error-message-text></sz-error-message-text>")
 J("</get-fortune-vO>");

/************************************************************************
 * Declare structures
*************************************************************************/
DSDE954000 dsDE954000 = {0};

/************************************************************************
 * Declare pointers
*************************************************************************/

/************************************************************************
 * Check for NULL pointers
*************************************************************************/
if ((lpBhvrCom == (LPBHVRCOM) NULL) ||
 (lpVoid == (LPVOID) NULL) ||
 (lpDS == (LPDSD5500001) NULL))
 {
 jdeErrorSet (lpBhvrCom, lpVoid, (ID) 0, _J("4363"), (LPVOID) NULL);
 return ER_ERROR;
 }

/************************************************************************
 * Set pointers
*************************************************************************/

/************************************************************************
 * Main Processing
*************************************************************************/
jdeStrcpy((JCHAR *)lpDS->szErrorMessageId,(const JCHAR *)_J(" "));
jdeStrcpy((JCHAR *)lpDS->szErrorMessageText,(const JCHAR *)_J(" "));

/* Initialize Xerces Engine */
XRCSStatus = XRCS_initEngine();
if(XRCSStatus != XRCS_SUCCESS)
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_initEngine failed"));  
  return ER_ERROR;  
}

/* Get the Xerces Parser */
XRCSStatus = XRCS_getParser(&hParser);
if(XRCSStatus != XRCS_SUCCESS)  
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_getParser failed"));  
  XRCS_terminateEngine();  
  return ER_ERROR;  
}

/* Parse the XML String */
XRCSStatus = XRCS_parseXMLString(hParser, xmlString, &hDoc);
if(XRCSStatus != XRCS_SUCCESS)  
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_parseXMLString failed"));  
  XRCS_freeParser(hParser);
  XRCS_terminateEngine();  
  return ER_ERROR;  
}

/* Get Root Element */
XRCSStatus = XRCS_getDocumentElement(hDoc,&hRootElm);
if(XRCSStatus != XRCS_SUCCESS)  
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_getDocumentElement failed"));  
  XRCS_freeParser(hParser);
  XRCS_freeDocument(hDoc);
  XRCS_terminateEngine();  
  return ER_ERROR;  
}

/* Get the topic value element and set its value to passed in topic parameter */
XRCSStatus = XRCS_getElementsByTagName(hRootElm, _J("sz-topic"), &hElm,&nElmCount);
if(XRCSStatus != XRCS_SUCCESS)  
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_getElementsByTagName failed"));  
  XRCS_freeParser(hParser);
  XRCS_freeDocument(hDoc);
  XRCS_freeElement(hRootElm);
  XRCS_freeElementArray(hElm,nElmCount);
  XRCS_terminateEngine();  
}

XRCSStatus = XRCS_setElementText(hElm[0], lpDS->szTopic);
if(XRCSStatus != XRCS_SUCCESS)  
{  
  jdeVWriteLogEntry(_J("B55000001"), __FILE__, __LINE__, 0, _J("XRCS_setElementText failed"));  
  XRCS_freeParser(hParser);
  XRCS_freeDocument(hDoc);
  XRCS_freeElement(hRootElm);
  XRCS_freeElementArray(hElm,nElmCount);
  XRCS_terminateEngine();  
}
return ER_ERROR;
}

/* Get the Suppress Error Messages value element and set its value to passed in Suppress Error Message parameter */
XRCSStatus = XRCS_getElementsByTagName(hRootElm, _J("<CSuppressErrorMessage"), &hElm,&nElmCount);
if(XRCSStatus != XRCS_SUCCESS)
{
    jdeWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_getElementsByTagName failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElement(hRootElm);
    XRCS_terminateEngine();
    return ER_ERROR;
}

szSuppressErrorMessage[0] = lpDS->cSuppressErrorMessage;
szSuppressErrorMessage[1] = _J('"0');

XRCSStatus = XRCS_setElementText(hElm[0], szSuppressErrorMessage);
if(XRCSStatus != XRCS_SUCCESS)
{
    jdeWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_setElementText failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElement(hRootElm);
    XRCS_freeElementArray(hElm,nElmCount);
    XRCS_terminateEngine();
    return ER_ERROR;
}

/* Serialize the XML DOC to XML String */
XRCSStatus = XRCS_serializeDocumentToXMLStringNoEncoding(hDoc, &newXMLString);
if(XRCSStatus != XRCS_SUCCESS)
{
    jdeWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_serializeDocumentToXMLStringNoEncoding failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElement(hRootElm);
    XRCS_freeElementArray(hElm,nElmCount);
    XRCS_terminateEngine();
    return ER_ERROR;
}

/* call the Business Service */
idReturnValue = jdeCallBusinessService(lpBhvrCom,
    lpVoid,
    _J("oracle.el.bssv.J5500001.FortuneProcessor"),
    _J("getFortune"),
    TRUE,
    newXMLString,
    &bssvPayloadReturn);

