SAMPLE NAME : bpel-101-HelloWorld

COMPONENT : BPEL

OVERVIEW

This sample contains instructions for how to create a simple HelloWorld BPEL process and test it. Along the way, you will also learn how to create an application server connection between JDeveloper and WebLogic server. This connection will allow you to easily deploy your processes from inside of the JDeveloper environment.

PROVIDED FILES

ReadMe.txt – A basic readme file that covers the general aspects of the sample.
Sample_Instructions.pdf – The file you are now reading.

STEP BY STEP INSTRUCTIONS

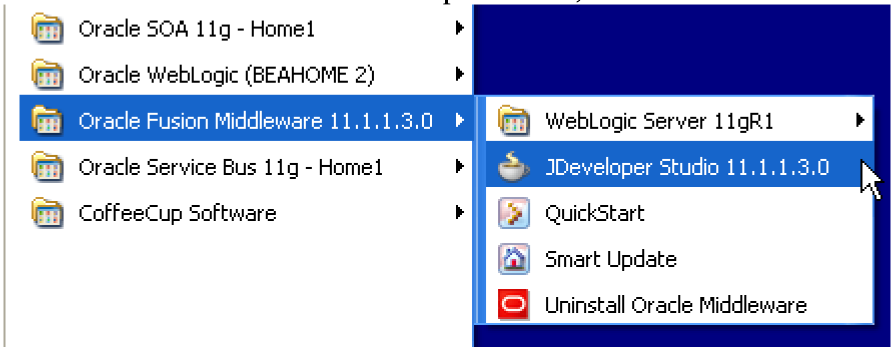
Prerequisites

Before you can use this sample you must install the Oracle SOA Suite 11g product suite. This sample was developed using the PS2 (Patch Set 2) release of Oracle SOA Suite 11g. To install Oracle SOA Suite 11g PS2 you will need to follow the detailed instructions at http://download.oracle.com/docs/cd/E14571_01/doc.1111/e13925/toc.htm

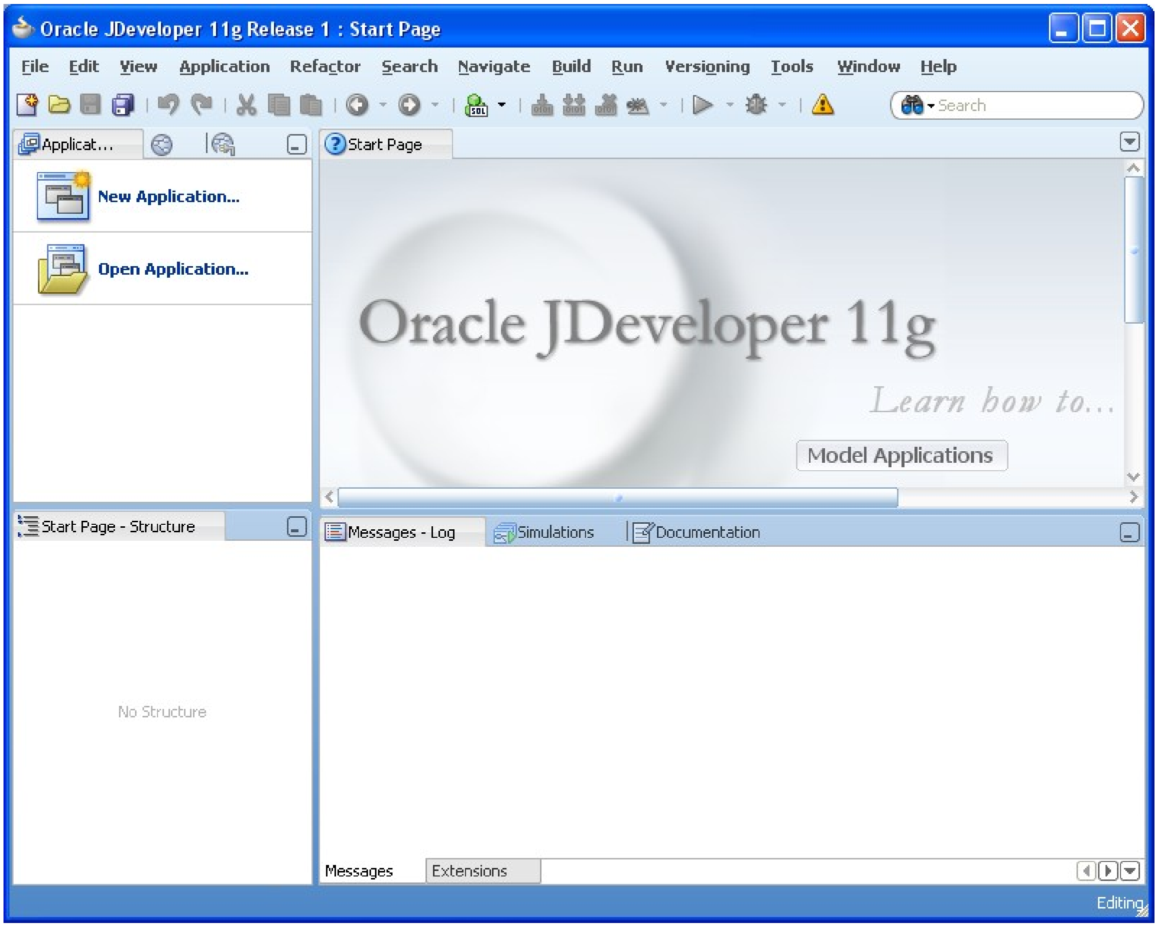
You must have all of the components installed before you can proceed with this sample. The screenshots and direction in this example use Microsoft Windows as the operating system. If you are using a Unix/Linux operating system, you will need to translate some of the instructions as appropriate for your OS. Where such translation is not obvious, we will specifically detail the steps necessary.

Launch JDeveloper 11g

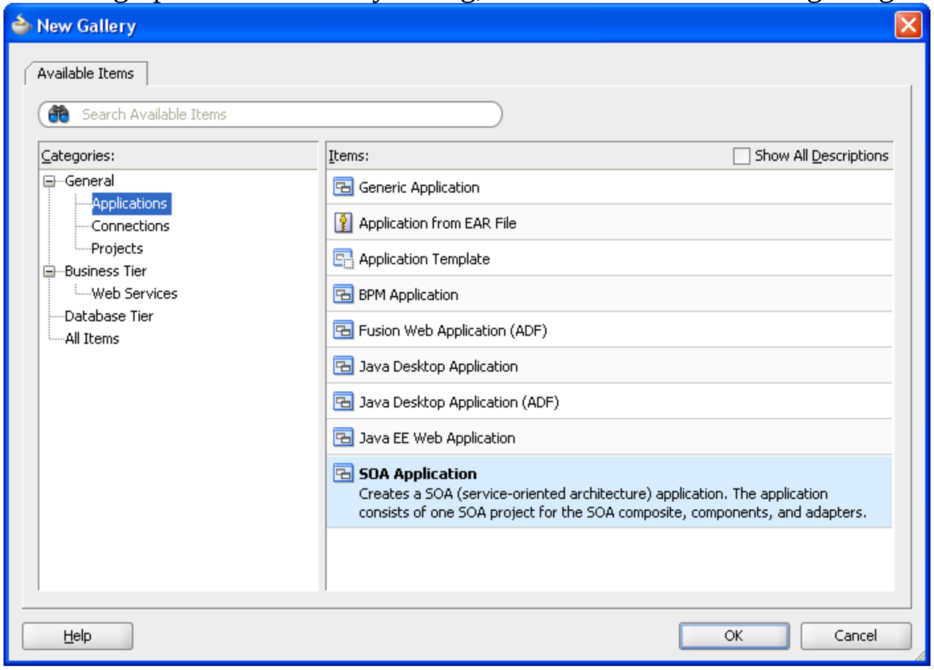
From the Windows Start menu, select the group named Oracle Fusion Middleware 11.1.1.3.0 and then select JDeveloper Studio, as shown in the following image.


Oracle SOA 11g - Home1
Oracle WebLogic (BEAHOME 2)
Oracle Fusion Middleware 11.1.1.3.0
Oracle Service Bus 11g - Home1
CoffeeCup Software
WebLogic Server 11gR1
JDeveloper Studio 11.1.1.3.0
QuickStart
Smart Update
Uninstall Oracle Middleware

When JDeveloper loads, it will appear as follows:

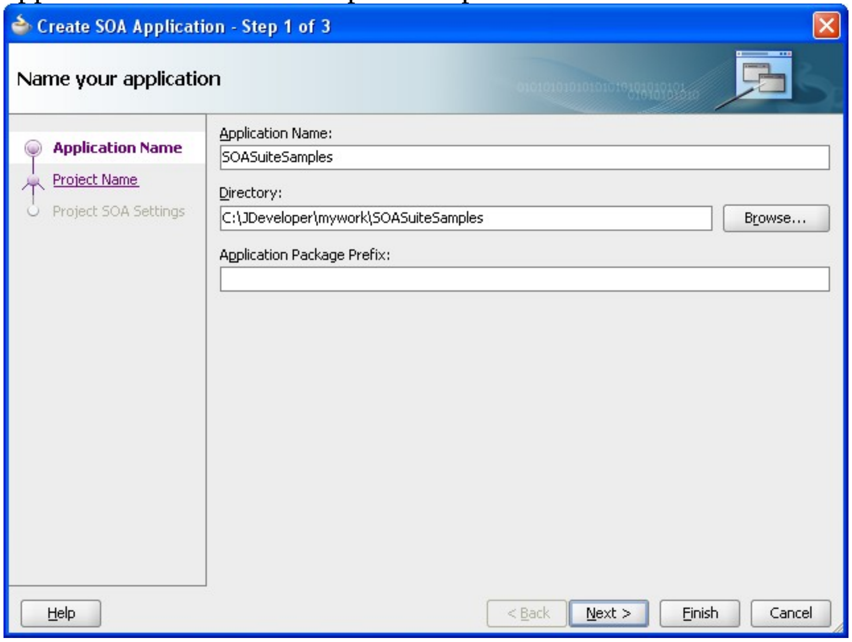


Our first step is to create a new application for our HelloWorld project. Press the New Application icon in JDeveloper or select File -> New from the JDeveloper menu. This will bring up the New Gallery dialog, as shown in the following image.



SOA Application
Creates a SOA (service-oriented architecture) application. The application consists of one SOA project for the SOA composite, components, and adapters.

No matter which technique you use, you want to create an SOA application. Name the application SOASUiteSamples and press the Next button.



Create SOA Application - Step 1 of 3

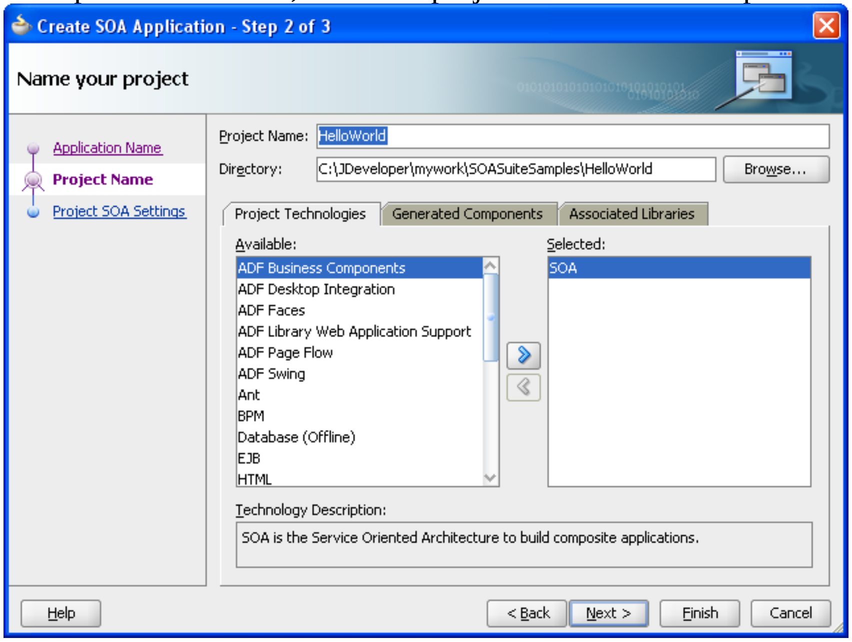
Name your application

Application Name: SOASuiteSamples

Directory: C:\JDeveloper\mywork\SOASuiteSamples

Application Package Prefix:

In step 2 of the wizard, name the project HelloWorld and press the Next button.



