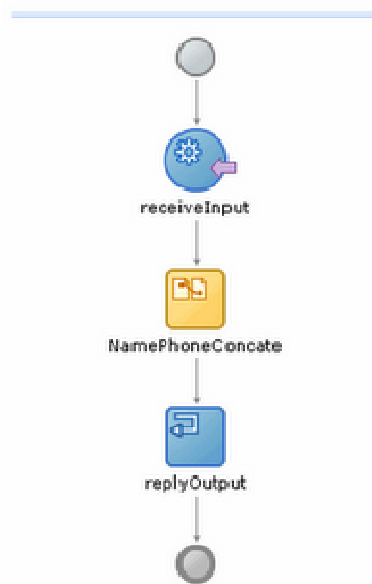


Invoking BPEL process from ADF

First create one synchronous BPEL process for this purpose which we want to call from an ADF application. We are creating this BPEL process for demonstration purpose. Later we'll call this BPEL process from ADF application.

Here our process is very simple one -> one can input his **name** and **phone** no. here and this will concat and return the name and phone no. in a string.

The diagram of this BPEL process is like this ->



The input schema of this BPEL process is following ->

```
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/BPELwithADF"
xmlns="http://www.w3.org/2001/XMLSchema">
```

And then in an assignment function we have concatd the input name and phone no. Our output is the following one ->

```
http://xmlns.oracle.com/BPELwithADF>
http://xmlns.oracle.com/BPELwithADF>name = Deba phone = 9820040024
```

In the following article we'll describe how to invoke this BPELProcess, pass the two parameters to this process and how to show the results in one ADF faces component on a JSF page.

Creating the webservice data control for calling the BPEL process

Web services is the term for a technology that consists of a set of messaging protocols and programming standards that expose **business** functions over the Internet using open XML-based standards. An individual web **service** is a discrete reusable software component that is accessed programmatically over the Internet, using HTTP/HTTPS but you can also invoke them using other protocols, such as JMS or SMTP, to return a response.

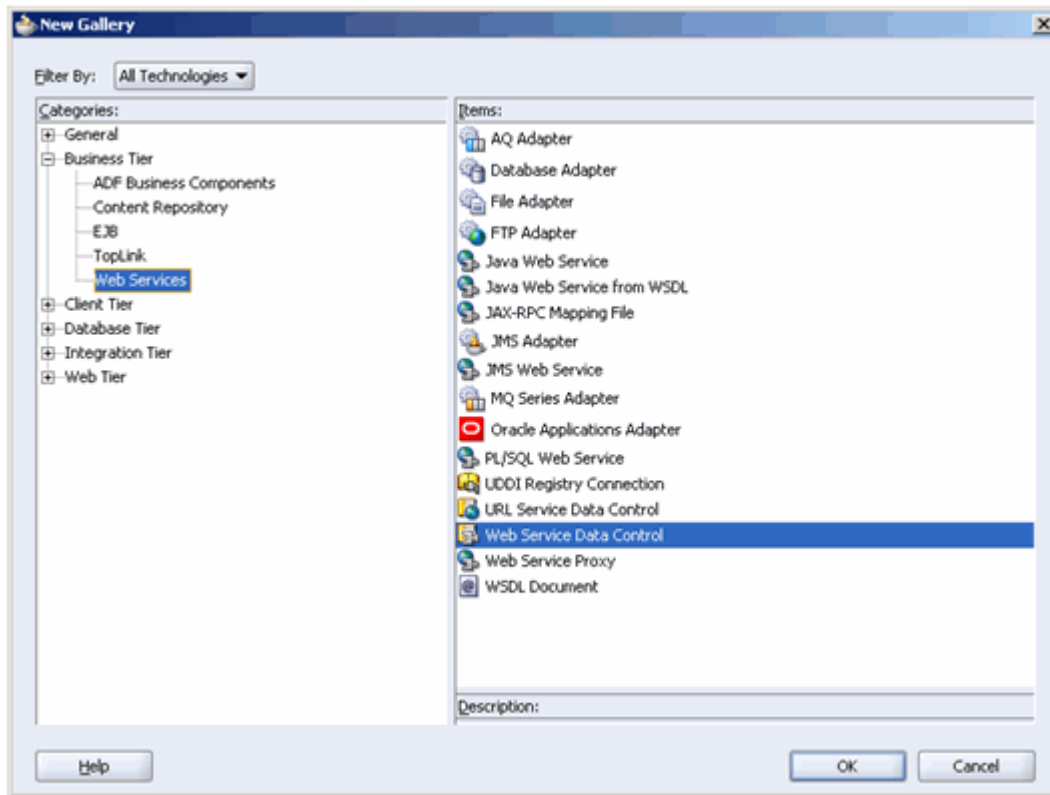
Web services are modular business services that can be easily integrated and reused, and it is this that makes them ideally suited as components within service-oriented architecture (SOA). OracleAS SOA **helps** you to create top-down and bottom-up web services or to create web services from existing functionality. You can do this using JDeveloper, alternatively you use Ant or integrate the Oracle SOA Suite with another tool.