if ( idReturnValue == CallBSSVNoError || idReturnValue == CallBSSVNoErrorWithMessages)
{
    XRCSStatus = XRCS_parseXMLStringRemoveEncoding(hParser, bssvPayloadReturn, &hBSSVDoc);
    if(XRCSStatus != XRCS_SUCCESS)


```c
XRCSStatus = XRCS_getDocumentElement(hBSSVDoc,&hRootElm);
if(XRCSStatus != XRCS_SUCCESS)
{
    jdeVWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_getDocumentElement failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElementArray(hElm,nElmCount);
    XRCS_freeDocument(hBSSVDoc);
    XRCS_terminateEngine();
    jdeFree(newXMLString);
    jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
    return ER_ERROR;
}

XRCSStatus = XRCS_getElementChildren(hRootElm,&hChildElms,&nChildCount);
if(XRCSStatus != XRCS_SUCCESS)
{
    jdeVWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_getElementChildren failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElementArray(hElm,nElmCount);
    XRCS_freeDocument(hBSSVDoc);
    XRCS_freeElement(hRootElm);
    XRCS_terminateEngine();
    jdeFree(newXMLString);
    jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
    return ER_ERROR;
}

for ( i=0;i<nChildCount;i++)
{
    XRCSStatus = XRCS_getElementName(hChildElms[i],&elmName);
    if(XRCSStatus != XRCS_SUCCESS)
    {
        jdeVWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_getElementName failed"));
        XRCS_freeParser(hParser);
        XRCS_freeDocument(hDoc);
        XRCS_freeElementArray(hElm,nElmCount);
        XRCS_freeDocument(hBSSVDoc);
        XRCS_freeElement(hRootElm);
        XRCS_freeElementArray(hChildElms,nChildCount);
        XRCS_terminateEngine();
        jdeFree(newXMLString);
        jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
        return ER_ERROR;
    }
}
```
XRCSStatus = XRCS_getElementText(hChildElms[i], &elmValue);
if (XRCSStatus != XRCS_SUCCESS) {
    jdeWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, _J("XRCS_getElementText failed"));
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElementArray(hElm, nElmCount);
    XRCS_freeElement(hRootElm);
    XRCS_freeElementArray(hChildElms, nChildCount);
    XRCS_terminateEngine();
    XRCS_freeString(elmName);
    jdeFree(newXMLString);
    jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
    return ER_ERROR;
}
if ((JCHAR*)NULL != elmValue) {
    if (jdeStricmp(elmName, _J("sz-fortune")) == 0) {
        jdeStrncpyTerminate(lpDS->szFortune, elmValue, DIM(lpDS->szFortune));
    } else if (jdeStricmp(elmName, _J("sz-error-message-id")) == 0) {
        jdeStrncpyTerminate((JCHAR *)lpDS->szErrorMessageId, elmValue, DIM(lpDS->szErrorMessageId));
    } else if (jdeStricmp(elmName, _J("sz-error-message-text")) == 0) {
        jdeStrncpyTerminate((JCHAR *)lpDS->szErrorMessageText, elmValue, DIM(lpDS->szErrorMessageText));
    }
    XRCS_freeString(elmName);
    XRCS_freeString(elmValue);
    elmName = (JCHAR*)NULL;
    elmValue = (JCHAR*)NULL;
} /*end of for loop*/
else {
    memset((void *)(&dsDE954000), (int)(_J('0')), sizeof(dsDE954000));
    errorText = jdeGetBusinessServiceErrorText(idReturnValue);
    jdeWriteLogEntry(_J("B5500001"), __FILE__, __LINE__, 0, errorText);
    jdeStrncpy(dsDE954000.szWSCallExceptionInfo, errorText, DIM(dsDE954000.szWSCallExceptionInfo));
    jdeStrncpy(lpDS->szErrorMessageId, (const JCHAR *)_J("007FIS"));
    jdeStrncpy(lpDS->szErrorMessageText, errorText, DIM(lpDS->szErrorMessageText));
    if (lpDS->cSuppressErrorMessage != _J('1')) {
        jdeSetGBRErrorSubText(lpBhvrCom, lpVoid, (ID) 0, _J("007FIS"), &dsDE954000);
    }
    XRCS_freeParser(hParser);
    XRCS_freeDocument(hDoc);
    XRCS_freeElement(hRootElm);
    XRCS_freeElementArray(hElm, nElmCount);
    XRCS_terminateEngine();
    jdeFree(newXMLString);
jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
return ER_ERROR;
}

/************************************************************************
* Function Clean Up
************************************************************************/
XRCS_freeParser(hParser);
XRCS_freeDocument(hDoc);
XRCS_freeElementArray(hElm,nElmCount);
XRCS_freeDocument(hBSSVDoc);
XRCS_freeElement(hRootElm);
XRCS_freeElementArray(hChildElms,nChildCount);
XRCS_terminateEngine();
jdeFree(newXMLString);
jdeFreeBSSVPayloadReturn (&bssvPayloadReturn);
#else
jdeTraceSz(NULL, _J("B5500001 - callGetFortuneBSSV() called but system does not have SOA_ENABLED defined."));
#endif

return (ER_SUCCESS);
}
• Copying and pasting from a PDF document removes any leading tabs and spaces. Therefore to replace them, select all the text (Ctrl-A or menu option Edit/Select All) and then take menu option Edit/Advanced/Format Selection or press Ctrl-K and then Ctrl-F:

• Press the **Save All** button on the Microsoft Development Design [design] screen and then close the window.
• Press **OK** on the Business Function Design screen and then press the **Build Business Function** icon to launch the BusBuild tool:

![BusBuild tool](image)

• The BusBuild screen should show 'Built successfully' at the bottom of the Build Output pane:

![BusBuild output](image)
• Close the BusBuild window and press OK on the OMW Business Function Design screen to return to the main OMW screen.

6.0 Creating the Business Service Object

Summary:
* Add a Business Function and specify the type as BSSV
* Setup the JDeveloper install location if not already done
* Invoke JDeveloper

Let’s work through the steps to create a Business Service that will consume the external web service.

We start by adding a new BSSV object to the OMW project:

• Select the OMW project BSSVTEST01 that we created in step 3.0 and click the Add button on the toolbar.
• Select the Business Function radio button on the Add EnterpriseOne Object to OMW Project screen and press OK:
• A new BSSV radio button added in Tools Release 8.97 on the Add BSFN Object screen allows us to create a business service:

![Image showing OMW Business Service Design screen with selected BSSV source language and associated fields filled in.]

• Enter the following information:

<table>
<thead>
<tr>
<th>Source Language:</th>
<th>BSSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Name:</td>
<td>J5500001</td>
</tr>
<tr>
<td>Description:</td>
<td>Get Fortune Business Service</td>
</tr>
<tr>
<td>Package Prefix:</td>
<td>oracle.e1.bssv (select this via the visual assist)</td>
</tr>
<tr>
<td>Product Code:</td>
<td>55</td>
</tr>
<tr>
<td>Product System Code:</td>
<td>55</td>
</tr>
</tbody>
</table>

Note that Business Service object names should start with a J (Java).
Also note that the Package Prefix is only visible if BSSV is selected as the Source Language.

• Press OK to add the Business Service to OMW and return to the OMW Business Service Design screen.
• Before invoking JDeveloper from the OMW Business Service Design screen, you will need to set the location of the JDeveloper install path, if this has not already been done.

• Click on the **JDeveloper Install Path** icon and enter the path to the root of the JDeveloper install, where the jdeveloper.exe file is located, eg: C:\Oracle\jdev101310, and press **OK**:

7.0 Get the WSDL Document

**Summary:**
- Obtain the WSDL document for the External Web Service

The WSDL (Web Services Description Language) document is the 'contract' that describes the service to the outside world and other systems. It provides information such as where to find the service (the **service** element defining the **endpoint**), how to communicate with it (the **binding** element), the data structures that will be passed to and from the service (the **types** element), and the functions that are available (the **interface** element).

Browse the lists of web services available on publicly available libraries of services, such as [http://www.xmethods.net/](http://www.xmethods.net/) and [http://wsfinder.jot.com/WikiHome](http://wsfinder.jot.com/WikiHome). These sites allow you to obtain the URL to the WSDL document or to download the document for use in creating the Web Service Proxy.

For this tutorial, we will use the WSDL document located at [http://www.alagad.com/WebServices/fortune/fortune.cfc?wsdl](http://www.alagad.com/WebServices/fortune/fortune.cfc?wsdl)
The Fortune web service provided by this WSDL document is a simple example that returns an amusing or interesting message based on the topic specified. It is being used with the kind permission of Alagad Inc.