Create SOA Application - Step 2 of 3

Name your project

Project Name: HelloWorld

Directory: C:\JDeveloper\mywork\SOASuiteSamples\HelloWorld

Project Technologies

Available:

ADF Business Components
ADF Desktop Integration
ADF Faces
ADF Library Web Application Support
ADF Page Flow
ADF Swing
Ant
BPM
Database (Offline)
EJB
HTML

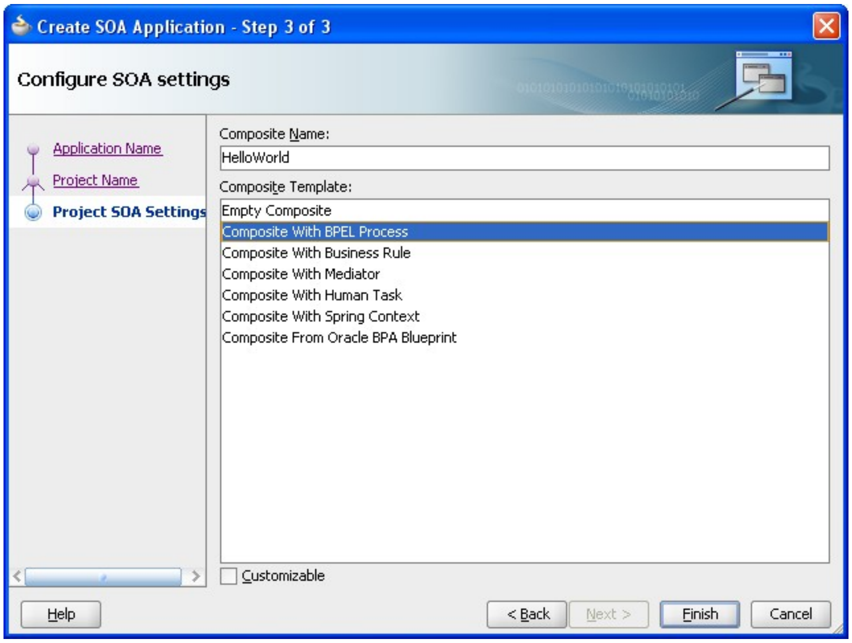
Selected:

SOA

Technology Description:

SOA is the Service Oriented Architecture to build composite applications.

In the last step of the wizard, select “Composite with BPEL Process” from the list of composite templates.

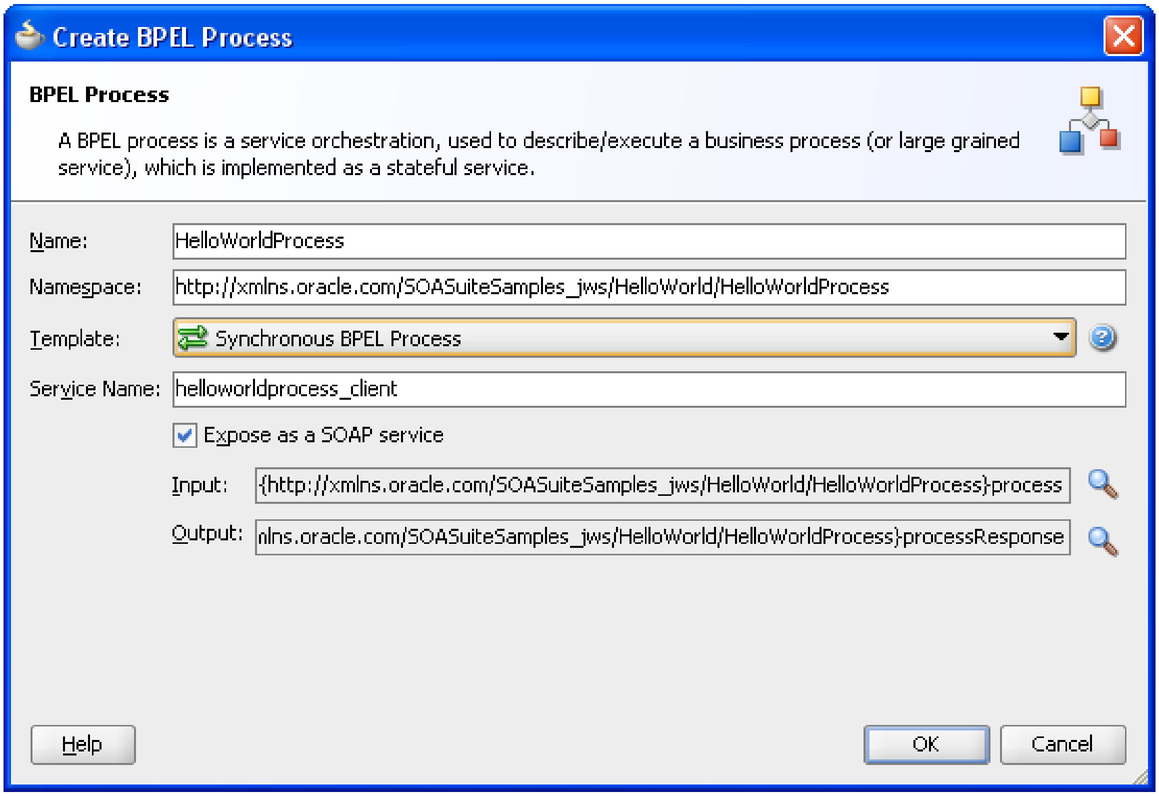


Composite Name: HelloWorld

Composite Template:
- Empty Composite
- Composite With BPEL Process
- Composite With Business Rule
- Composite With Mediator
- Composite With Human Task
- Composite With Spring Context
- Composite From Oracle BPA Blueprint

□ Customizable

Press the Finish button to close the wizard. JDeveloper will create your new application and project. Because you specified the “Composite with BPEL Process”, the Create BPEL Process dialog will appear to prompt you for a name for the new process. Set the name to HelloWorldProcess. In the Template field, be sure to select Synchronous BPEL Process.



A BPEL process is a service orchestration, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

Name: HelloWorldProcess

Namespace: http://xmlns.oracle.com/SOASuiteSamples_jws/HelloWorld/HelloWorldProcess

Template: Synchronous BPEL Process

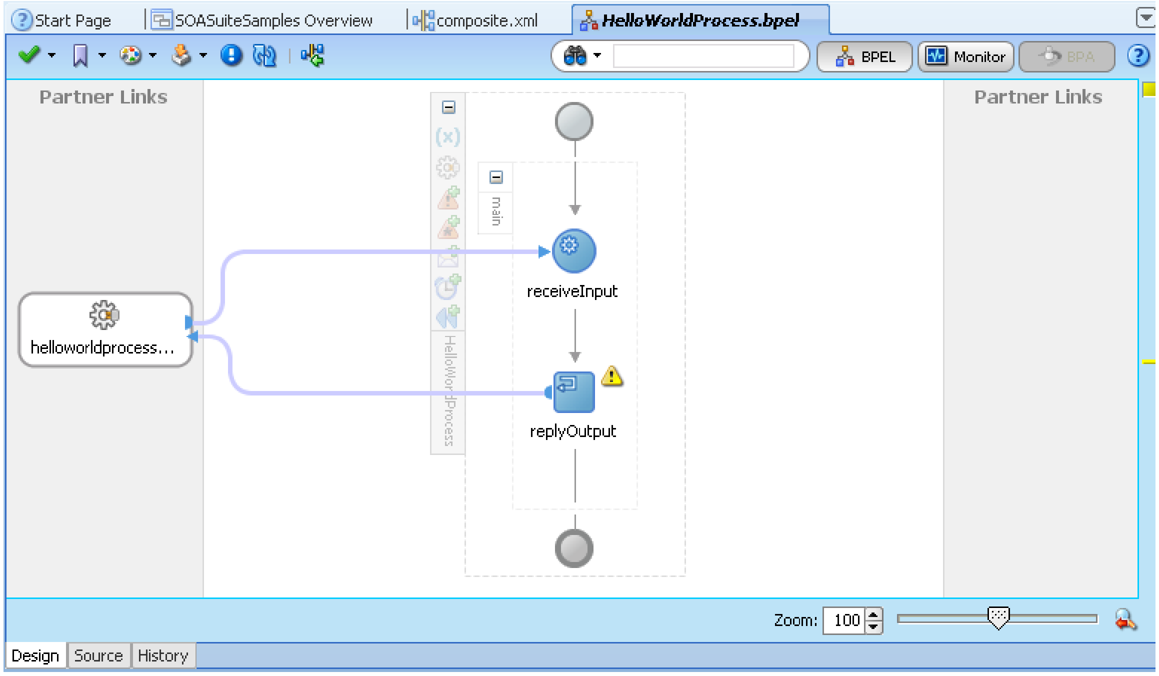
Service Name: helloworldprocess_client

☑ Expose as a SOAP service

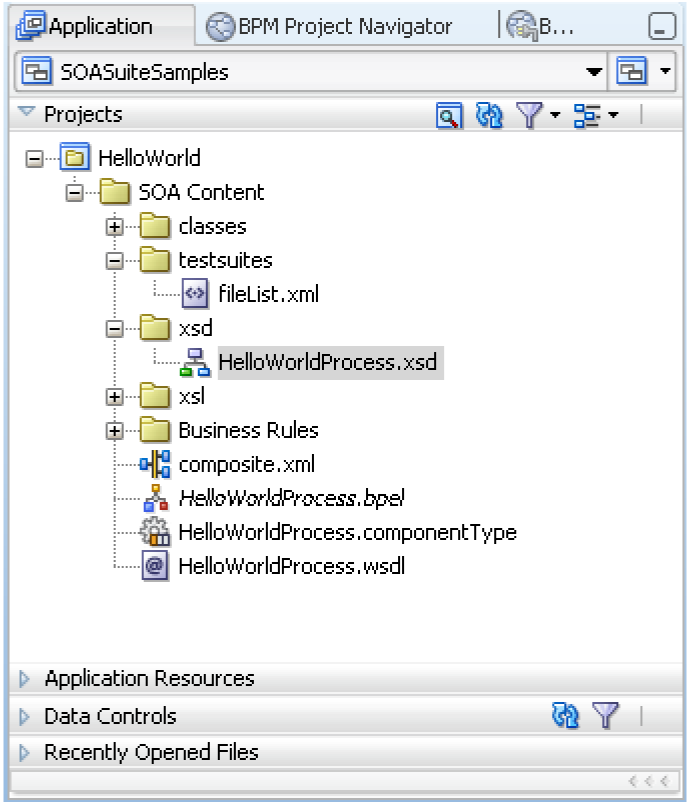
Input: {http://xmlns.oracle.com/SOASuiteSamples_jws/HelloWorld/HelloWorldProcess}process

Output: nlns.oracle.com/SOASuiteSamples_jws/HelloWorld/HelloWorldProcess}processResponse