JDeveloper provides wizards and other tools to develop web services that are J2EE 1.4-compliant, with the ability to create JAX-RPC clients and services. In addition JDeveloper automatically **generates** web service metadata tag annotations into implementation classes (for compliance with JSR-181). There is support for defining ->

- custom handlers
- serializers and deserializers
- MIME/DIME attachments
- WS-Security, security settings including SAML for developing SSO web service applications
- WS-Reliability to set quality of service
- WS-Management to set logging properties
- WSIF bindings that allow the publishing of Java, EJB, Java Message Service (JMS), and enterprise applications connected via Java Connector Architecture (JCA) adapters
- REST (Representational State Transfer)-based service

Here is a way of creating web service data control from an existing WSDL [In our case the BPEL process WSDL file].

First, In the project node of Jdeveloper click new. Select Business Tier > Web Service > Web Service Data Control.

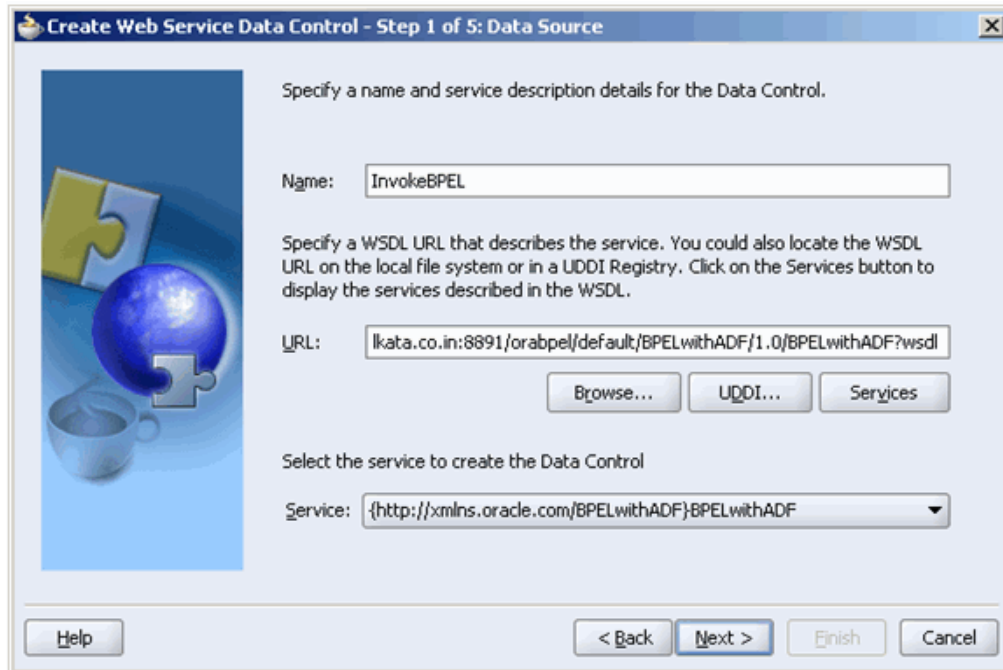


2

Click ok.