8.0 Create the Web Service Proxy Client

Summary:
- Launch JDeveloper
- Add a Web Service Proxy Client (Business Tier/Web Services)
- Use the Wizard to specify the location of the WSDL document

- The Web Service Proxy Client is the code that will communicate with the external web service using the information provided by the WSDL document. The proxy client uses the SOAP message protocol.
- Launch JDeveloper by selecting the **Invoke JDeveloper** icon on the OMW Business Service Design screen:

```
Oracle JDeveloper (10.1.3.1.0)
Productivity with Choice
```

- JDeveloper start up should be initiated:
• Press **Yes** on the Save Files prompt:

```
C:\e812\QA5DV812\Java\source\oracle\e1\bssv\J5500001\J5500001.jpr
has been modified. Save changes?
```

• Once JDeveloper has started, we next need to create a Web Service Proxy client. JDeveloper creates this automatically for us when we provide it with the WSDL document for the web service.

• Expand the Application node (same name as your E1 pathcode, eg: DV812) by clicking on the plus sign to the left of it, and the Project node (BSSV Object name = J5500001) that JDeveloper has created.

• Right click on the Project node and choose **New...**
• Change the Filter By to All Technologies. Expand the Business Tier, select Web Services and pick **Web Service Proxy** from the list:

![Web Service Proxy Wizard](image)

• The Web Service Proxy Wizard is displayed:

![Create Web Service Proxy - Welcome](image)
• Click **Next** to display step 1: Web Service Description.

• Enter the URL link to the WSDL document (http://www.alagad.com/WebServices/fortune/fortune.cfc?wsdl) and click **Next**:

![Create Web Service Proxy - Step 1 of 7: Web Service Description](image)

- **WSDL Document URL:**
  
  **http://www.alagad.com/WebServices/fortune/fortune.cfc?wsdl**

- **Mapping File:**
  
  **Browse…**

- **Copy WSDL Into Project:**
  
  **Browse…**

• In step 2: Port Endpoints, select **Run against a service deployed to an external server** radio button:

![Create Web Service Proxy - Step 2 of 6: Port Endpoints](image)

- **Run against a service deployed to Embedded OC4J**

- **Run against a service deployed to an external server**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fortune.cfc</td>
<td><a href="http://www.alagad.com/WebServices/Fortune/fortune.cfc">http://www.alagad.com/WebServices/Fortune/fortune.cfc</a></td>
</tr>
</tbody>
</table>
• **Press Next** until you get to the **Default Mapping Options:**

![Create Web Service Proxy - Step 4 of 5: Default Mapping Options](image)

- **Package Name:** `ide.e1.bssv.35500001.proxy`
- **Root Package for Generated Types:** `bssv.35500001.proxy.types`

- **Generate Data Binding Classes**
  - **Reuse Existing Type Classes**
  - **Unwrap Wrapped Parameters**
  - **Map Headers To Parameters**

- **Help**  
  - **< Back**  
  - **Next >**  
  - **Finish**  
  - **Cancel**

• **Deselect Map Headers To Parameters** and press **Finish** to generate the proxy client:

![Generation In Progress](image)

- **Generating Proxy**
The generated proxy client code will be displayed:

```java
public class FortuneCfcClient {
    private oracle.el.bsv.J5500001.proxy.Fortune_Service _port;

    public FortuneCfcClient() throws Exception {
        ServiceFactory factory = ServiceFactory.newInstance();
        _port = (oracle.el.bsv.J5500001.proxy.Fortune_Service) factory
            .loadService(oracle.el.bsv.J5500001.proxy.Fortune_Service_Service)
        ;
    }

    // Add your own code here
}
```

At this point it is possible to test the web service by entering code in the main method to initialise and call the class that will perform the required web service. Enter the following after the "Add your own code here" comment:

```java
String topic = "computers";
String fortune = myPort.getFortune(topic, 0, 0);
System.out.println("Your fortune is " + fortune);
```
• Your code should look like this:

```java
public class FortuneCfcClient {
    private oracle.el.bsv.JSS00001.proxy.Fortune_PortType_port;

    public FortuneCfcClient() throws Exception {
        ServiceFactory factory = ServiceFactory.newInstance();
        _port = factory.loadService(oracle.el.bsv.JSS00001.proxy.Fortune_Service);}

    /*
     * @param args
     */
    public static void main(String[] args) {
        try {
            oracle.el.bsv.JSS00001.proxy.FortuneCfcClient myPort = new oracle.el.bsv.JSS00001.proxy.FortuneCfcClient;
            System.out.println("calling " + myPort.getFortune());
            // Add your own code here
            String topic = "computers";
            String fortune = myPort.getFortune(topic, 0, 0);
            System.out.println("
your fortune is " + fortune);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Ensure that the cursor is positioned within the “main” method where you entered the code and press the **Run** button (▶). The reply will be displayed in the log at the bottom of the screen:

```
C:\Oracle\jdev10110\jdev\bin\javaw.exe -client -classpath C:\e012\QASDV012\Java\classes,C:\Oracle\code\http://www.doughughes.net/MyServices/fortune/fortune.cfc
Your fortune is a doctor, an architect, and a computer scientist were arguing about whose profession was the oldest. In the course of their arguments, they got all the way back to the Garden of Eden, whereupon the doctor said, "The medical profession is clearly the oldest, because Eve was made from Adam's rib, as the story goes, and that was a simply incredible surgical feat."

The architect did not agree. He said, "But if you look at the Garden itself, in the beginning there was chaos and void, and out of that the Garden and the world were created. So God must have been an architect."

The computer scientist, who'd listened carefully to all of this, then commented, "Yes, but where do you think the chaos came from?"

Process exited with exit code 0.
```
9.0 Rename the Business Service Package

**Summary:**
* Rename references to the Business Services Package to uppercase

When JDeveloper created the proxy, it also created a package oracle.e1.bssv.J5500001. However, there may be references in the proxy code where this has been generated in lowercase.

- Select **Replace in Files** from the Search menu and enter the lowercase name `j5500001` in the Search Text and the uppercase name `J55000001` in Replace With. Ensure that **Match Case** is selected and Search Path **Active Project** and press **OK**.

- Press the **Save All** button on the toolbar and then rebuild the code by pressing the **Rebuild** button on the toolbar (or by pressing **Alt-F9**) to ensure that there are no errors.

10.0 Create Value Object Classes

**Summary:**
* Use the wizard to create a Value Object to pass data to the Business Service Class

We will be creating a new Business Service class in the next step that will call the Web Service Proxy. A Value Object class is used to pass the data from the Business Function to the Business Service class. In effect, it defines the parameters to the Business Service.
Right click on the Project node (J5500001) and select **New...**
• Select category **EnterpriseOne/Classes**, and then select **Business Function Value Object Class**:

The Create EnterpriseOne Business Function Value Object wizard is displayed. Press **Next** on the first page:
• Log into EnterpriseOne if prompted.
• The next page allows you to select the Business Function that we created in step 5.0. Enter B5500001 in the Object Name and press **Find**. Select the grid row and press **Next**.

![Create EnterpriseOne Business Function Value Object](image)

• The list of Business Function parameters is displayed on the next page. Select all the parameters and press **Next**.

![Create EnterpriseOne Business Function Value Object](image)
- On the final page, enter:

<table>
<thead>
<tr>
<th>Value Object Name:</th>
<th>GetFortuneVO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope:</td>
<td>Internal</td>
</tr>
</tbody>
</table>

- Note that by convention class names should start with an **uppercase** character:

![](image-url)

**EnterpriseOne Java Class**
Enter the details of your new Value Object Class.
Choose 'Publish' option if you want the object to be exposed to clients.

Value Object Name:
GetFortuneVO

Package:
oracle.e1.bssv.JS500001.valueobject

Scope
- Publish
- Internal

Field names are derived from ESFN Data Structure name i.e. mnAddressNumber.
Data types used include MathNumeric, String and Date.
• Press **Finish** to generate the Value Object Class source:

```java
package oracle.e1.bssv.J5500001.valueobject;

/**
 * Value Object Class
 * 
 * TDO: Description using Glossary Text from EnterpriseOne if appropriate.
 * 
 * Alpha Spec. Data 30
 * 
 * TDO: Description using Glossary Text from EnterpriseOne if appropriate.
 * 
 * EnterpriseOne Key Field: false
 * 
 * EnterpriseOne Alias: 30A
 * 
 * EnterpriseOne Field length: 30
 * 
 */

private String ssTopic = null;

/**
 * Version Description
 * 
 * TDO: Description using Glossary Text from EnterpriseOne if appropriate.
 * 
 * EnterpriseOne Key Field: false
 * 
 * EnterpriseOne Alias: DESG2000
 * 
 * EnterpriseOne Field length: 2000
 * 
 */

private String ssFortune = null;

/**
 * Suppress Error Message
 * 
 * TDO: Description using Glossary Text from EnterpriseOne if appropriate.
 * 
 * 
 */
```
• The fields in the Value Object Class cannot be accessed directly as they are private strings. The Java convention for accessing these fields is to use public methods, commonly called ‘getters’ and ‘setters’. JDeveloper allows us to create these automatically. Right click in the white space of the value object class and select **Generate Accessors**.

```java
package oracle.el.bssv.J5500001.valueobject;

import ...

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

public class GetFortuneVO extends ValueObject implements Serializable {

/**
 * Alpha Spec. Data 30
 * <p>
 * TODO: Description using Glossary Text from EnterpriseOne if appropriate.
 * <p>
 * TODO: EnterpriseOne Key Field: false <br>
 * EnterpriseOne Alias: AA30 <br>
 * EnterpriseOne field length: 30 <br>
 * /

private String szTopic = null;

```
On the Generate Accessors dialog, select the checkboxes for all fields and press OK.
The ‘Getter’ and ‘Setter’ methods are generated at the bottom of the Value Object Class:

```java
private String cSuppressErrorMessage = null;

//
// Data Item
//
/**
 * TODO: Description using Glossary Text from EnterpriseOne if appropriate.
 * 
 */
/**
 * TODO: EnterpriseOne Key Field: false<br>
 * EnterpriseOne Alias: DTAI<br>
 * EnterpriseOne Field length: 10 char<br>
 */
private String szErrorMessageId = null;

public void setSzTopic(String szTopic) {
    this.szTopic = szTopic;
}

public String getSzTopic() {
    return szTopic;
}

public void setSzFortune(String szFortune) {
    this.szFortune = szFortune;
}

public String getSzFortune() {
    return szFortune;
}

public void setCSuppressErrorMessage(String cSuppressErrorMessage) {
    this.cSuppressErrorMessage = cSuppressErrorMessage;
}

public String getCSuppressErrorMessage() {

}
```

- Press the **Save All** button on the toolbar.
11.0 Create Business Service Class

Summary:
- Use the wizard to create the Business Service Class
- Complete the coding for the class to call the Web Service Proxy and handle any exceptions

- Right click on the Project node (J55000001) and select New…
Select category **EnterpriseOne/Classes** and item **Business Service Class** and press **OK**:

![New Gallery Window]

Enter the following information on the Create EnterpriseOne BSSV Class dialog and press **OK**:

<table>
<thead>
<tr>
<th>Name</th>
<th>FortuneProcessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method Name</td>
<td>getFortune</td>
</tr>
<tr>
<td>Input Class</td>
<td>oracle.e1.bssv.J5500001.valueobject.GetFortuneVO</td>
</tr>
</tbody>
</table>
• Note that the Method Name should by convention start with a **lowercase** character:

![Create EnterpriseOne BSSV Class dialog box](image)

- **Name**: FortuneProcessor
- **Package**: oracle.e1.bssv.J5500001
- **Extends**: BusinessService

**Define Method**

- **Method Name**: getFortune
- **Input Class**: `e.e1.bssv.J5500001.valueobject.GetFortuneVO` [Browse]
- **Return Class**: `oracle.e1.bssvFoundation.util.E1MessageList`
The generated Java code will be displayed:

```java
package oracle.el.bssv.J5500001;

import ...;

/**
 * TDS0: Java Doc for BusinessService
 */
public abstract class FortuneProcessor extends BusinessService {
    static
    // TDS0: Create Java doc for getFortune
    // Method getFortune is used for...
    // New XML context conditionally provides the connection for the database operation and logging information
    // New context can either be an explicit connection or null. If null the default connection is used.
    // New JVM represents data that is passed to EnterpriseOne for processing. TDS0: complete javadoc for 
    // Return an E1Message containing the text of any errors or warnings that have occurred
    public static E1MessageList getFortune(IContext context, IConnection connection, GetFortuneVO internalVO) {

        // Call start internal method, passing the context (which was passed from PublishedBusinessService).
        startInternalMethod(context, "getFortune", internalVO);
        E1MessageList messages = new E1MessageList();

        // TDS0: call method [created by the wizard], which then executes Business Function or Database operation.
        messages = addMethod.returnMessages();

        // Call finish internal method passing context.
        finishInternalMethod(context, "getFortune");

        return messages;
    }

    Source Design History 4
    
    Position the cursor on the line after "import ...;" and copy/paste the following text:

    import oracle.el.bssvfoundation.util.SoftCodingRecordAccess;
    import org.w3c.dom.Element;
    import javax.xml.rpc.Stub;
    import oracle.e1.bssv.J5500001.proxy.FortuneCfcClient;
    import oracle.e1.bssvfoundation.exception.InvalidSoftCodingRecordException;
    import oracle.el.bssvfoundation.util.E1Message;
```
Expand the import statements block by pressing the + sign next to "import ...":

```java
package oracle.e1.bssv.J5500001;

import oracle.e1.bssv.J5500001.wo.valueobject.GetFortunesVO;
import oracle.e1.bssvfoundation.base.BusinessService;
import oracle.e1.bssvfoundation.base.IContext;
import oracle.e1.bssvfoundation.connection.IConnection;
import oracle.e1.bssvfoundation.util.EMessageList;
import oracle.e1.bssvfoundation.util.SOFTCodingRecordAccess;
import org.w3c.dom.Element;
import javax.xml.rpc.Stub;

import oracle.e1.bssv.J5500001.proxy.FortuneCfcClient;
import oracle.e1.bssvfoundation.exception.InvalidSOFTCodingRecordException;
import oracle.e1.bssvfoundation.util.EMessage;