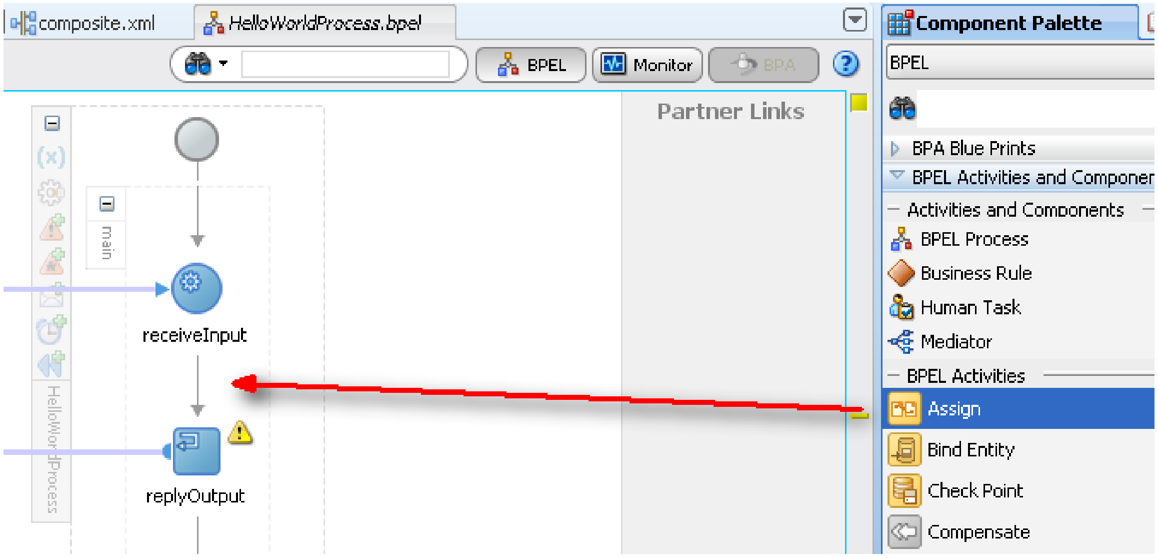
Leave the rest of the fields at their default values and press the OK button. JDeveloper should look like the following.



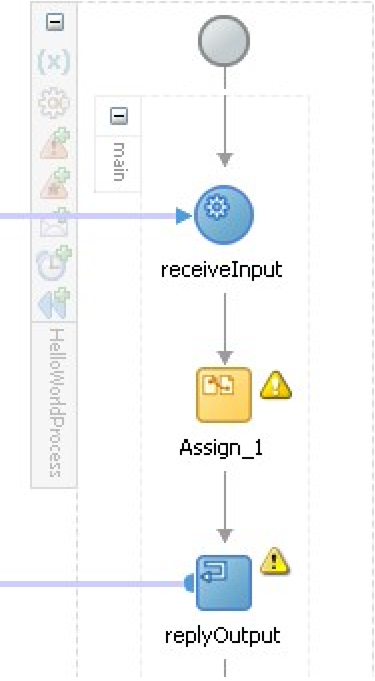
JDeveloper has created a WSDL for the BPEL process called *HelloWorldProcess.wsdl*. The *helloworldprocess* partner link (in the Partner Links swimlane on the left side of the figure) specifies that WSDL as the interface for the BPEL process. If you open the HelloWorld.wsdl file you would see that it has one operation named *process*. JDeveloper also created a HelloWorld.xsd file which is in the xsd/ folder of the project. The HelloWorld.xsd file contains the definitions for the operation arguments *process* and *processResponse*.You can find all of these files easily from the Application navigation window (pictured next).



All you need to do to complete this sample is to add an Assign activity between the receiveInput and replyOutput activities in the BPEL flow. You do this by dragging the Assign activity and dropping it between the receiveInput and replyOutput actions, as shown below.

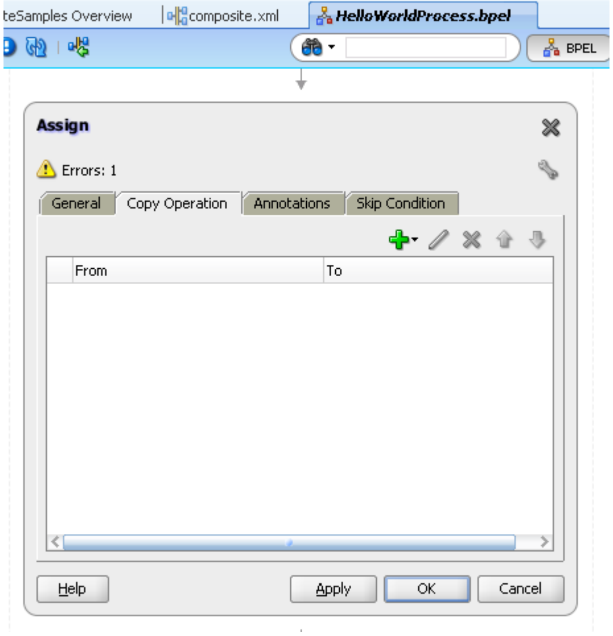


Your flow should now match the following:

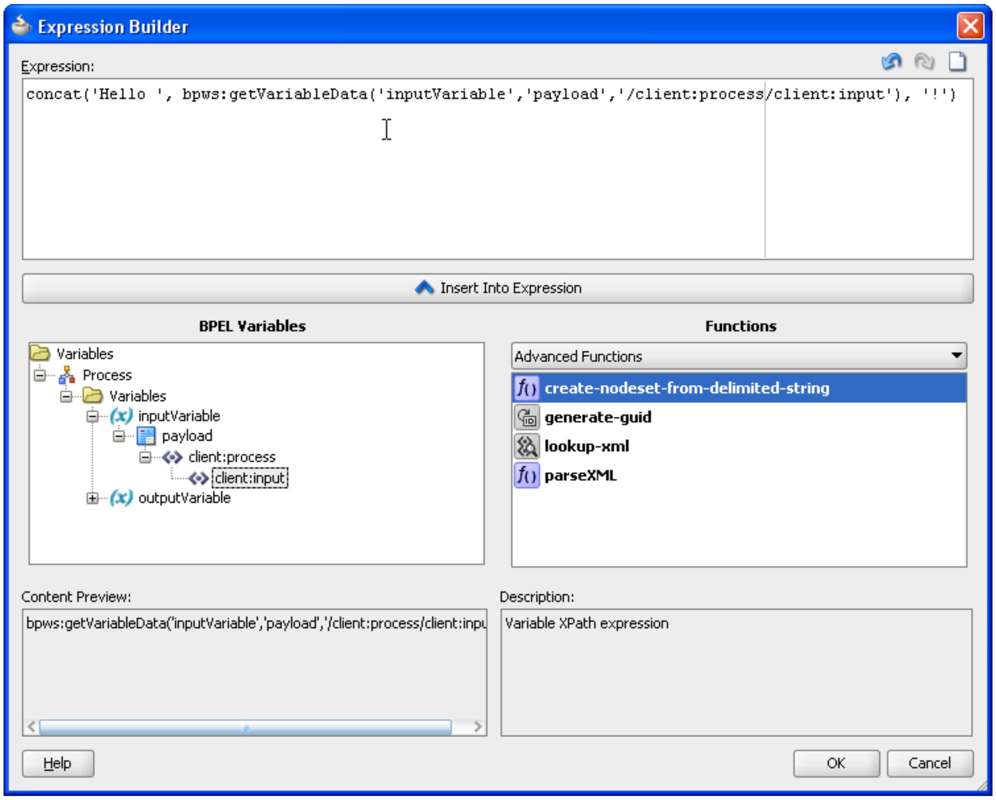


Next, you need to modify the assign statement so that it creates a new response string. The *process* message contains a string and for this sample we will assume that the string contains the name of a person. We want to create a new string that contains a response like “Hello Jeff!” We do that by modifying the Assign activity to concatenate the string in the *process* message with the other elements of our response message.

Double-click on the Assign activity that you placed into the flow. This will open up the Assign activity so you can edit it, as shown below.



Click on the green + symbol and select *Copy Operation…* from the popup menu. This will bring up the Create Copy Operation dialog where you will configure the copy action.



Expression:

```
concat('Hello ', bpws:getVariableData('inputVariable','payload','/client:process/client:input'), '!')
```

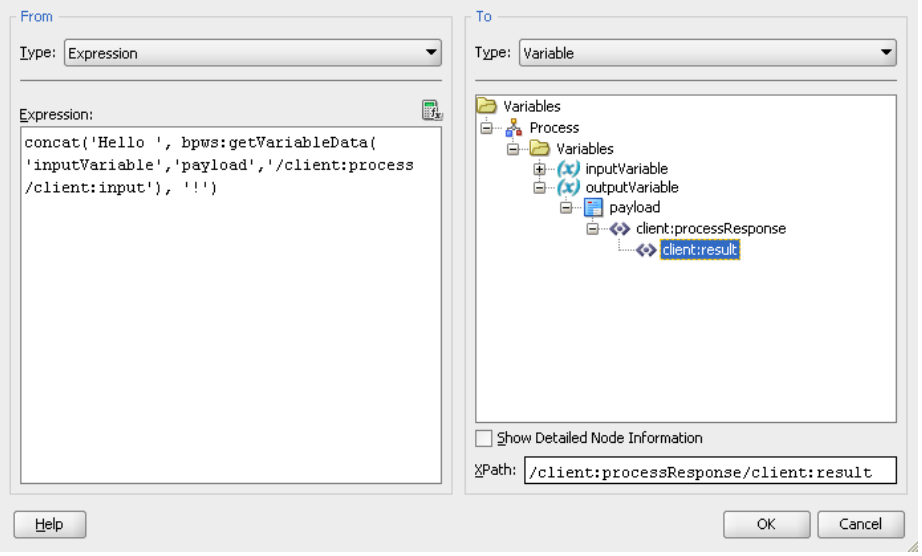
Content Preview:

```
bpws:getVariableData('inputVariable','payload','/client:process/client:inpu
```

Description: Variable XPath expression

In case the expression in the screenshot above is not clear, here it is again:
concat('Hello ',
bpws:getVariableData('inputVariable','payload','/client:process/client:input'), '!')

Press the OK button. Now you can set the To part of the copy operation. Select the client:result element as shown in the following screenshot.