In the url field of the next window give the WSDL file location of the BPEL process. Give a suitable name in the Name field. Choose the proper service method which we are going to call.

F3



Create Web Service Data Control - Step 1 of 5: Data Source

Specify a name and service description details for the Data Control.

Name:

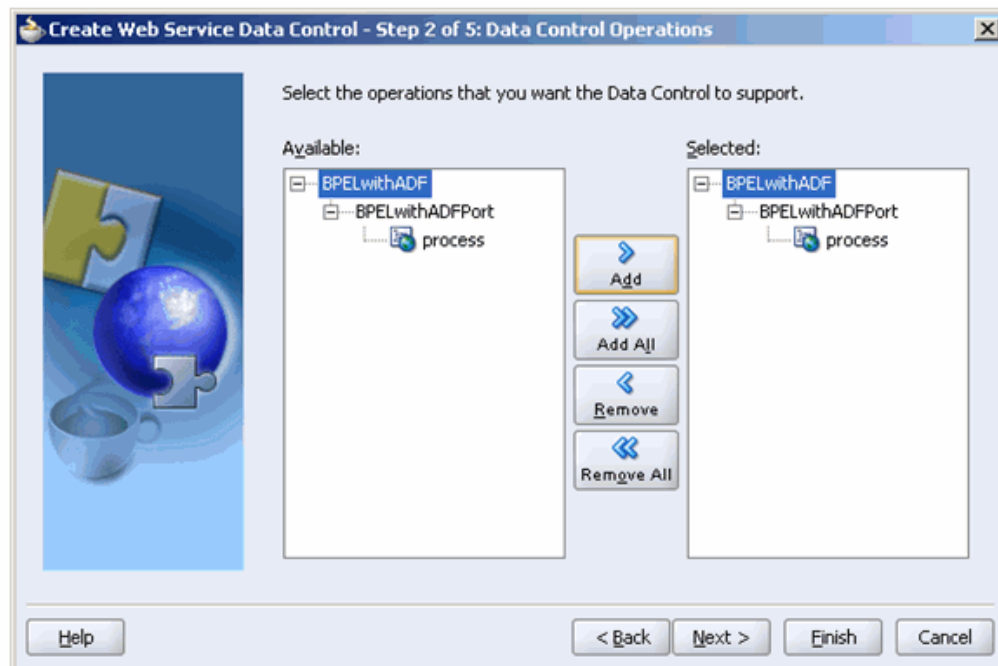
Specify a WSDL URL that describes the service. You could also locate the WSDL URL on the local file system or in a UDDI Registry. Click on the Services button to display the services described in the WSDL.

URL:

Select the service to create the Data Control

Service:

In the next screen click on add button for the processes which you want to initiate from your ADF application.



Create Web Service Data Control - Step 2 of 5: Data Control Operations

Select the operations that you want the Data Control to support.

Available:

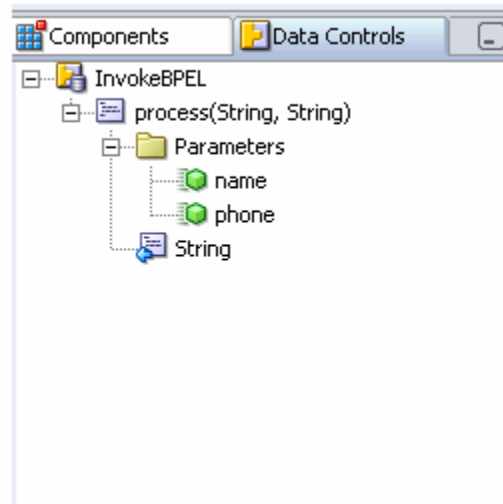
- BPELwithADF
 - BPELwithADFPort
 - process

Selected:

- BPELwithADF
 - BPELwithADFPort
 - process

Click on next two times. In the end point authentication page give the proper username and password if the web service is a secured one. Click on Finish in the next page.