/**
 * TODO: Java Doc for BusinessService
 */

class FortuneProcessor extends BusinessService {

    /**
     * TODO: Create Java Doc for: getFortune
     * @param context The connection for the database operation and loggin
     * @param context Can either be an explicit connection or null. If null the default connection will be used.
     * @param internalVO represents data that is passed to EnterpriseOne for processing. TODO: complete.
     * @return an EMessage containing the text of any errors or warnings that may have occurred.
     */

default EMessageList getFortune(IContext context, IConnection connection, GetFortunesVO internalVO) {

    // Call start internal method, passing the context (which was passed from PublishedBusinessService)
    // Create new message list for BusinessService processing.
    EMessageList messages = new EMessageList();
}
```
Delete the two comments shown below that start //TODO by selecting the text and pressing Delete:

```java
public abstract class FortuneProcessor extends BusinessService {  
    /**
     * //TODO: Create Java Doc for getFortune
     * Method getFortune is used for...
     */
    //Http connection conditionally provides the connection for the database operation and logging information
    //Http connection can either be an expicit connection or null. It will use the default connection if used.
    //HttpConnection internalVO represents data that is passed to EnterpriseEne for processing //TODO: complete javadoc for
    //Return on ELMessage containing the text of any errors or warnings that may have occurred
    public static ELMessageList getFortune(IContext context, IConnection connection, GetFortuneVO internalVO) {
        //Call start internal method, passing the context (which was passed from PublishedBusinessService).
        startInternalMethod(context, "getFortune", internalVO);
        //Create new message list for BusinessService processing.
        ELMessageList messages = new ELMessageList();
        //TODO: call method [created by the wizard], which then executes Business Function or Database operation.
        //TODO: add messages returned from EL processing to BusinessService message list.
        //Call finish internal method passing context.
        finishInternalMethod(context, "getFortune");
        //Call finish internal method passing context.
        return messages;
    }
}
```
String softCodingKey = "E1_J5500001";

    // Fetch the Suppress Error Messages flag from the value object
    boolean suppressErrorMessages = false;
    if(internalVO.getCSuppressErrorMessage().equals(\"1\") {    
        suppressErrorMessages = true;
    }

    try {
        // Create a new proxy client
        FortuneCfcClient myPort = new FortuneCfcClient();

        // Fetch the Softcoding Record which contains the override web service endpoint URL
        System.out.println("Fetching SoftcodingRecord using key " + softCodingKey);
        Element softCodingRecord = SoftCodingRecordAccess.getSoftCodingRecord(context, softCodingKey);

        // Override the endpoint contained in the proxy client
        if (softCodingRecord != null) {
            ((Stub)myPort.getPort())._setProperty(oracle.webservices.ClientConstants.CLIENT_CONFIG, softCodingRecord);
        } else {    
            // If not found, then throw an exception
            throw new InvalidSoftCodingRecordException("Unable to find Softcoding Record");
        }

        // Get the parameters from the value object
        String topic = new String(internalVO.getSzTopic());

        // Output web service call info to logs
        System.out.println("Endpoint: " +myPort.getEndpoint());

        String wsParameters = "Web Service Parameter: " + topic;
        context.getBSSVLogger().app(context, wsParameters, null, null, null);
        System.out.println(wsParameters);

        // Now call the Web Service
        String fortune = myPort.getFortune(topic, (double)0, (double)0);

        // Return the result to the calling function
        internalVO.setSzFortune(fortune);

        // And output to the log
        System.out.println("\nYour fortune is:\n" + fortune);
    } catch (InvalidSoftCodingRecordException e) {    
        //Build text substitution elements for error message separated by pipe character
        String substitutionText = e.getStackTrace()[0].getClassName() + "." + e.getStackTrace()[0].getMethodName() 
            + "\" + e.getMessage() + "\" + softCodingKey;

        //Create new E1 Message for invalid soft coding record exception.
        E1Message msg = new E1Message(context, "004FIS", substitutionText);

        //Add messages to final message list to be returned.
        if (!suppressErrorMessages) {
            //...
messages.addMessage(msg);

//Send message to the logs and return error id and text to calling function
context.getBSSVLogger().app(context, msg.getCallObjError().getGlossaryText(), null, null, e);
System.out.println(msg.getCallObjError().getGlossaryText());

internalVO.setSzErrorMessageId(msg.getCallObjError().getDDItem());
internalVO.setSzErrorMessageText(msg.getCallObjError().getAlphaDescription());

} catch (Exception e) {

//Build text substitution elements for error message separated by pipe character
String substitutionText = e.getStackTrace()[0].getClassName() + "." + e.getStackTrace()[0].getMethodName() + "|" + e.getMessage();

//Create new E1 Message for the web service exception.
E1Message msg = new E1Message(context, "003FIS", substitutionText);

//Add messages to final message list to be returned.
if (!suppressErrorMessages) {
    messages.addMessage(msg);
}

//Send message to the logs and return error id and text to calling function
context.getBSSVLogger().app(context, msg.getCallObjError().getGlossaryText(), null, null, e);
System.out.println(msg.getCallObjError().getGlossaryText());

internalVO.setSzErrorMessageId(msg.getCallObjError().getDDItem());
internalVO.setSzErrorMessageText(msg.getCallObjError().getAlphaDescription());
}
• Select menu option **Source/Reformat** to re-insert leading tabs removed by the copy/paste from the PDF document:

• Press the **Rebuild** button on the toolbar or press **Alt-F9** to compile the class. Ensure that there are no errors in the Messages log:

• Press the **Save All** button on the toolbar.
12.0 Configure Soft Coding Record

**Summary:**  
Configure the Soft Coding Record in EnterpriseOne

- The Soft Coding Record is used to hold information such as the web service endpoint (i.e., the URL to where the web service can be found) and security information. By having these values soft-coded, provides the possibility of having different sets of values for different environments, or for user/role based variations in web service credentials.

- We use a standard EnterpriseOne application to enter the Soft Coding Records (P954000). Launch the EnterpriseOne menu by selecting the **Tools/EnterpriseOne** menu option from the Solutions Explorer menu:

  ![EnterpriseOne Menu](image)

- From the EnterpriseOne web-client menu, enter **P954000** in the Fastpath and press the **Enter** key:

  ![Fastpath](image)
• After the Work with Web Service Soft Coding Records application launches, press the **Add** button:

![Work with Web Service Soft Coding Records - Microsoft Internet Explorer provided](image)

**Work with Web Service Soft Coding Records**

<table>
<thead>
<tr>
<th>Select</th>
<th>Find</th>
<th>Add</th>
<th>Copy</th>
<th>Delete</th>
<th>Close</th>
<th>Tools</th>
</tr>
</thead>
</table>

No records found.

<table>
<thead>
<tr>
<th>User / Role</th>
<th>Environment</th>
<th>Soft Coding Key</th>
<th>Soft Coding Description</th>
</tr>
</thead>
</table>

• Enter the following formation on the Add Web Service Soft Coding Record form:

```
User/Role: *PUBLIC
Environment Name: Your E1 environment name, eg: DV811
Soft Coding Key: E1_J5500001
Soft Coding Description: Get Fortune Business Service Endpoint
Soft Coding Value:

<port-info>
  <stub-property>
    <name>javax.xml.rpc.service.endpoint.address</name>
    <value>http://www.alagad.com/WebServices/fortune/fortune.cfc</value>
  </stub-property>
</port-info>
```
• Note that as we are currently testing locally, the environment name should be the non-J environment name, eg:

```
*PUBLIC
DV812
E1_J5500031
```

• However, when the Business Function and Business Service are called from applications running on the EnterpriseOne Web Server, the system will be looking for the J environment. Therefore, a second record would normally be required to specify the Soft Coding Record for that environment. For the purpose of this tutorial, it is not needed.

• Note that the Soft Coding Value here does not contain any logon credentials, since this is not required for the web service that we are calling. Further information about coding logon credentials in the Soft Coding Value can be found in the JD Edwards EnterpriseOne Tools 8.98 Business Services Development Guide.
13.0 Test the Business Service Class

Summary:
* Create an XML document that can be passed to the Business Service
* Create a test class to call the Business Service
* Run the Test Class

To test the Business Service, we need to generate the Business Service payload that would normally be sent through from the Business Function in the form of an XML document. We can generate this automatically based on our value object.

- Expand the following folders under the Project node (J5500001) in JDeveloper:
  - Application Sources
    - oracle.e1.bssv
    - J5500001
    - valueobject

- Right click on the `GetFortuneVO.java` value object and select **Generate XML Document Template**.
The XML source is displayed:

```
<get-fortune-VO encoding="UTF-8">
  <sz-error-message-text></sz-error-message-text>
  <sz-fortune></sz-fortune>
</get-fortune-VO>
```

Enter the topic name **people** between `<sz-topic>` and `</sz-topic>`:

```
<sz-error-message-id></sz-error-message-id>
<sz-topic>people</sz-topic>
<CSsuppressErrorMessage></CSsuppressErrorMessage>
```

Press **Save All** on the toolbar.

We now need to create a test class that will call the Business Service, passing this XML document. Right click on the Project node (J5500001). NB: select the Project node (under the Application/pathcode node) and not the J5500001 package which is lower down the hierarchy. Select **New...**
• Select category **General/Simple Files** and item **Java Class**:

On the Create Java Class dialog, enter **TestGetFortuneBSSV** as the class name. Deselect **Generate Default Constructor** and select **Generate Main Method**, and press **OK**.
• The simple Java class is created:

```java
package oracle.e1.bssv.J5500001;

public class TestGetFortuneBSSV {
    public static void main(String[] args) {
        TestGetFortuneBSSV testGetFortuneBSSV = new TestGetFortuneBSSV();
    }
}
```

• Copy/Paste the following code into the main method, after the line “TestGetFortuneBSSV testGetFortuneBSSV = new TestGetFortuneBSSV();”, but before the ”}” on the next line:

```java
try {
    //call required prior to executing test from application (main())
    TestBusinessService.startTest();

    String file = "C:\\e812\\source\\oracle\\e1\\bssv\\J5500001\\valueobject\\GetFortuneVO.xml";

        "getFortune",
        file,
        IConnection.AUTO);

} finally {
    //call required at the end of testing from application (main())
    TestBusinessService.finishTest();
}
```
Select **Reformat** from the **Source** menu:

- Replace `<pathcode>` in the XML file path with the pathcode for your E1 environment, eg: DV812. Your code should now look like:
We now need to add the import statements. JDeveloper can do this automatically for us. You should see prompts, such as `import oracle.EnterpriseOnebssvfoundation.base.TestBusinessService` (Alt-Enter). Press Alt-Enter to add this import statement into the code.

Press Alt-Enter again to add `import oracle.e1.bssvfoundation.connection.IConnection`.

Press the Rebuild button on the toolbar or Alt-F9 to compile the classes in the project. Ensure that there are no errors.

Press Save All to save the code.

To test the Business Service, select anywhere in the main method and press the Run button on the toolbar or press F11. Any messages will appear in the Running log at the bottom of the screen:

```
C:\Oracle\jdev101310\jdk\bin\javaw.exe -client -classpath C:\e812\QASDV812\Java\classes
Fetching SoftcodingRecord using key El_J5500001
Endpoint: http://www.doughughes.net/WebServices/fortune/fortune.cfc
Topic: people

Your fortune is:
It takes less time to do a thing right than it does to explain why you did it wrong.

        -- H.W. Longfellow
Process exited with exit code 0.
```
14.0 Deploy the Development Business Services Server

**Summary:**
* Create a web archive file and deploy this as an application to your local development web client application server

- Up to this point we have only tested the Business Service and the call to the web service from within JDeveloper. We still need to call the Business Service from the new EnterpriseOne Business Function that we created in step 5.0. However, to do this we need to deploy the Business Service to our local web server (Oracle's OC4J or Websphere Express).
- In JDeveloper, right click on the Application node (E1 pathcode name) and select **Deploy Development BSSV Server**:

```
public abstract class ...
/**
 * TODO: Create Java Method getFortun
 * @param context ... ...
 * @param connection ...
 * @param internal ...
 * @return an EIMessageList ...
 */
public static EIMessageList ...

// Call start it
startInternalMe ...
// Create new me
EIMessageList ...

String sopCode;