From — Type: Expression

Expression:

```
concat('Hello ', bpws:getVariableData(
'inputVariable','payload','/client:process
/client:input'), '!')
```

To — Type: Variable

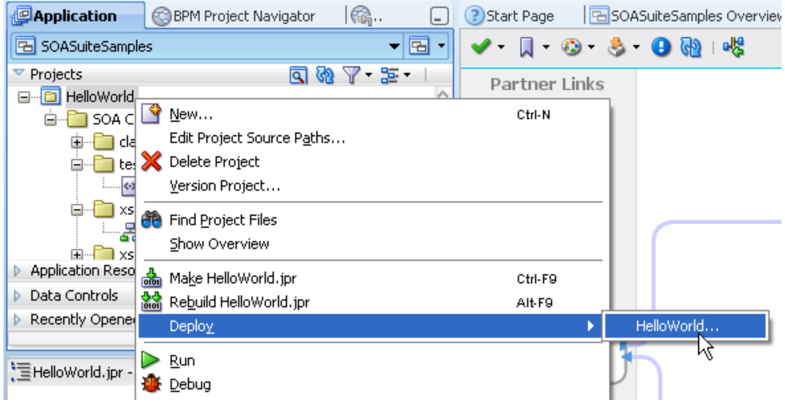
□ Show Detailed Node Information

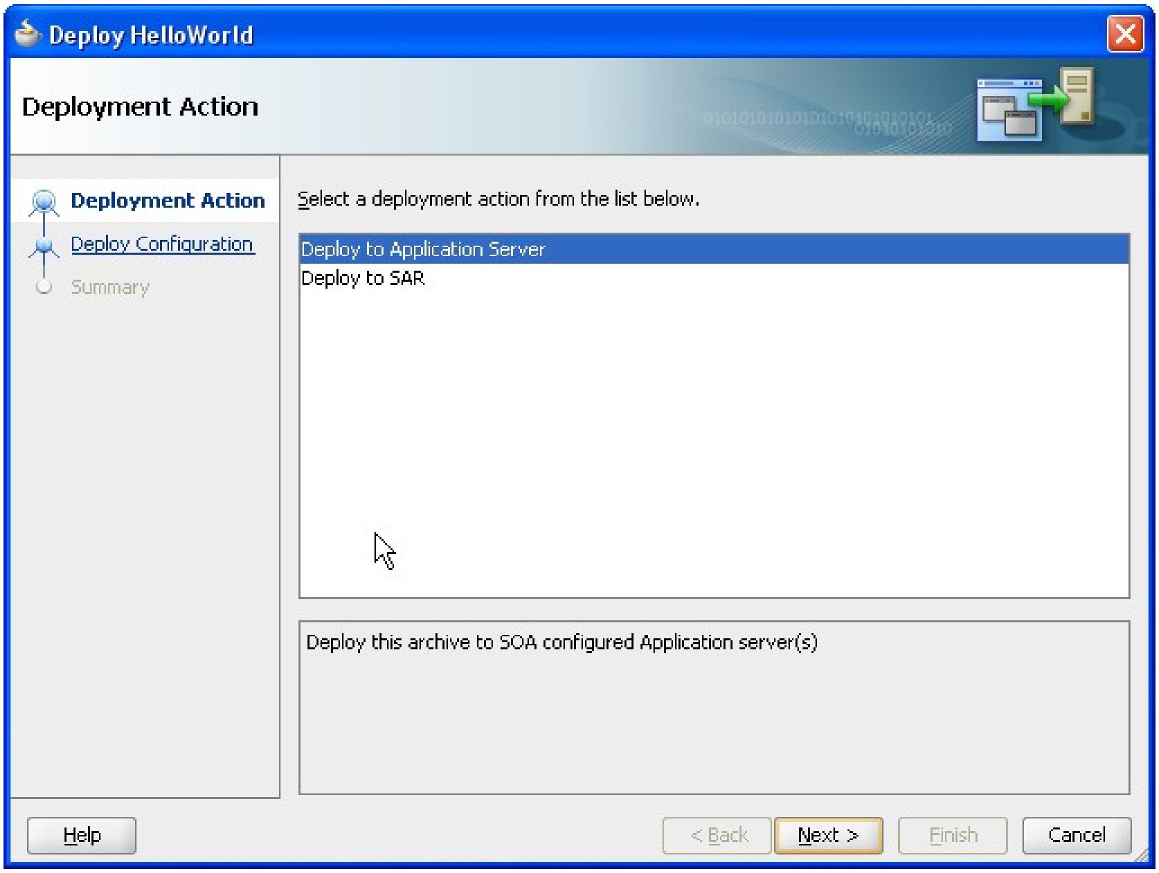
XPath: /client:processResponse/client:result

Press the OK button to close this dialog. In the Assign activity, press the OK button to close the activity. Your BPEL process is now complete and ready for deployment and testing.

Deploying the BPEL Process

You must deploy the BPEL process to a running server before you can use it, even for testing.



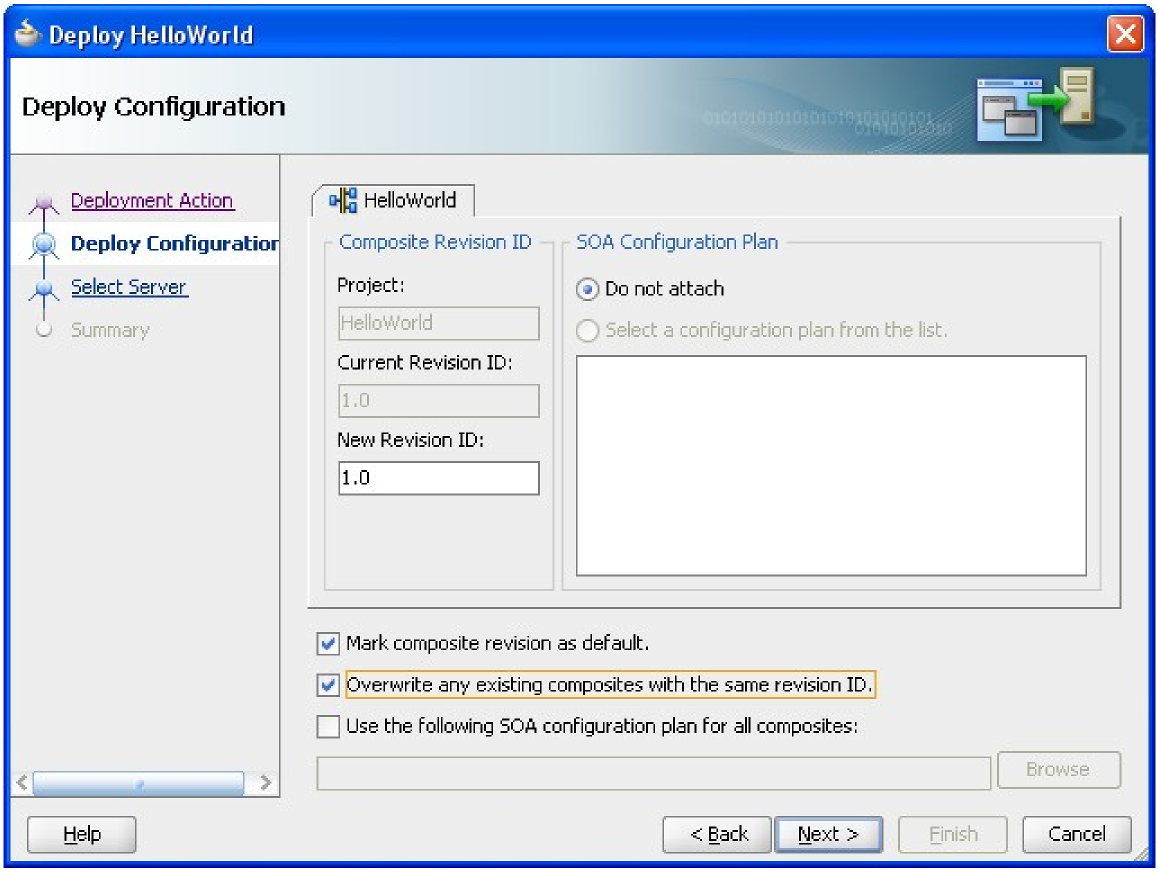


Deployment Action

Select a deployment action from the list below.

Deploy to Application Server
Deploy to SAR

Deploy this archive to SOA configured Application server(s)



Deploy HelloWorld

Deploy Configuration

- Deployment Action
- Deploy Configuration
- Select Server
- Summary

HelloWorld

Composite Revision ID

Project: HelloWorld

Current Revision ID: 1.0

New Revision ID: 1.0

SOA Configuration Plan

● Do not attach

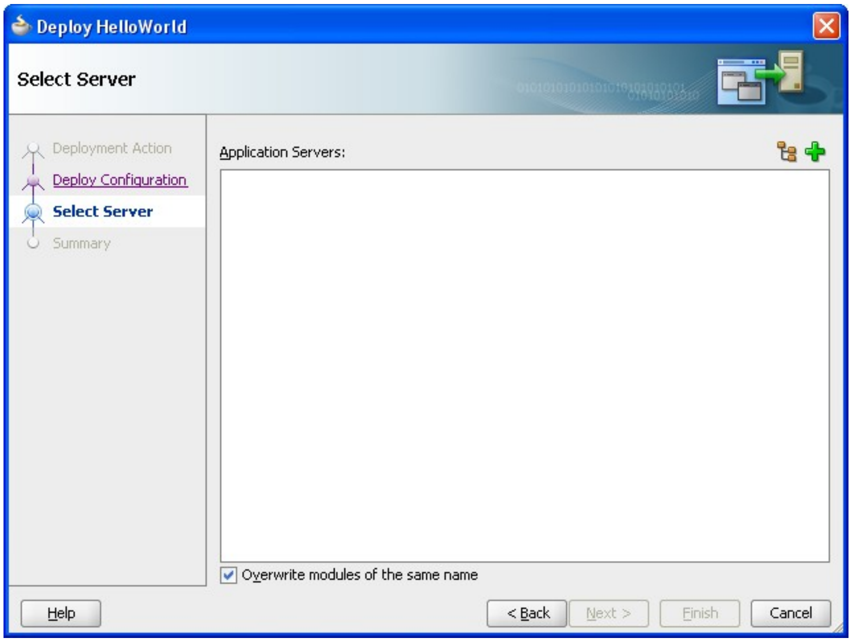
○ Select a configuration plan from the list.

☑ Mark composite revision as default.

☑ Overwrite any existing composites with the same revision ID.

□ Use the following SOA configuration plan for all composites:

Next you must select your application server. If do not yet have an application server defined, the following steps will help you create an application server for testing purposes. If you do have an application server defined that you want to use, simply select it and press the Next button. You can then skip over the following section on creating a server.



Deploy HelloWorld

Select Server

- Deployment Action
- Deploy Configuration
- Select Server
- Summary

Application Servers:

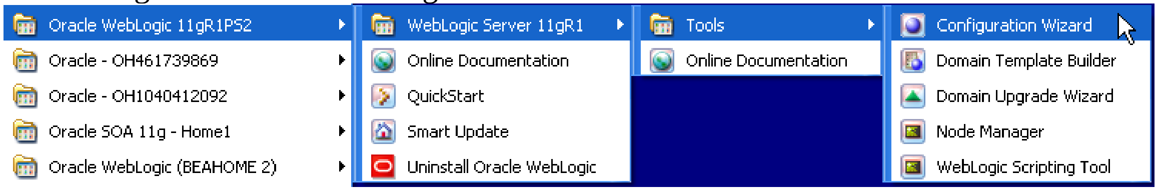
☑ Overwrite modules of the same name

Creating a Server

THIS MAY NOT BE NECESSARY

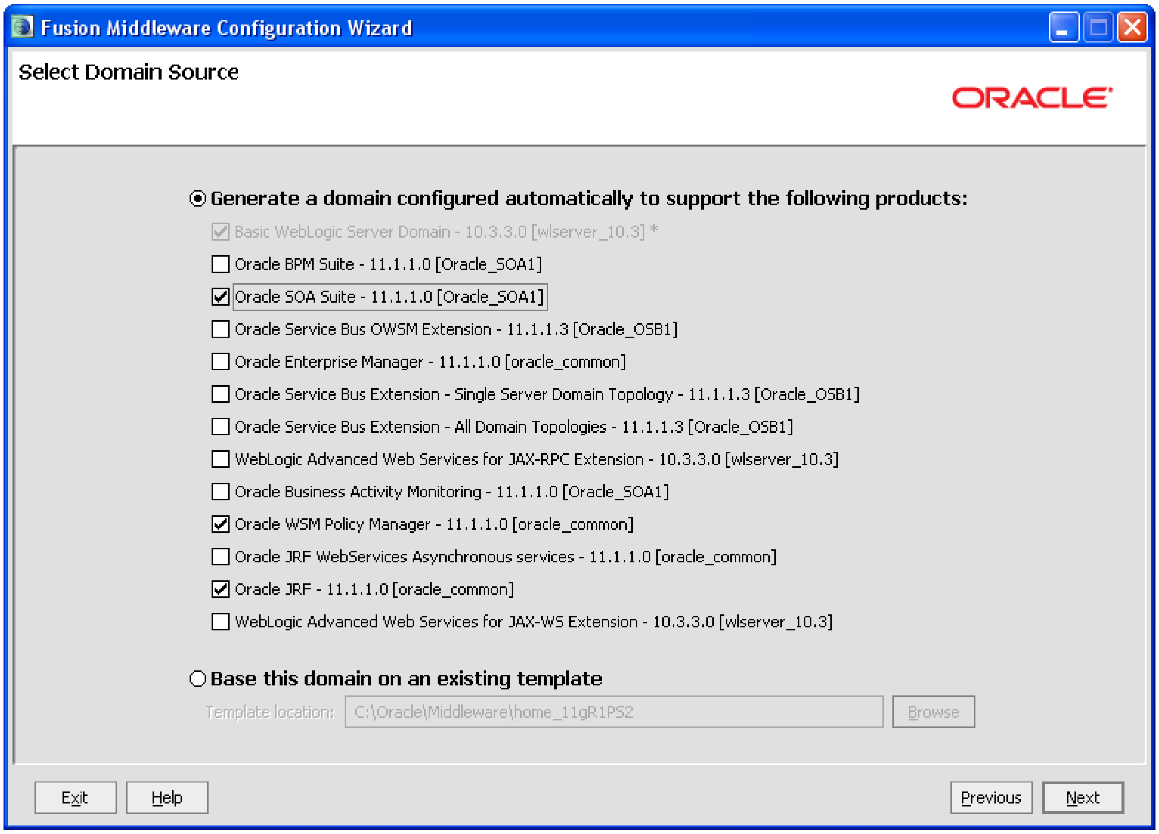
If you have not created a WebLogic Server domain to host your work, we will walk you through that process now. Follow these steps:

From the Windows Start menu, select All Programs -> Oracle WebLogic -> WebLogic Server 11gR1 -> Tools -> Configuration Wizard.



In the screenshot above you'll notice that my installation folder is called Oracle WebLogic 11gR1PS2 and not Oracle WebLogic. That is a configuration specific to my machine so don't be confused if your is different.

Once you start the configuration wizard, you will be asked to select a domain source, as shown in the following figure. Select the Oracle SOA Suite – 11.1.1.0 [Oracle_SOA1] and press the next button.



Select Domain Source

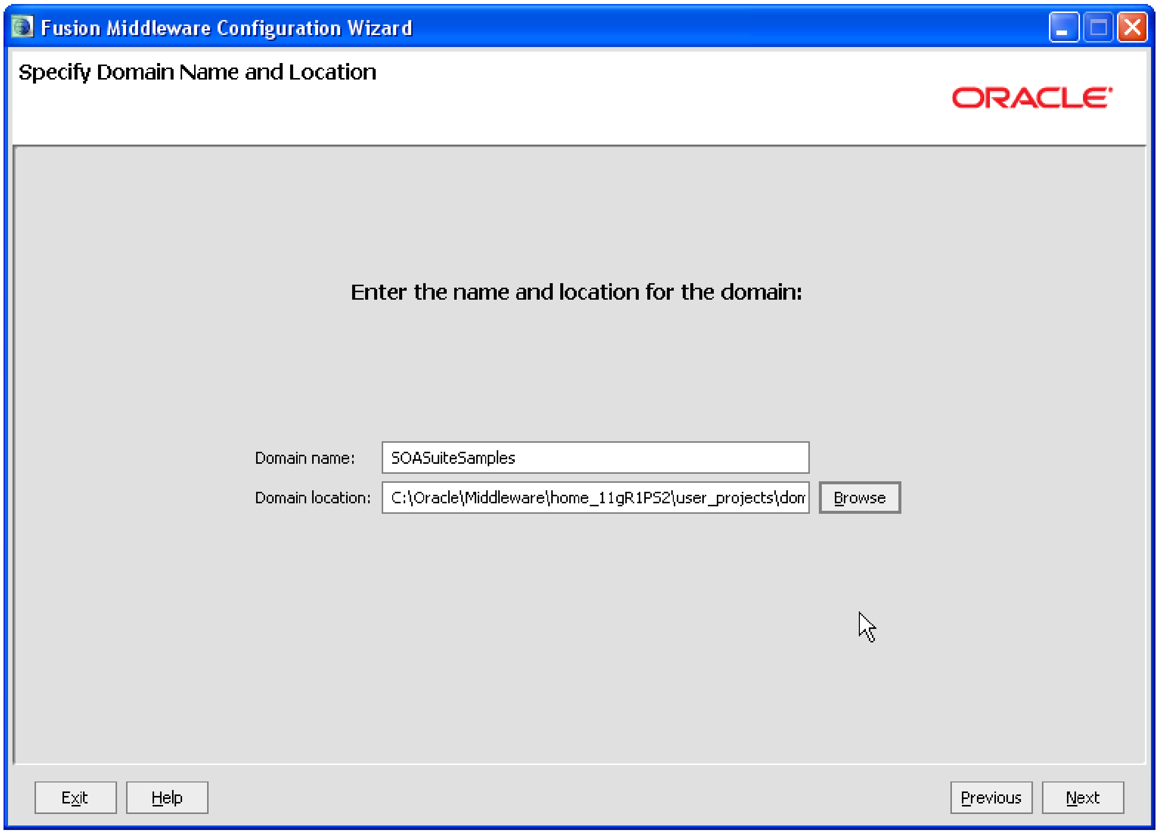
● Generate a domain configured automatically to support the following products:

☑ Basic WebLogic Server Domain - 10.3.3.0 [wlserver_10.3] *
☐ Oracle BPM Suite - 11.1.1.0 [Oracle_SOA1]
☑ Oracle SOA Suite - 11.1.1.0 [Oracle_SOA1]
☐ Oracle Service Bus OWSM Extension - 11.1.1.3 [Oracle_OSB1]
☐ Oracle Enterprise Manager - 11.1.1.0 [oracle_common]
☐ Oracle Service Bus Extension - Single Server Domain Topology - 11.1.1.3 [Oracle_OSB1]
☐ Oracle Service Bus Extension - All Domain Topologies - 11.1.1.3 [Oracle_OSB1]
☐ WebLogic Advanced Web Services for JAX-RPC Extension - 10.3.3.0 [wlserver_10.3]
☐ Oracle Business Activity Monitoring - 11.1.1.0 [Oracle_SOA1]
☑ Oracle WSM Policy Manager - 11.1.1.0 [oracle_common]
☐ Oracle JRF WebServices Asynchronous services - 11.1.1.0 [oracle_common]
☑ Oracle JRF - 11.1.1.0 [oracle_common]
☐ WebLogic Advanced Web Services for JAX-WS Extension - 10.3.3.0 [wlserver_10.3]

○ Base this domain on an existing template

Template location: C:\Oracle\Middleware\home_11gR1PS2 Browse

Next you will be prompted to select a name for the new WebLogic domain. Name your server as you wish. In the screenshot below we have named the server SOASuiteSamples. You can safely keep the default domain location or change it to suit your needs. Press the Next button when you are ready.



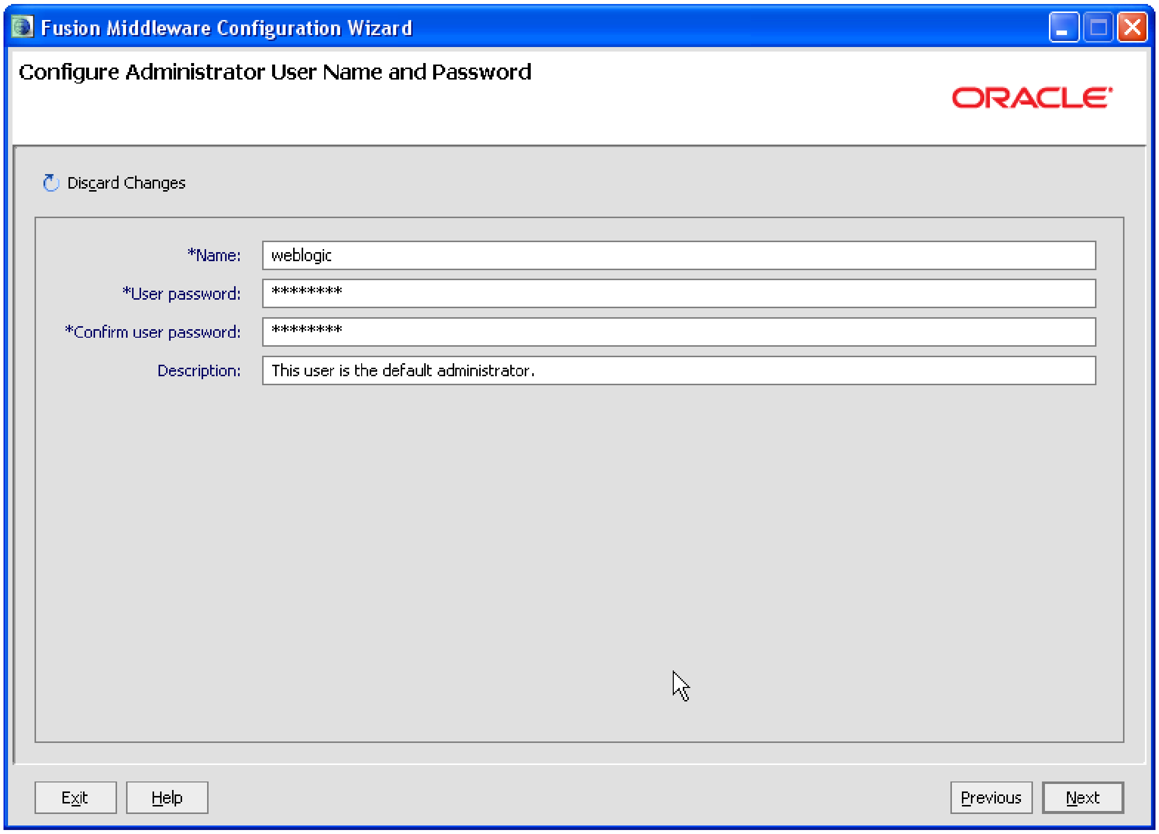
Specify Domain Name and Location

Enter the name and location for the domain:

Domain name: SOASuiteSamples

Domain location: C:\Oracle\Middleware\home_11gR1PS2\user_projects\dom

In the following dialog you specify the user name and password for the WebLogic administrator. Be sure you remember the username and password. You will need them to start the server in just a little bit. Press the Next button when you are ready.



Configure Administrator User Name and Password

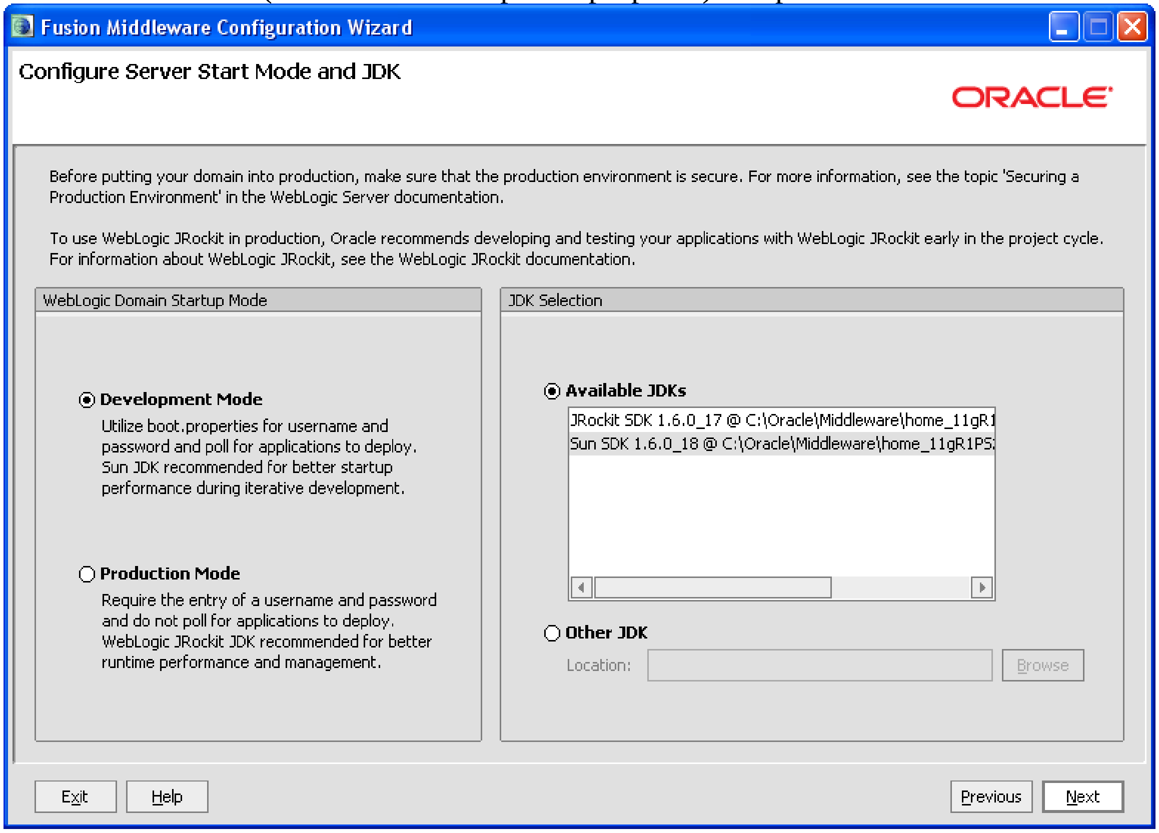
*Name: weblogic

*User password: ********

*Confirm user password: ********

Description: This user is the default administrator.

Select the Sun JDK (its best for development purposes) and press the Next button.



Configure Server Start Mode and JDK

Before putting your domain into production, make sure that the production environment is secure. For more information, see the topic 'Securing a Production Environment' in the WebLogic Server documentation.

To use WebLogic JRockit in production, Oracle recommends developing and testing your applications with WebLogic JRockit early in the project cycle. For information about WebLogic JRockit, see the WebLogic JRockit documentation.

WebLogic Domain Startup Mode

● **Development Mode**
Utilize boot.properties for username and password and poll for applications to deploy. Sun JDK recommended for better startup performance during iterative development.

○ **Production Mode**
Require the entry of a username and password and do not poll for applications to deploy. WebLogic JRockit JDK recommended for better runtime performance and management.

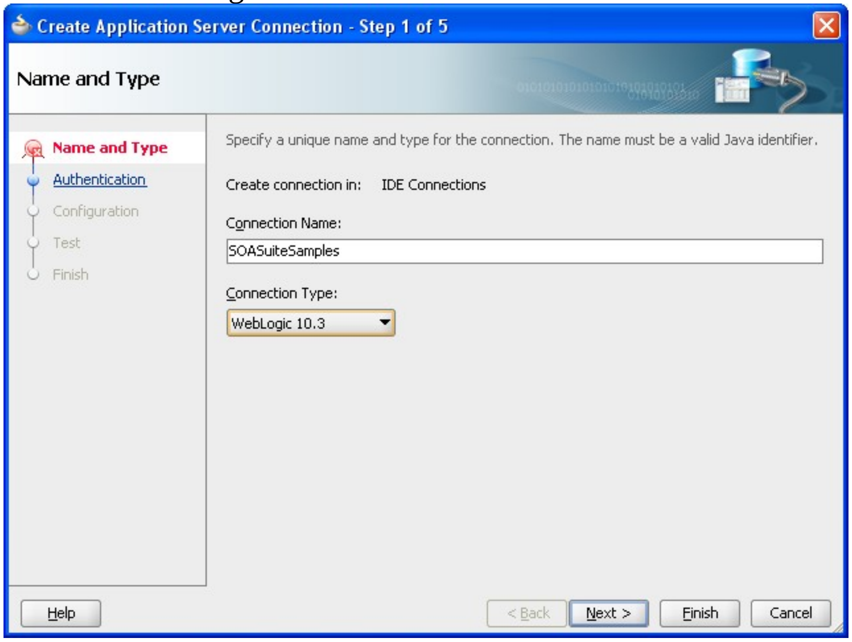
JDK Selection

● **Available JDKs**
JRockit SDK 1.6.0_17 @ C:\Oracle\Middleware\home_11gR1
Sun SDK 1.6.0_18 @ C:\Oracle\Middleware\home_11gR1PS2

○ **Other JDK**
Location:

Creating A Server Connection

To create a server, click on the green + icon in the top right corner of the Deploy HelloWorld dialog.



Create Application Server Connection - Step 1 of 5

Name and Type

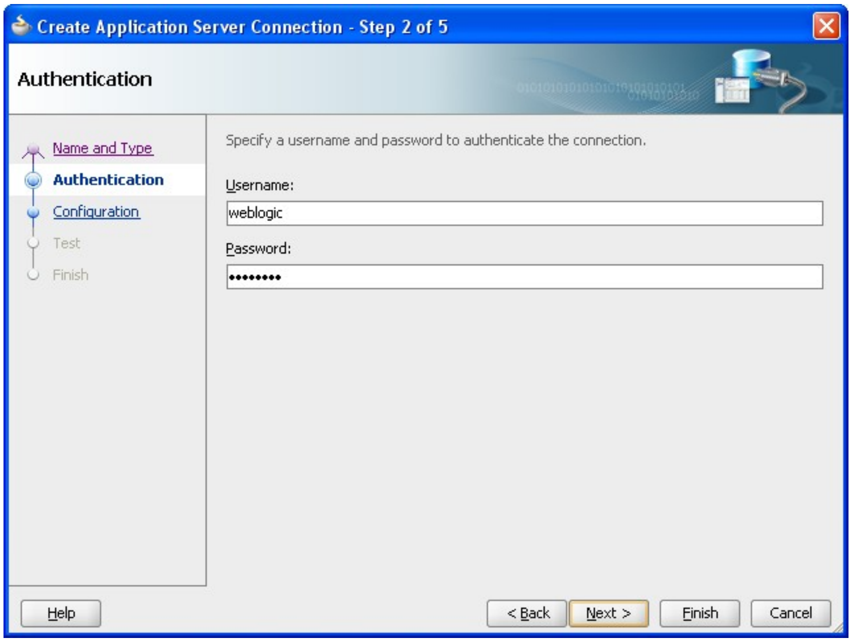
Specify a unique name and type for the connection. The name must be a valid Java identifier.

Create connection in: IDE Connections

Connection Name: SOASuiteSamples

Connection Type: WebLogic 10.3

Press Next



Create Application Server Connection - Step 2 of 5

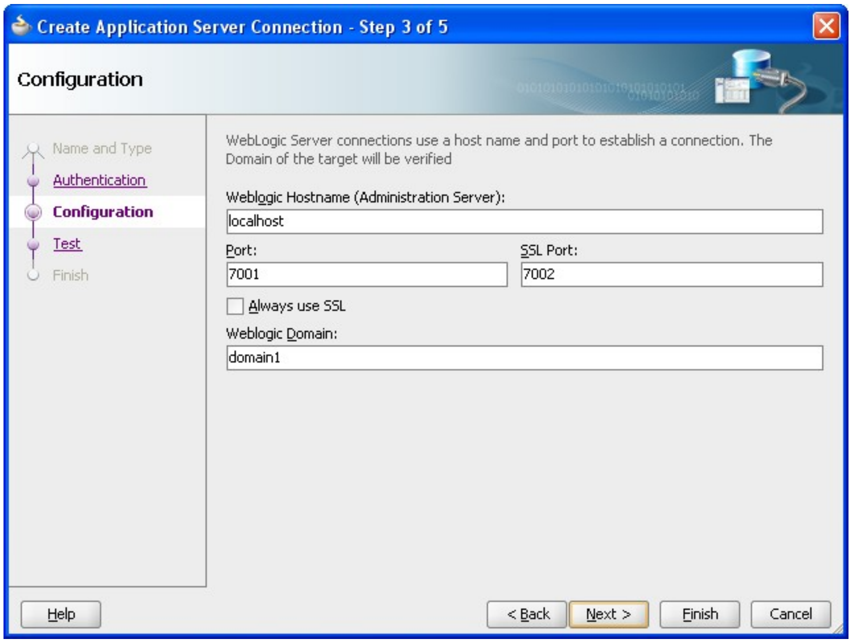
Authentication

Specify a username and password to authenticate the connection.

Username: weblogic

Password: ••••••••

Select a password and press Next. In step 3 of the wizard you specify the connection information. If you are using the domain that you created when you installed SOA Suite (which we recommend), be sure to set the WebLogic Domain field to *domain1*. Press Next when you are ready.



WebLogic Server connections use a host name and port to establish a connection. The Domain of the target will be verified

Weblogic Hostname (Administration Server): localhost

Port: 7001

SSL Port: 7002

□ Always use SSL

Weblogic Domain: domain1

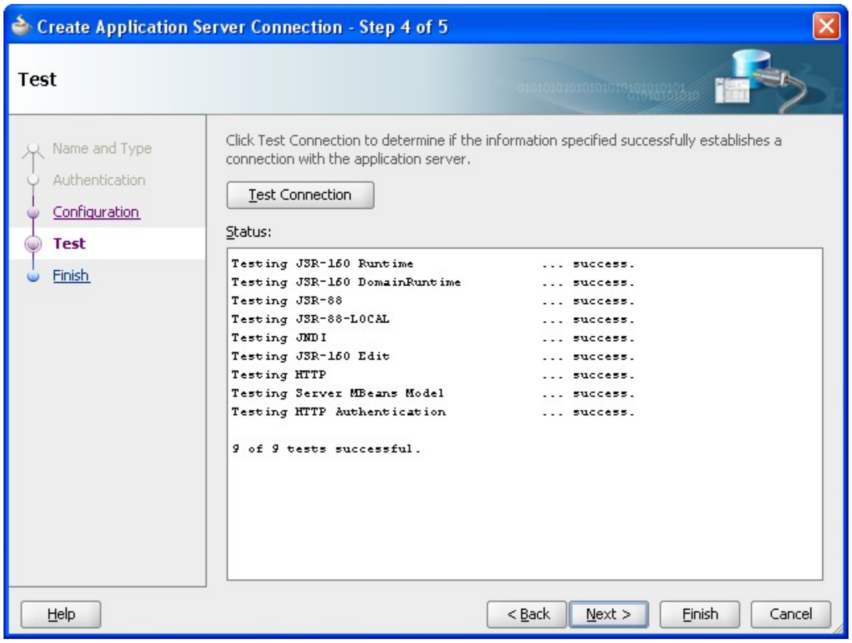
If you have not already started WebLogic domain1, you must do so before you can test your connection. Navigate to the domain directory (located at <ORACLE HOME>\user_projects\domains\domain1 and run the startWebLogic.cmd file. This will start the administration server for the domain1 domain. Wait a minute or two for the server to fully load and start running. You will know that the admin server is completely loaded when you see the following output on the console:

```
<May 17, 2010 4:49:53 PM PDT> <Notice> <WebLogicServer> <BEA-000365>
<Server state changed to RUNNING>
<May 17, 2010 4:49:53 PM PDT> <Notice> <WebLogicServer> <BEA-000360>
<Server started in RUNNING mode>
```

You also need to start the soa_server1 instance. The easiest way to do that is to open a command prompt in the domain1 directory and enter the following command:

```
bin\startManagedWebLogic.cmd soa_server1
```

Now you can test your application server connection by pressing the Test Connection button as shown in the following screenshot.



Click Test Connection to determine if the information specified successfully establishes a connection with the application server.

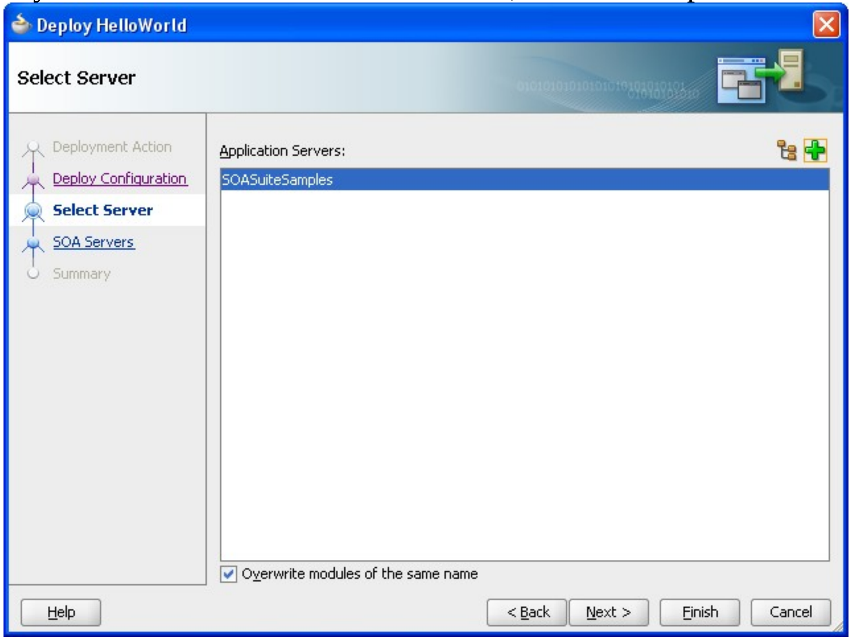
```
Testing JSR-160 Runtime                ... success.
Testing JSR-160 DomainRuntime          ... success.
Testing JSR-88                         ... success.
Testing JSR-88-LOCAL                   ... success.
Testing JNDI                           ... success.
Testing JSR-160 Edit                   ... success.
Testing HTTP                           ... success.
Testing Server MBeans Model            ... success.
Testing HTTP Authentication            ... success.

9 of 9 tests successful.
```

If all of the tests are successful press the Finish button. This will close the Create Application Server Connection wizard and return you to the Deployment wizard

Finishing the Deployment

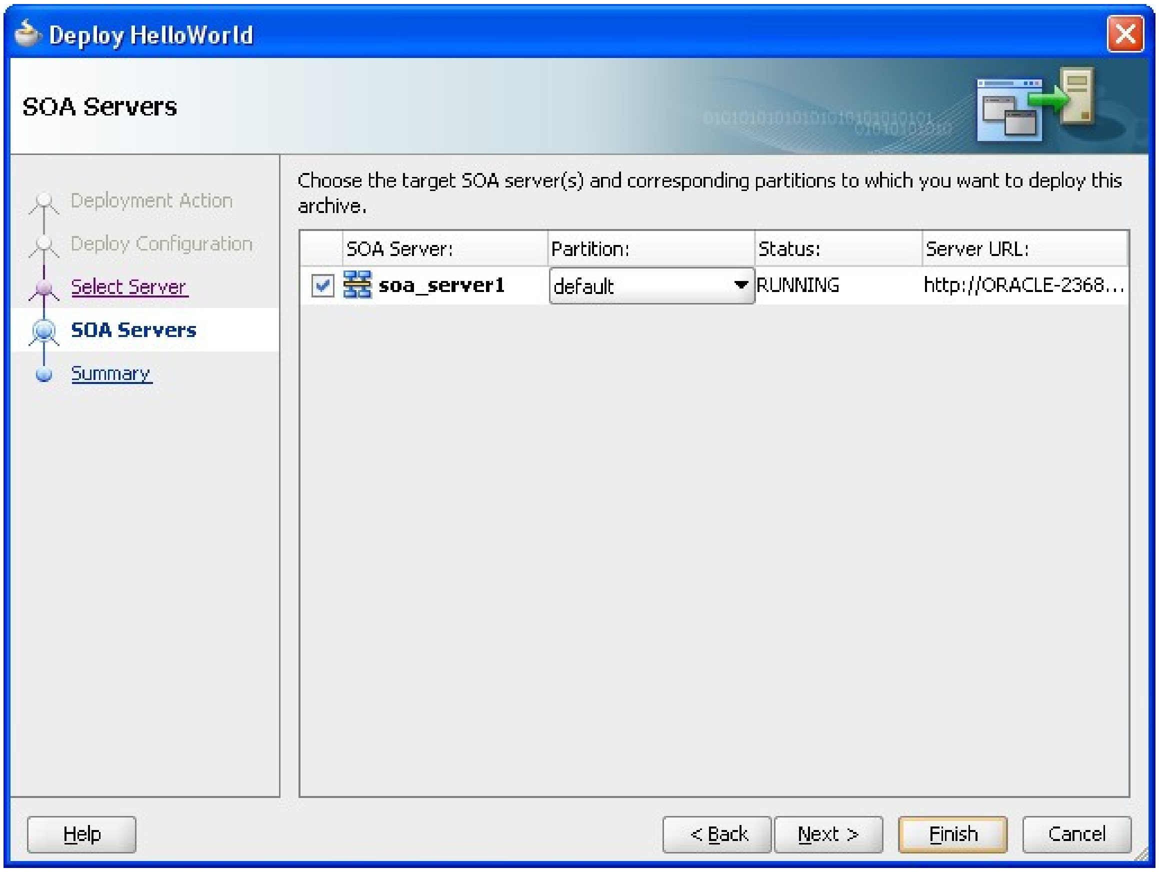
If you have a server connection defined, select it and press the Next button.



SOASuiteSamples

☑ Overwrite modules of the same name

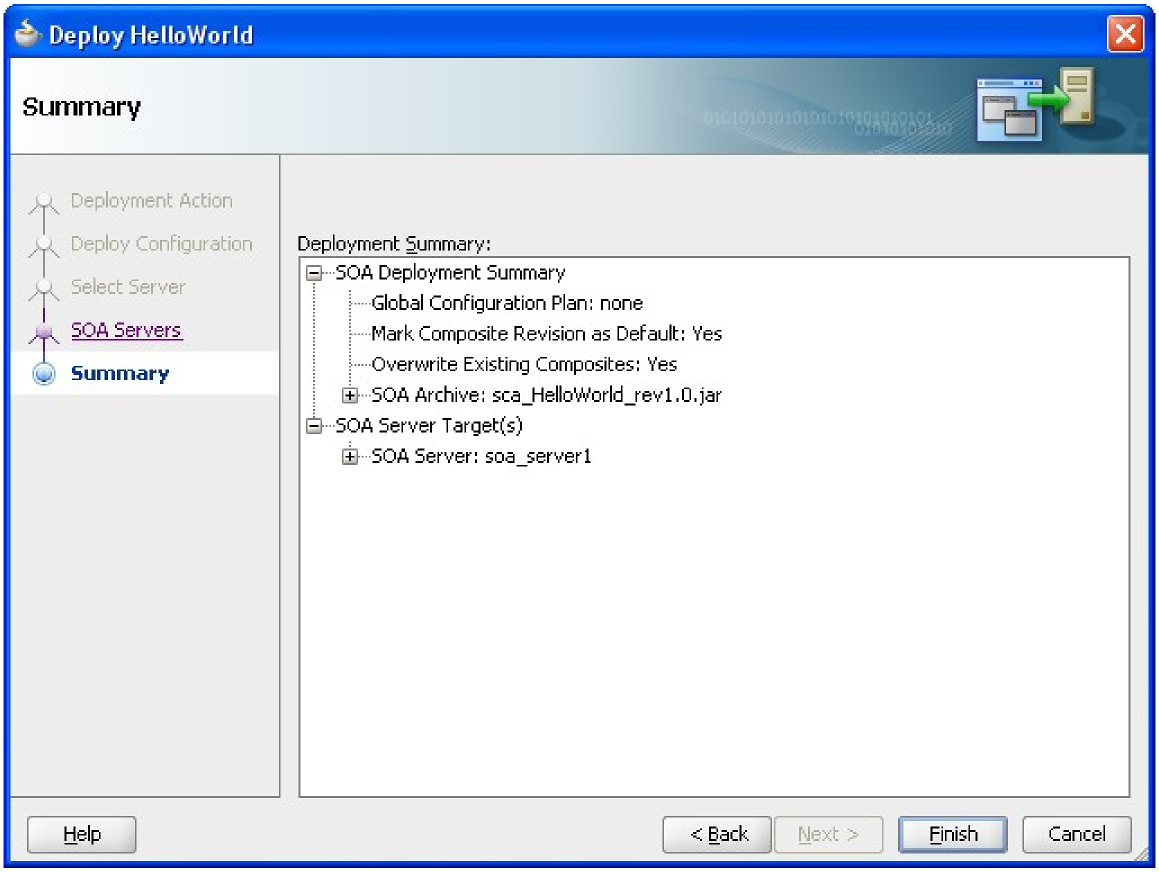
The next step in the wizard allows you to define the specific server(s) and partitions on which to run the deployment. This is shown in the following figure. Leave the defaults and press the Next button.



Choose the target SOA server(s) and corresponding partitions to which you want to deploy this archive.

	SOA Server:	Partition:	Status:	Server URL:
☑	**soa_server1**	default	RUNNING	http://ORACLE-2368...

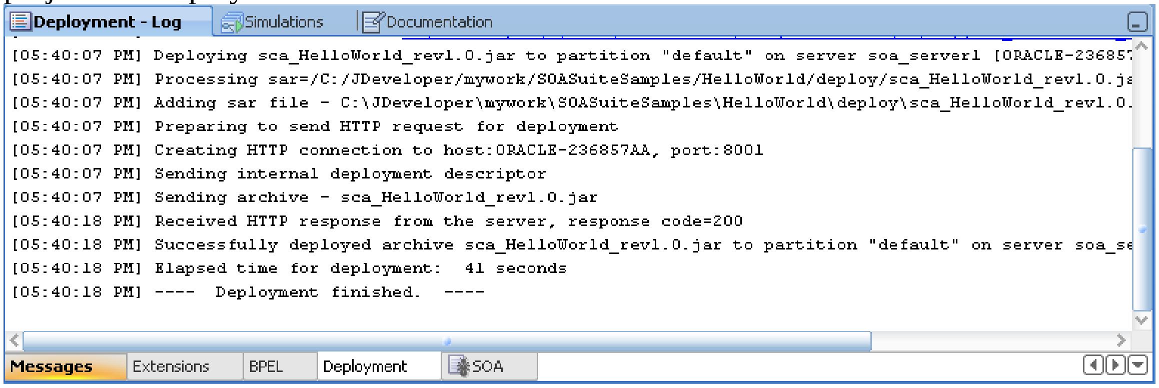
The last step of the deployment wizard is the confirmation page, as follows:



Deployment Summary:

- SOA Deployment Summary
  - Global Configuration Plan: none
  - Mark Composite Revision as Default: Yes
  - Overwrite Existing Composites: Yes
  - SOA Archive: sca_HelloWorld_rev1.0.jar
- SOA Server Target(s)
  - SOA Server: soa_server1

Press the Finish button. JDeveloper will now build your project and deploy it to the server. Check the Deployment Log at the bottom of JDeveloper to ensure that your project was deployed without errors.

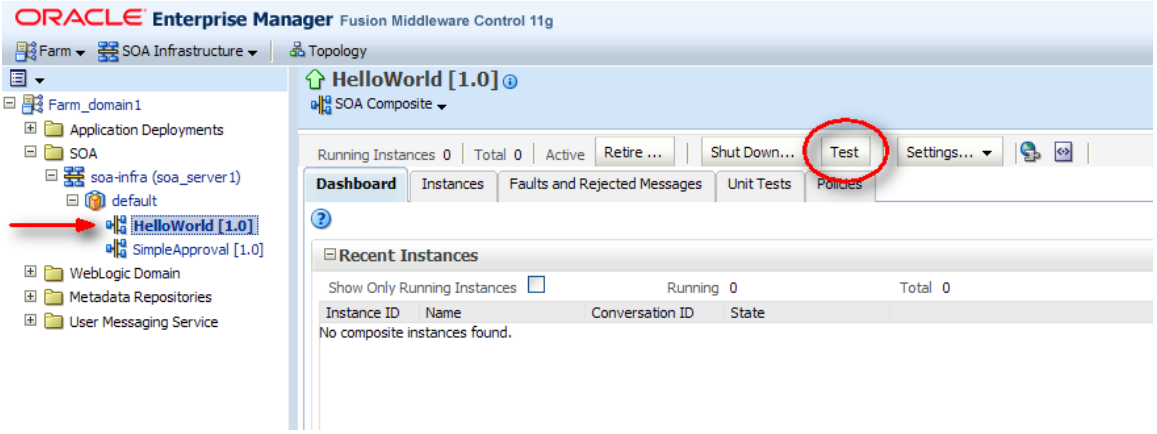


```
[05:40:07 PM] Deploying sca_HelloWorld_rev1.0.jar to partition "default" on server soa_server1 [ORACLE-236857
[05:40:07 PM] Processing sar=/C:/JDeveloper/mywork/SOASuiteSamples/HelloWorld/deploy/sca_HelloWorld_rev1.0.ja
[05:40:07 PM] Adding sar file - C:\JDeveloper\mywork\SOASuiteSamples\HelloWorld\deploy\sca_HelloWorld_rev1.0.
[05:40:07 PM] Preparing to send HTTP request for deployment
[05:40:07 PM] Creating HTTP connection to host:ORACLE-236857AA, port:8001
[05:40:07 PM] Sending internal deployment descriptor
[05:40:07 PM] Sending archive - sca_HelloWorld_rev1.0.jar
[05:40:18 PM] Received HTTP response from the server, response code=200
[05:40:18 PM] Successfully deployed archive sca_HelloWorld_rev1.0.jar to partition "default" on server soa_se
[05:40:18 PM] Elapsed time for deployment:  41 seconds
[05:40:18 PM] ----  Deployment finished.  ----
```

If it deployed successfully, you are ready to test your HelloWorld BPEL process.

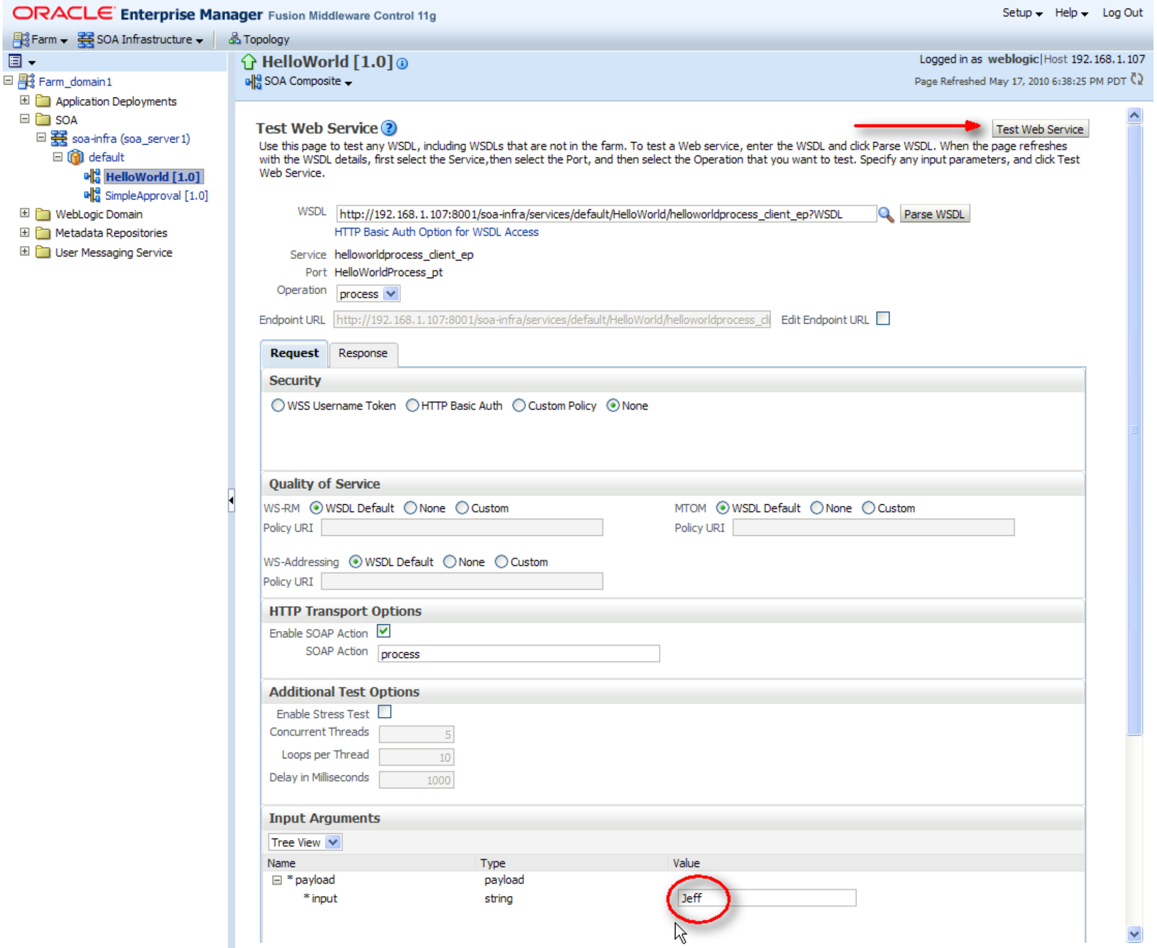
Testing the BPEL Process

Too test your BPEL process you need to use the Oracle Enterprise Manager (EM) web interface. Open your browser to http://localhhost:7001/em and login as the admin user. Using the navigation bar on the left side of the EM interface, navigate to the HelloWorld [1.0] entry and select it. When the page loads, press the Test button.



Instance ID	Name	Conversation ID	State
No composite instances found.			

Be sure too enter your name in the *input* field at the bottom of the test page. Then press the *Test Web Service* button.



Test Web Service

Use this page to test any WSDL, including WSDLs that are not in the farm. To test a Web service, enter the WSDL and click Parse WSDL. When the page refreshes with the WSDL details, first select the Service,then select the Port, and then select the Operation that you want to test. Specify any input parameters, and click Test Web Service.

WSDL http://192.168.1.107:8001/soa-infra/services/default/HelloWorld/helloworldprocess_client_ep?WSDL

Service helloworldprocess_client_ep

Port HelloWorldProcess_pt

Operation process

Endpoint URL http://192.168.1.107:8001/soa-infra/services/default/HelloWorld/helloworldprocess_cl Edit Endpoint URL □

Security

○ WSS Username Token ○ HTTP Basic Auth ○ Custom Policy ● None

Quality of Service

WS-RM ● WSDL Default ○ None ○ Custom Policy URI

MTOM ● WSDL Default ○ None ○ Custom Policy URI

WS-Addressing ● WSDL Default ○ None ○ Custom Policy URI

HTTP Transport Options

Enable SOAP Action ☑

SOAP Action process

Additional Test Options

Enable Stress Test □

Concurrent Threads 5

Loops per Thread 10

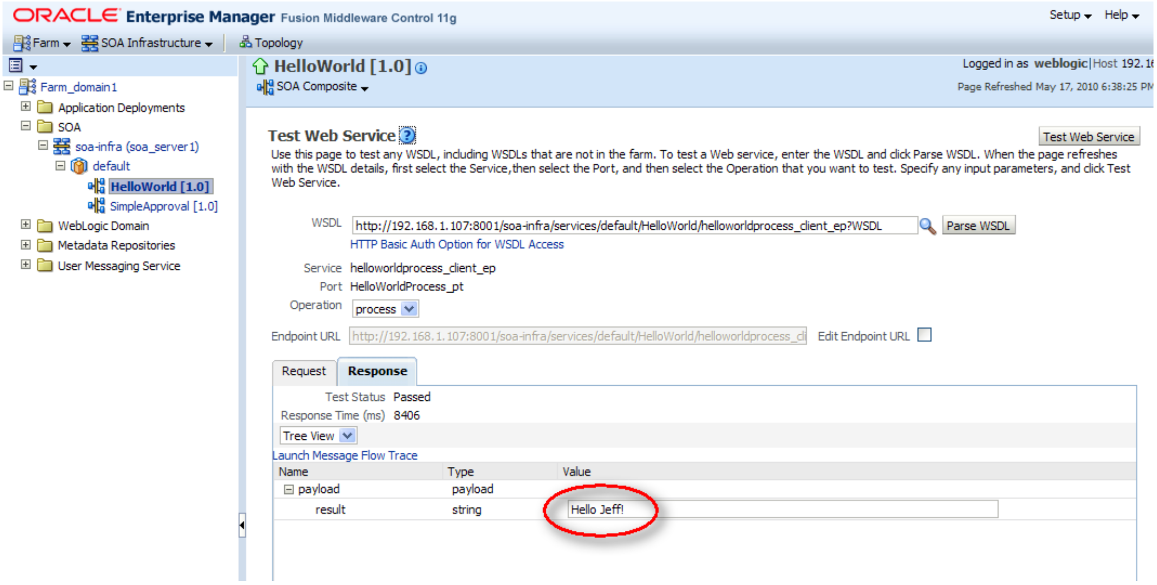
Delay in Milliseconds 1000

Input Arguments

Tree View

Name	Type	Value
* payload	payload	
* input	string	Jeff

The Response tab will automatically be selected when the web service call returns. Examine the result string at the bottom of the page to see the response message from the web service.



Test Web Service

Use this page to test any WSDL, including WSDLs that are not in the farm. To test a Web service, enter the WSDL and click Parse WSDL. When the page refreshes with the WSDL details, first select the Service,then select the Port, and then select the Operation that you want to test. Specify any input parameters, and click Test Web Service.

WSDL http://192.168.1.107:8001/soa-infra/services/default/HelloWorld/helloworldprocess_client_ep?WSDL

Service helloworldprocess_client_ep

Port HelloWorldProcess_pt

Operation process

Endpoint URL http://192.168.1.107:8001/soa-infra/services/default/HelloWorld/helloworldprocess_cli

Edit Endpoint URL □

Test Status Passed

Response Time (ms) 8406

Tree View

Name	Type	Value
⊟ payload	payload	
result	string	Hello Jeff!

The first time you call the web service it may actually take a couple of seconds to respond. That is because the WebLogic Server that is hosting the web service needs to initialize a few things in the background on the very first call. Click on the *Request* tab in the EM Test Web Service page and enter a different name in the *input* field and run the test again. You will see the second time it responds instantly!

Importing the Sample Code

If you want to see the working code provided with this sample you will need to import the application into your JDeveloper environment.

Next Steps

.Check out the other samples on the OTN page for SOA Suite 11g. You'll find all of the samples located at `https://soasamples.samplecode.oracle.com/`

SUMMARY

In this sample we have covered the very basics of developing a BPEL process and deploying it as an SCA composite. We have also shown you how to test your BPEL processes to ensure that they are working properly.