You can see your web service with its method has become a component in the data control pallet.



F5

There is one file InvokeBPEL.xml which holds all the Input/Output parameter signature like the following way ->

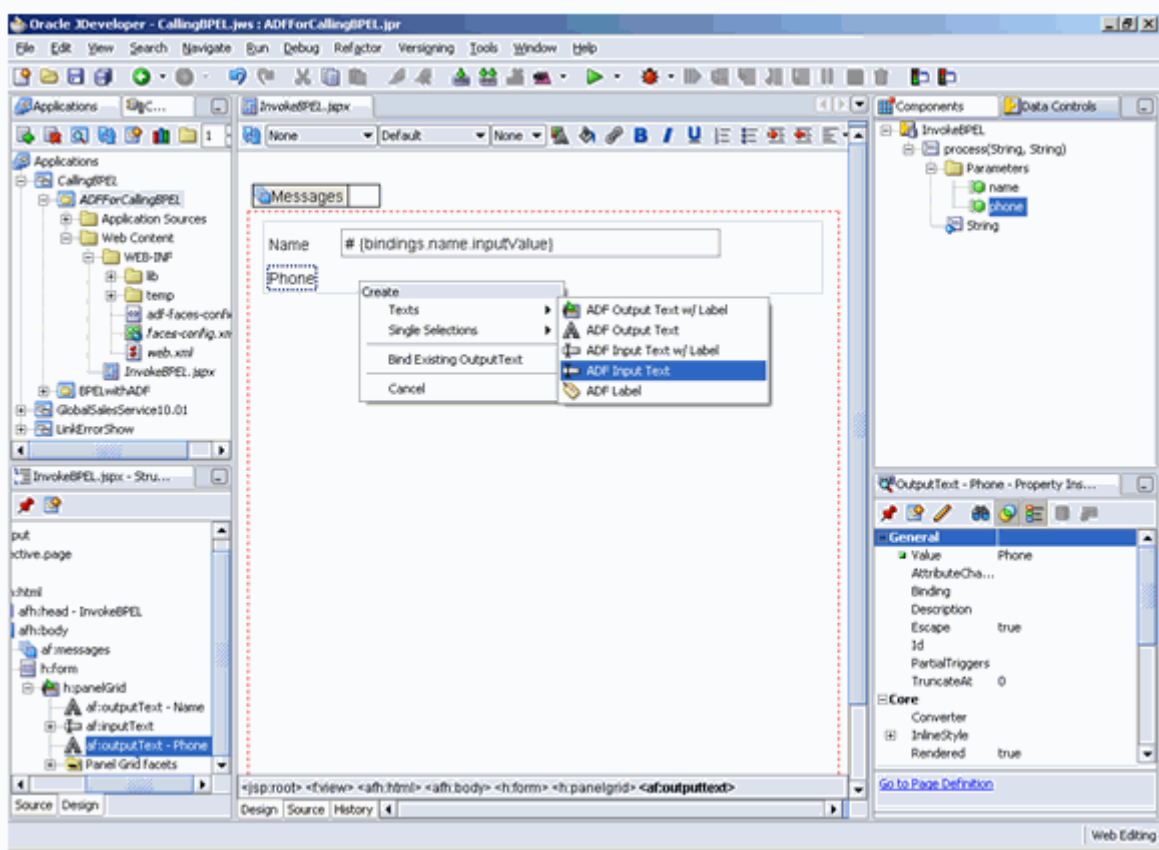
```
id="InvokeBPEL" BeanClass="InvokeBPEL" isJavaBased="false">
```

```
ReturnNodeName="Return">
```

There is another file "**DataControls.dcx**" which holds other information about the webservice specially the WSDL location.

Integrating Web Service Data Control in JSF pages with ADF faces component

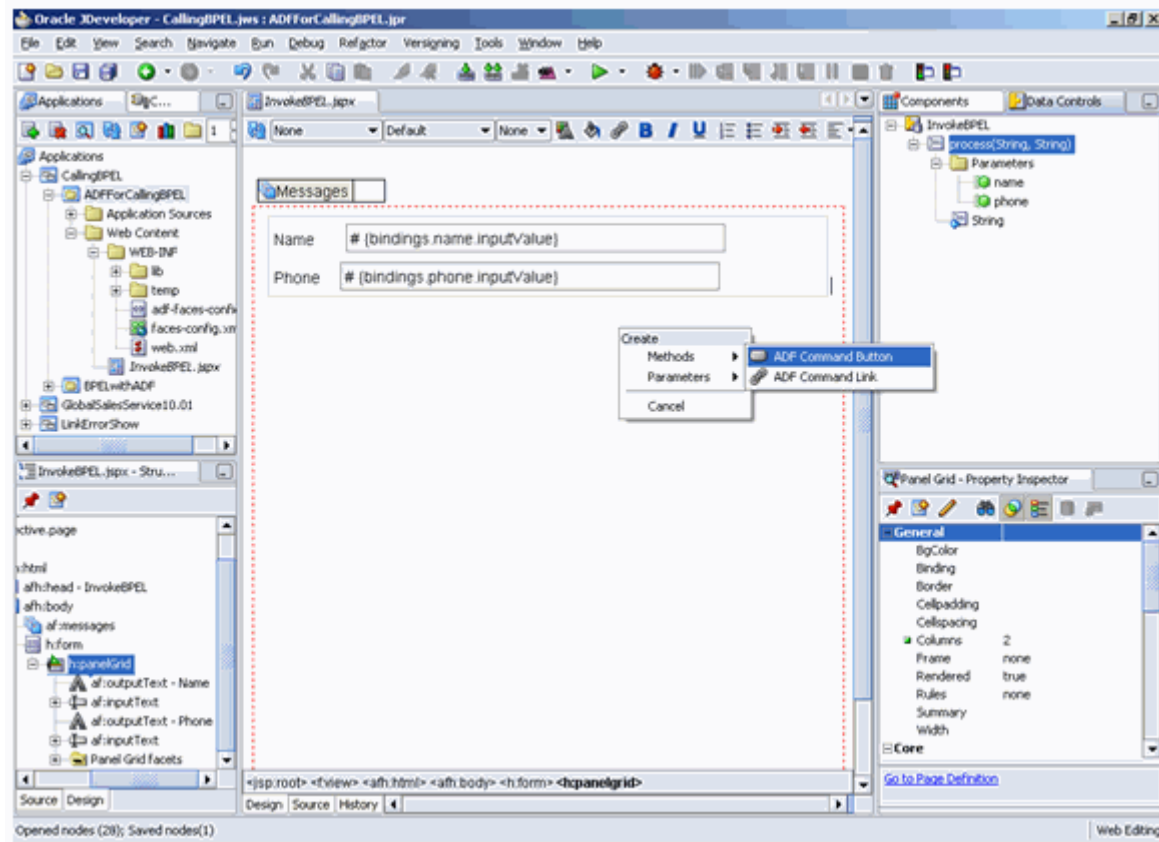
Create one JSF/JSP page like InvokeBPEL.jspx in JDeveloper. Now just drag and drop the input parameters of the BPEL process on the JSF page ->



F6

In our case it is Name and Phone no. which we can attach in the JSF page.

Drag and drop process on the page and select methods > ADF Command Button.

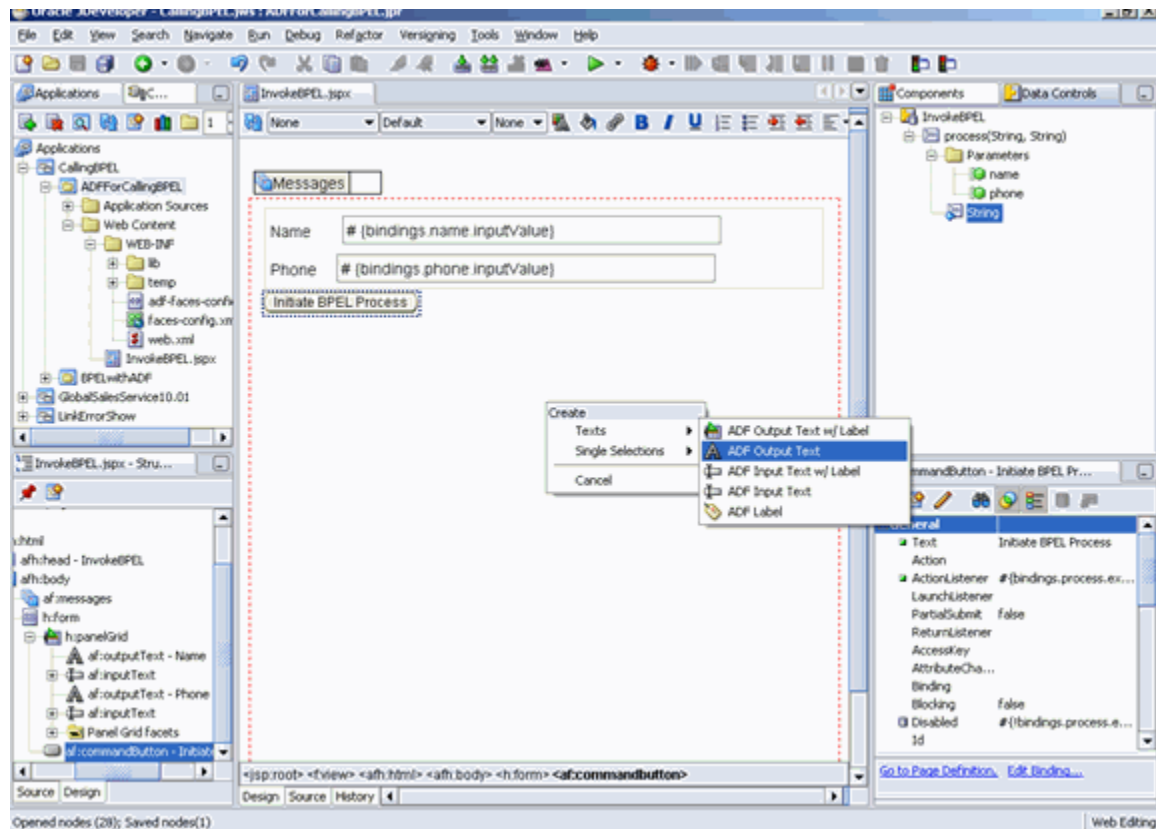


F7

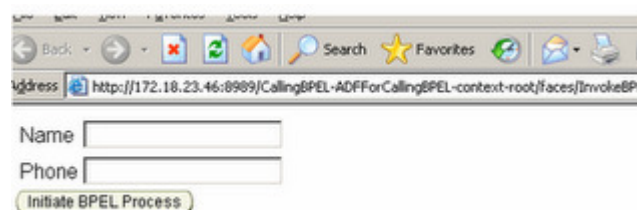
Change the name to “Initiate BPEL Process”.

Now, drag and drop the Return variable from InvokeBPEL webservice data control.
Select Texts > ADF Output Text.

F8

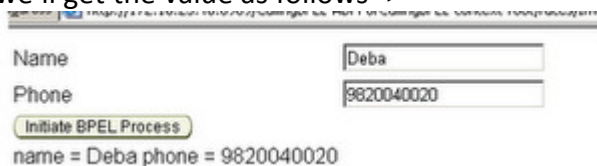


Now if you run this page, the page will be displayed as follows ->



F9

After running the page we'll get the value as follows ->



F10

With ADF application is getting the Name and Phone value from user input, and then it is invoking the BPEL process. In the input parameter of the BPEL process it is passing the name

and phone no. as Input parameter. When the BPEL process will end it sends the result to ADF application and ADF is displaying the end result in one ADF faces outputText component.