try {
    // Create 
    FortuneCfco
```
• On the Deploy Development Business Services Server screen, if your organisation uses an HTTP Proxy Server for routing web traffic, then enter the following information and press **OK**:

<table>
<thead>
<tr>
<th>Use HTTP Proxy Server:</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name:</td>
<td>Your organisation’s HTTP proxy server name</td>
</tr>
<tr>
<td>Port Number:</td>
<td>Your organisation’s HTTP proxy server port</td>
</tr>
<tr>
<td>Exceptions:</td>
<td><strong>localhost</strong></td>
</tr>
</tbody>
</table>

**Deploy Development BSSV Server**

The Development Business Services Server is about to be deployed to the OC4J application server at this location: `C:\e812\system\OC4J`.

Specify proxy server settings that will be used when you access web services.

- **Use HTTP Proxy Server**
  - Host Name: `httpserver`
  - Port Number: `3080`
  - Exceptions: `localhost`

  Use a vertical bar (`|`) to separate exception entries. You can use an asterisk (`*`) as a wildcard.

- **Proxy Server Requires Authentication**
  - User Name: 
  - Password: 

- **Use HTTPS Proxy Server**
  - Host Name: 

[Collapse/Expand]

[OK] [Cancel]
• If your organisation does not use an HTTP proxy server for routing web traffic, then leave the **Use HTTP Proxy Server** checkbox as unticked and do not enter values into the **Host Name**, **Port Number** or **Exceptions** and press **OK**:

![Deploy Development Business Services Server](image)

**Deploy Development BSSV Server**

The Development Business Services Server is about to be deployed to the OC4J application server at this location: C:/e812/system/OC4J.

Specify proxy server settings that will be used when you access web services.

- **Use HTTP Proxy Server**
  - **Host Name**: 
  - **Port Number**: 
  - **Exceptions**: Use a vertical bar (|) to separate exception entries. You can use an asterisk (*) as a wildcard.
  - **Proxy Server Requires Authentication**
    - **User Name**: 
    - **Password**: 

- **Use HTTPS Proxy Server**
  - **Host Name**: 

[OK]  [Cancel]

• If the local web server is already started and the proxy server settings have changed, then you may receive a warning message that it will be restarted and any applications will be closed. Press **Yes** to indicate that you would like to continue:

![Restart Required](image)

**Updating http proxy settings requires restarting the development web server.**

All unsaved E1 transactions should be saved prior to continuing. Would you like to continue?

[Yes]  [No]
• The build messages will appear in the Apache Ant log. You should see ‘BUILD SUCCESSFUL’ at the bottom:

![Ant Build Log]

• The deployment of the Development Business Services Server may take several minutes, particularly the first time that it is deployed, and especially if WebSphere Express is being used as your development Application Server.

• **Warning**: If Websphere Express is used and you do not have a proxy server, then do not specify values on the Deploy Development BSSV Server screen. If you do, then the proxy server settings will be written to Websphere Express and the Apache Ant deployment process does not allow them to be removed. Removal will require logging on to the Websphere Express Administrative Console for the DevBSSVSvr profile (Start/Programs/IBM Websphere/Application Server - Express v6/Profiles/DevBSSVSvr/Administrative console) and then navigating to Servers/Application Servers/server1/Server Infrastructure/Process Definition/Java Virtual Machine/Custom Properties, and deleting the entries for the HTTP Proxy Server.

15.0 Updating startOC4J.bat

**Summary:**

🌟 Cater for no proxy server when starting the OC4J local web server

**Only required if the Oracle OC4J web application server is being used, and if your organisation does not use a proxy server.**

• When the local OC4J web server is started, it uses the HTTP Proxy Server settings that were written to the jde.ini file when the Development Business Services Server was deployed (step 14.0). However, these values will be blank if your organisation does not use a proxy server. The OC4J instance is started using these blank values, and an error will be received when the Business Service tries to call out to the external web service. A change is required to the batch file that starts the OC4J to not use the proxy server settings.
• Use a text editor (eg: Notepad) to modify the `startOC4J.bat` file located at `C:\e812\system\OC4J`.

```bat
pushd %~dp0
call setenv.bat
popd
pushd %~dp0\jee\home
set PATH=%~dp0..\JDK\bin;%PATH%
start "OC4J H4A" /B "..\..\JDK\bin\java" "-Dhttp.proxyHost=%1" "-Dhttp.proxyPort=%2" "-Dhttp.proxySet=%3" "-Dhttp.nonProxyHosts=%4" "-Xbootclasspath/p:%CP%" "-jar" "oc4j.jar"
```

• Comment out the following line by inserting `REM` in front of it:

```bat
start "OC4J H4A" /B "..\..\JDK\bin\java" "-Dhttp.proxyHost=%1" "-Dhttp.proxyPort=%2" "-Dhttp.proxySet=%3" "-Dhttp.nonProxyHosts=%4" "-Xbootclasspath/p:%CP%" "-jar" "oc4j.jar"
```

• Insert the following line after this. Copy/paste this text into the editor on a new line after the commented out line:

```bat
start "OC4J H4A" /B "..\..\JDK\bin\java" "-Xbootclasspath/p:%CP%" "-jar" "oc4j.jar"
```
• The code should now look like:

```
pushd %~dp0
call setenv.bat
popd
pushd %~dp0\j2ee\home
set PATH=%~dp0..\jdk\bin;%PATH%
REM start "OC4J H4A" /s "..\..\..\jdk\bin\java" "-Dhttp.proxyHost=" -dhttp
start "OC4J H4A" /b "..\..\..\jdk\bin\java" "-Xbootclasspath/p:%CP%" "-jar" 
popd
```

• Save the changes.
• Exit from the EnterpriseOne Solution Explorer and log on again so that the OC4J web server is restarted with the new settings.
• These changes will need to be repeated whenever a full package or update package containing a tools release is deployed to the workstation.

### 16.0 Configure OCM Service Configuration

**Summary:**

Create the OCM Service Configuration record to point to the Development Business Services Server

• EnterpriseOne needs to know where the Business Services Server is located. There could be several Business Services Servers, eg: one for each EnterpriseOne environment. In our case we want to ensure that any calls to the Business Server from the Business Function B5500001 are sent to our Development Business Services Server. The configuration of the location of the Business Services Server is handled by the OCM Service Configuration.
On the EnterpriseOne Solution Explorer menu, expand folders System Administration Tools, System Administration Tools and Environment Management:
• Double click on the **Environment/Service Configuration** menu option to launch the Object Configuration Manager (OCM) Machine Search and Select screen:

Note that your machines and data sources will different to this list.
• Select the **System – 812** data source and press the **Select** button on the toolbar to display the Work with Service Configuration screen:

![Environment/Service Configuration](image)

• Press the **Add** button on the toolbar to display the Service Configuration Revisions screen. Enter the following information:

<table>
<thead>
<tr>
<th>Environment Name:</th>
<th>The E1 environment name that you are logged in to (eg: <strong>DV812</strong>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name:</td>
<td><strong>BSSV</strong></td>
</tr>
<tr>
<td>User/Role:</td>
<td><strong>PUBLIC</strong></td>
</tr>
<tr>
<td>Server:</td>
<td><strong>LOCALHOST</strong></td>
</tr>
<tr>
<td>Port:</td>
<td>6667</td>
</tr>
</tbody>
</table>
• The screen should look like this:

![Environment/Service Configuration screen](image1)

• Press **OK** to return to the Work with Service Configuration screen:

![Work with Service Configurations screen](image2)
• Select the row that you have just added and press row exit **Change Status**:

![Image of Environment/Service Configuration window with row selection and status change]

- The Object Status should change to ‘AV’.
- Press the Close button twice on the toolbar to exit back to the Solution Explorer menu.

### 17.0 Create a Test Interactive Application

**Summary:**

Create a test application that will call the Call Get Fortune Business Service Business Function

- We have now completed all the steps to enable the Business Function to call the Business Service on the Development Business Services Server, and for the Business Service to call the external Web Service via the Web Proxy Server. For the final step, we now need to create a test application to call the Business Function.
- From OMW, select the **BSSVTEST01** project and press the Add button on the Toolbar.
- On the Add EnterpriseOne Object to the Project screen, select the Interactive Application radio button and press OK:
Enter the following information on the Add Object screen:

<table>
<thead>
<tr>
<th>Object Name:</th>
<th>P5500001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Get Fortune Business Service Test</td>
</tr>
<tr>
<td>Product Code:</td>
<td>55</td>
</tr>
<tr>
<td>Product System Code:</td>
<td>55</td>
</tr>
<tr>
<td>Object Use:</td>
<td>1</td>
</tr>
</tbody>
</table>

Press **OK** on the toolbar:
• On the Interactive Application Design screen, click on the **Start Form Design Aid** icon:

![Start Form Design Aid icon](image1)

- **Start Form Design Aid**
- **Vocabulary Overrides**
- **Browse Event Rules**
- **View Forms**
- **Run HTML Application**
- **ER Compare**
- **Version List**

• Once the Form Design Aid has loaded, select **Form, Create, Fix/Inspect** menu option:

![Form Design Aid menu](image2)
• Enter the Form Title **Get Fortune Business Service Test** and select the **Entry Point** checkbox:

![Fix Inspect Form Properties dialog box](Image)

- Press **OK** and a blank form will be displayed:
• If the Data Dictionary Browser highlighted above is not visible, then select View/Data Dictionary Browser menu option:

• Enter **AA30** in the Alias column’s Query By Example (QBE) field on the Data Dictionary Browser and press **Enter**:
• Select the grid row and drag it onto the new form:

• Release the mouse button, position the box on the form, and then click the left mouse button to add the form control to the form:
On the Application Tree View in the top left corner of the screen, click on the **Alpha Spec Data 30** form control. Once selected it will be in rename mode. Type in **Topic:**
• Repeat for the **Alpha Spec Data 30** label:
• Now repeat the above process for Data Dictionary item **DESC2000**:
• Rename to **Fortune**:

• Double click on the Fortune form control to display the properties:
Select the **Size/Pos** tab and enter Height **100** and Width **200**:

Select the **Advanced** tab and select the **Multiple Lines** radio button and **Auto Vertical Scrolling** radio button:
• Press **OK**:

• Click on the blank space on the form to select it (the black resizing handles around the form should be visible).
• From the **Insert** menu, select the **Push Button** menu option:
• Position the box to the right of the Topic form control:
• Press the left mouse button to position the Push Button:
• On the **Application Tree View**, select the Push Button and rename to **Get Fortune**:

• Right click on the Get Fortune Push Button and select **Event Rules**:
• On the blank Event Rules Design screen, press the **Business Function** icon:

![Image of Event Rules Design screen]

• On the Business Function Search screen, enter **B5500001** in the Source Module QBE field and press **Find**:

![Image of Business Function Search screen]
• Select the row, and press the **Select** button to display the Business Functions screen:
- Select the **FC Topic** form control in the Available Objects pane on the left and press the > button. This will copy the form control to the szTopic parameter of the Business Function call:
Now click on the <NOT Assigned> value on the szFortune parameter line:
• Select the **FC Fortune** form control and press the > button to copy to the szFortune parameter:

• Press **OK** to return to the Event Rules Design screen.
• Press the **Save** button on the toolbar to exit back to the form design:
• Press the **Save** button on the toolbar.

• **Close** the Form Design Aid window and ensure that there are no errors in the Validate Event Rules message form:

  - **Validate Event Rules**

    Get Fortune Business Service Test

    Validating Event Rules for P5500001
    Get Fortune Business Service Test
    Validation successful

    OK

• Press **OK** to exit Form Design Aid.
18.0 Run the Test Application

Summary:
- Run the Get Fortune Business Service Test application P5500001 from the EnterpriseOne menu

- From the Solution Explorer menu, select menu option Tools/EnterpriseOne Menu:
• When the EnterpriseOne menu has launched, type in **P5500001** in the Fastpath and press **Enter**.
When the application loads, enter **JOBS** in the Topic and press the **Get Fortune** button. The message returned from the Get Fortune web service should be displayed in the Fortune box:

![Get Fortune Business Service Test](image)

- Congratulations! You have now succeeded in calling an external web service from EnterpriseOne.

**19.0 Next Steps**

Once you’ve been able to test the calling of the external web service by running the Business Function and Business Service locally on your E1 development client, the next steps would be to:

- Use OMW to check in the objects you have created to the deployment server.
- Build an update package containing your objects. The ‘Business Services’ processing option on the Package Assembly application (P9601) should be set to ‘1’.
- Build the package. The Business Services will be compiled using JDeveloper into Java Classes. If deploying to WebSphere, then the Rational Application Developer (RAD) will also need to be installed.
- Deploy your package to your Enterprise Server and Business Services Server. The EnterpriseOne Server Manager is used to deploy the generated ear (Enterprise Archive) file containing the Java classes to the Business Services Server running on Oracle Application Server or Websphere.
- Configure the Web Service Soft Coding Record for the J-environment that you will log into on the Web Server to run the test application.
- Configure the OCM Service Configuration to point to the Business Services Server.
20.0 Further Information
Please refer to the following sources for further information about Business Services:

20.1 Oracle Documentation

General:
- JD Edwards EnterpriseOne Tools 8.97 Business Services Documentation Map:
  https://support.oracle.com/CSP/main/article?cmd=show&type=ATT&id=701005.1:DOCMAP

Development:
- JD Edwards EnterpriseOne Tools 8.98 Business Services Development Guide:
- JD Edwards EnterpriseOne Tools 8.98 Business Services Development Methodology Guide:
- JD Edwards EnterpriseOne Tools 8.98 Interoperability Guide:
- JD Edwards EnterpriseOne Tools 8.98 Object Management Workbench Guide:
  http://download.oracle.com/docs/cd/E13780_01/jded/acrobat/E1_TOOLS898TOM-B0908.pdf

Install and Configuration:
- Business Services Server Minimum Technical Requirements:
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=705324.1
- JD Edwards EnterpriseOne Tools Release 8.97 Business Services Server Reference Guide:
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=704213.1
- JD Edwards EnterpriseOne Tools 8.98 Security Administration Guide:
- JD Edwards EnterpriseOne Tools 8.98 Package Management Guide:

Reference Implementations:
- JD Edwards EnterpriseOne Tools 8.97 Reference Implementation Guide:
  https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=705194.1
- JD Edwards EnterpriseOne Business Services 9.0 Reference Guide:
  http://download.oracle.com/docs/cd/E13781_01/jded/acrobat/e190AFI-B0908.pdf

20.2 Transfer of Information Presentations

Functional:
- JD Edwards EnterpriseOne Business Services Functional Overview:
- JD Edwards EnterpriseOne Business Services Administration Functional Overview:
- JD Edwards EnterpriseOne Business Services Development Functional Overview:

Technical:
- JD Edwards EnterpriseOne Business Services Administration Technical Overview:
20.3 Self Study Training

- EnterpriseOne Business Service Development Rel 8.97 – RWC:

20.4 Other Sources

- JD Edwards EnterpriseOne - Fusion Middleware Best Practices Center:

- JD Edwards EnterpriseOne - Process and Integration Discussion Forum: