



Hello Jane Doe Sample

Hello World Series

AN ACTIVE ENDPOINTS SAMPLE

© 2010 Active Endpoints Inc. ActiveVOS is a trademark of Active Endpoints, Inc. All other company and product names are the property of their respective owners.



2010

Content

Hello World: Build a Process from Scratch	3
Part 1: Understanding the Building Blocks for Hello World	3
Part 2: Create an Orchestration Project.....	4
Step 1. Create a new Orchestration Project and BPEL file	4
Part 3: Determine the Input and Output Data Needed	5
Step 1. Create the sample data files for the receive and reply	5
Part 4: Use the Interface Wizard to Generate WSDL	6
Step 1. Create the WSDL interface for the process.....	6
Part 5: Create the Receive and Reply Activities	9
Step 1. Create the receive and reply activities.....	9
Part 6: Map Data in an Assign Activity	11
Step 1. Create a variable to handle the process input data	11
Step 2. Create an XPath expression for the output data	11
Step 3. Link the activities	Error! Bookmark not defined.
Part 7: Simulate the Process	13
Step 1. Modify the sample XML files to add data values	13
Step 2. Simulate the process.....	13
Part 8: Deploy the Process to the Server	15
Step 1. Create a process deployment descriptor file	15
Step 2. Starting the ActiveVOS Server.....	16
Step 3. Create a business process deployment archive	17
Part 9: Run the Process on the Server	19
Step 1. Open a browser to view the ActiveVOS Console	19
Step 2. Launch the Web Services Explorer.....	21
Step 3. Send a request to the process	22
Hello World Summary	25
Next Steps.....	25
About Active Endpoints	25

Hello World: Build a Process from Scratch

The Hello World Tutorial helps you begin your own orchestration project. We will begin without having any building blocks, such as WSDL files or schemas.

In this tutorial, we work with the following concepts and tasks:

- Analyze the input and output messages for a BPEL process
- Create sample XML documents for the input and output messages
- Generate the interface for the BPEL process
- Finish building the process and simulate it
- Deploy and run the process

Continue to Part 1: Understanding the Building Blocks for Hello World.

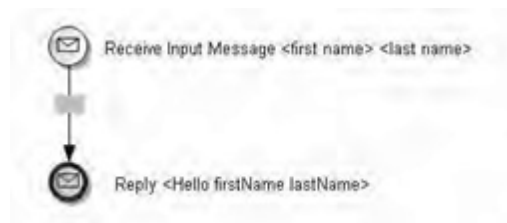
Part 1: Understanding the Building Blocks for Hello World

In this tutorial, we will build a BPEL process that provides a single service.

The Hello World service receives a client's name and then replies "Hello client". In a bit more detail, the service does the following:

- Receives client's data, namely <first name> and <last name>
- Replies to client with <Hello firstName lastName>

The following illustration shows the concept of the process:



The building blocks for this service include:

- Sample message data representing client input and process output
- WSDL interface for the BPEL process
- BPEL process

Continue to Part 2: Create an Orchestration Project.

Part 2: Create an Orchestration Project

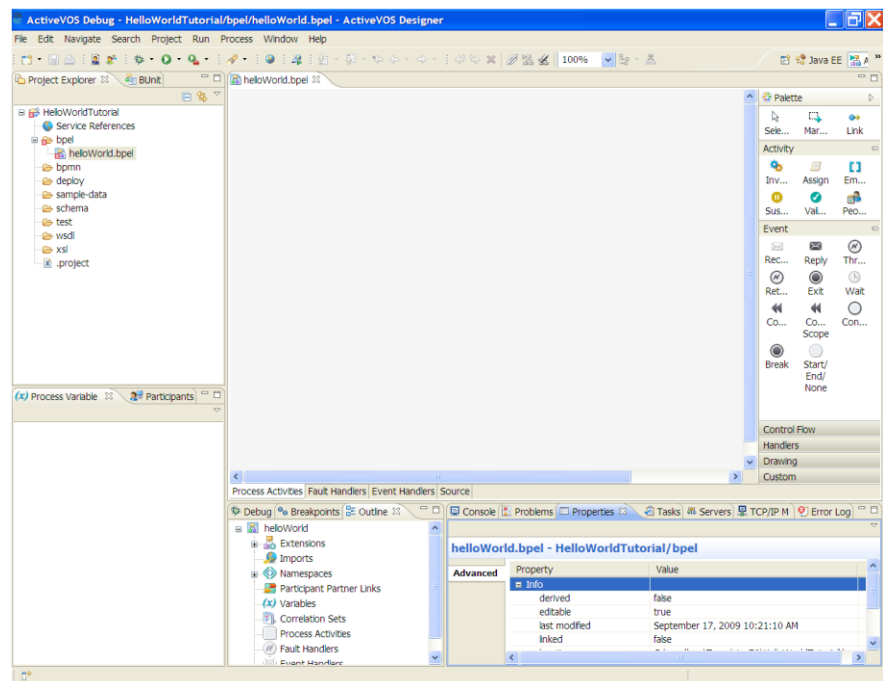
For an overview of the tutorial, see Hello World: Build a Process from Scratch.

We will create a workspace project to contain the building blocks described in Part 1: Understanding the Building Blocks for Hello World.

Step 1. Create a new Orchestration Project and BPEL file

1. In the Project Explorer, select File>New>Orchestration Project.
2. Name the project HelloWorldTutorial, and select Finish.
3. Select File>New>BPEL Process.
4. Create a new BPEL file in the HelloWorldTutorial\bpel folder called helloWorld.bpel.

Your ActiveVOS Designer perspective should look similar to the following:



Continue to Part 3: Determine the Input and Output Data Needed.

Part 3: Determine the Input and Output Data Needed

To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

The Hello World BPEL process needs a starting activity, in our case a receive activity, and a matching reply activity. A receive activity includes many properties defined in a WSDL file, but we do not (yet) have a WSDL.

So, let's start with something that we do know: the data our service receives and returns.

The data that a client sends to start the process includes two elements:

- <FirstName>
- <LastName>

The data returned to the client has only one element: <HelloResponse>.

With this data, we can create sample XML files. With the sample data files, we can create a WSDL definition.

Step 1. Create the sample data files for the receive and reply

1. Select File>New>XML File.
2. Select HelloWorldTutorial/sample-data for the parent folder, name the file helloWorldRequest.xml and select Next.
3. Select Create XML File from an XML Template. Select Finish. A new XML file opens in a Design tab.
4. Open the Source tab of the XML Editor.
5. Add the following lines to the helloWorldRequest.xml file:

Note: Ensure that there are no spaces in the namespace (due to line breaks) after you paste the following lines.

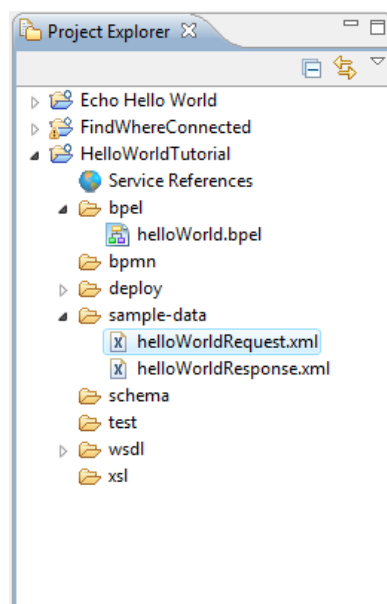
```
<hello:helloWorldRequest
  xmlns:hello="http://www.example.org/activevos/process
/helloWorld"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <hello:lastName>Doe</hello:lastName>
  <hello:firstName>Jane</hello:firstName>
</hello:helloWorldRequest>
```

6. Save the file.
7. Repeat Steps 1 - 6 to create the helloWorldResponse.xml file.

Add the following lines to the helloWorldResponse.xml file, ensuring that the namespace has no spaces in it:

```
<hello:helloWorldResponse
  xmlns:hello="http://www.example.org/activevos/process
/helloWorld"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <hello:response>Hello Jane Doe</hello:response>
</hello:helloWorldResponse>
```

Your Orchestration Project should look like the following illustration:



Continue to Part 4: Use the Interface Wizard to Generate WSDL.

Part 4: Use the Interface Wizard to Generate WSDL

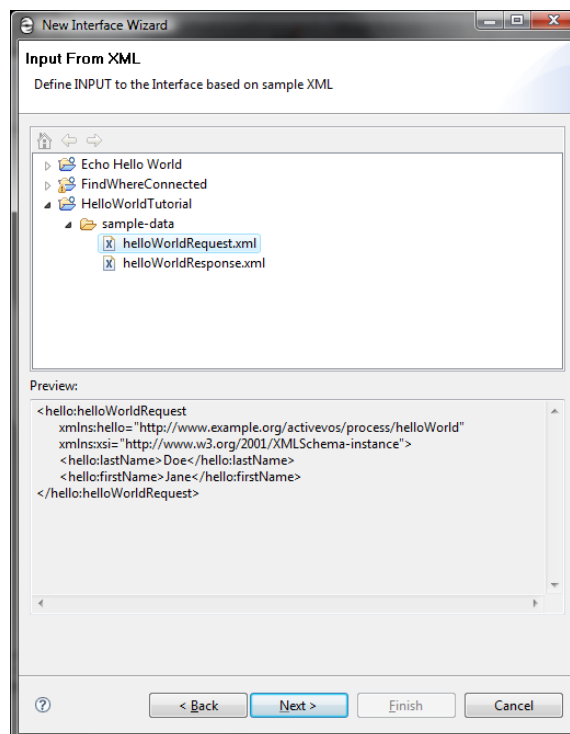
To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

To build valid receive and reply activities in a BPEL process, we need a valid WSDL. The ActiveVOS Interface Wizard can use the helloWorldRequest.xml and helloWorldResponse.xml sample data files to generate a WSDL with all the required element definitions.

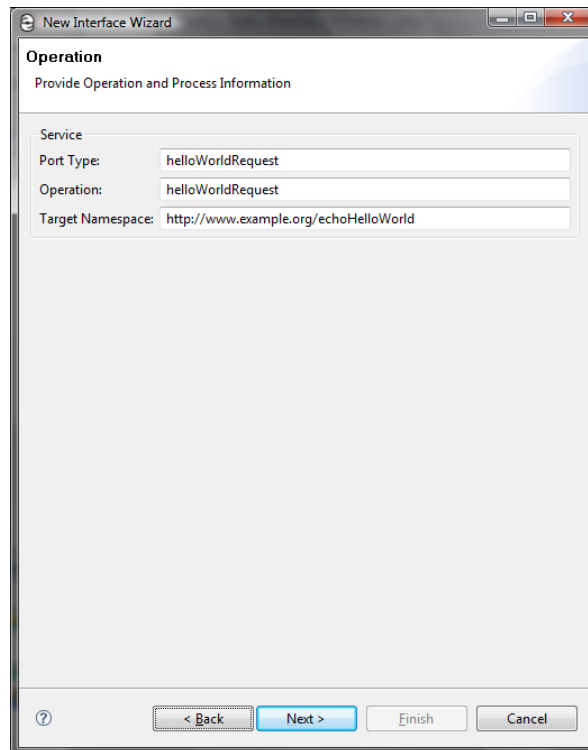
Step 1. Create the WSDL interface for the process

1. Close the XML editor and XML files, if any are open.
2. Open helloWorld.bpel.

3. In the Participants view, select the Process Service Consumers
4. Right button mouse click, and select “New Process Service Consumer...”
5. From Process Service Consumer dialog Source select “Generate Interface...”
6. On the New Interface Properties page, the Message Exchange Pattern Request-Response Operation is what we want and is selected. For the Input and Output types, select Using Sample XML. Select Next.
7. On the Input From XML page, select HelloWorldTutorial/sampledata/helloWorldRequest.xml. Select Next.

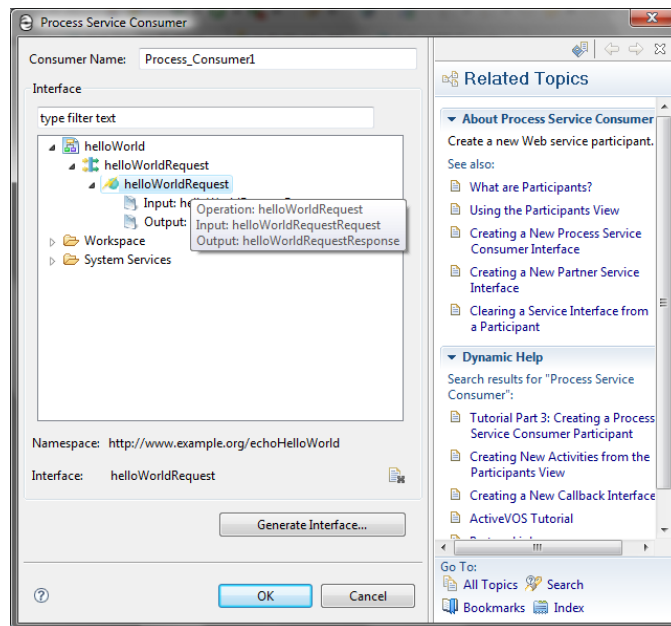


8. On the Output From XML page, select helloWorldResponse.xml. Select Next.
9. On the Operation dialog, you can leave the defaults provided
10. This page is collecting element definitions for a WSDL that will be generated. The page should look like this:



Select Next.

11. To save the WSDL file, select the parent folder HelloWorldTutorial/wsdl. In the File Name field, name the file helloWorld, (the wsdl extension is automatically added), and select Finish.



Continue to Part 5: Create the Receive and Reply Activities.

Part 5: Create the Receive and Reply Activities

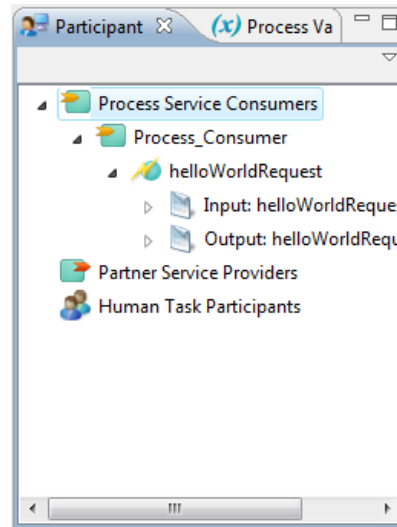
To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

After completing Part 4, we have created the WSDL interface for the process. Now we can start building the helloWorld BPEL process.

Step 1. Create the receive and reply activities

We will use the Create Activity wizard to create the receive and reply activities. There are several properties required for each activity, and the wizard generates them for us, based on our selections.

1. Open helloWorld.bpel, if it is not already open.
2. In the Participants View, expand the process consumer node as shown.



3. Drag the helloWorldRequest operation to the Process Editor Canvas to start the wizard. Your Process should look like the following illustration:



4. Select the receive activity and view its properties in the Properties view.
5. Name the receive activity ReceiveHelloWorldRequest.
6. Name the reply activity ReplyHelloWorld.
7. Save your file. An error is reported, which will be automatically fixed later.

Continue to Part 6: Map Data in an Assign Activity.

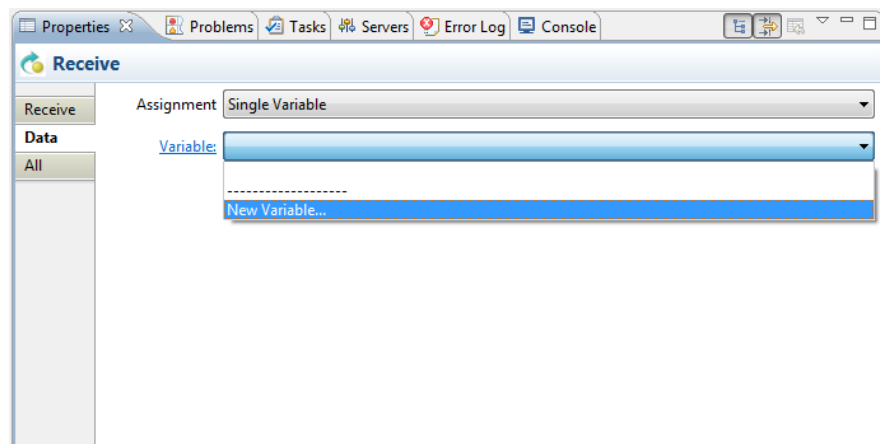
Part 6: Map Data in an Assign Activity

To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

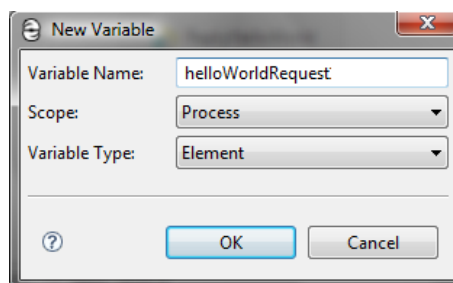
The BPEL process has two variables based on message definitions in helloWorld.wsdl. The process input data must be mapped, or copied, to an appropriate expression for the process response variable.

Step 1. Create a variable to handle the process input data

1. Select the ReceiveHelloWorldRequest, and then select the Data tab
2. Drop down the variable list box, and select New Variable...



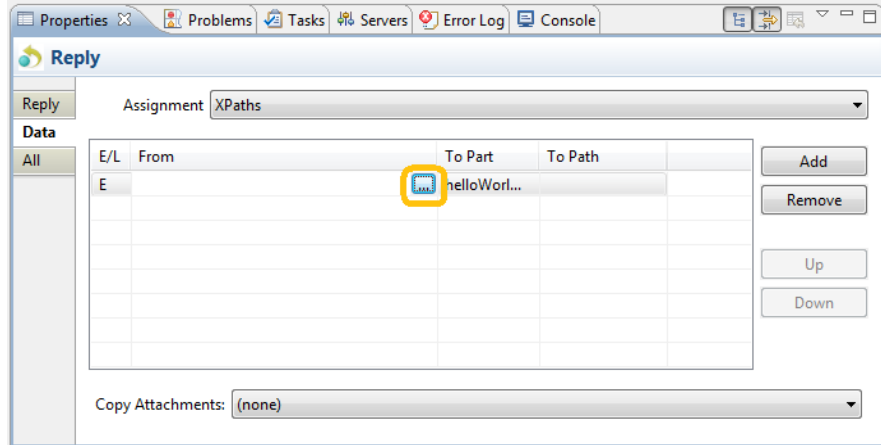
3. This will generate the following input variable



4. Select OK

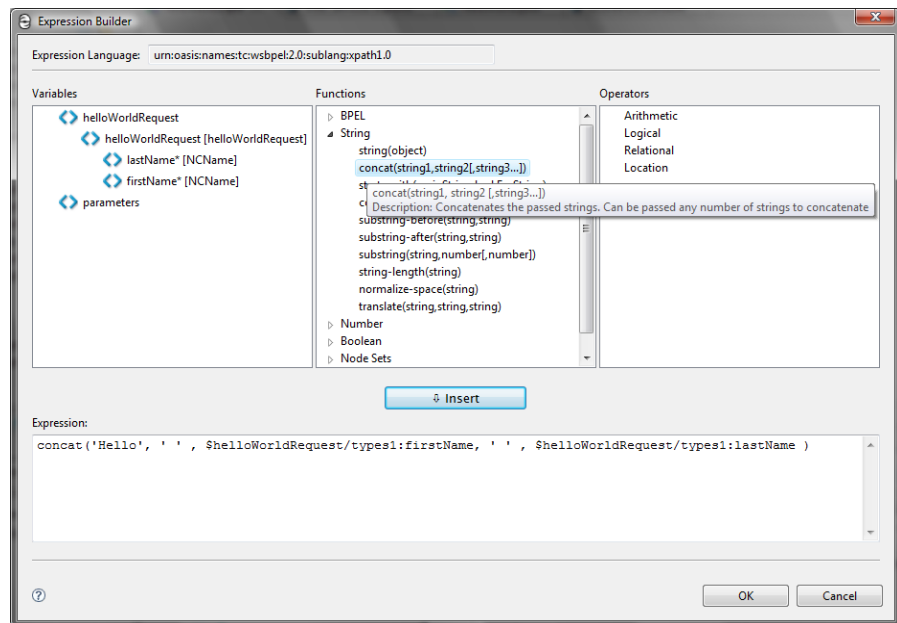
Step 2. Create an XPath expression for the output data

1. Select the ReplyHelloWorld activity's Data tab
2. Select the "Add" button
3. Create an expression using the "..." button in the From field



- Expand the helloWorldRequest variable, and double-click the lastName part of the variable as the final string. Here is the completed expression:

```
concat('Hello' , ' ', $helloWorldRequest/types1:firstName, ' ', $helloWorldRequest/types1:lastName )
```



- Save your file.

Part 7: Simulate the Process

To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

The helloWorld.bpel process is complete and ready for testing. Testing consists of using the sample data to simulate a client request coming into the process.

Step 1. Modify the sample XML files to add data values

1. In the Project Explorer, navigate to the sample-data folder and expand it.
2. Double-click helloWorldRequest.xml.
3. Replace the instances of string one and string two with the following values:

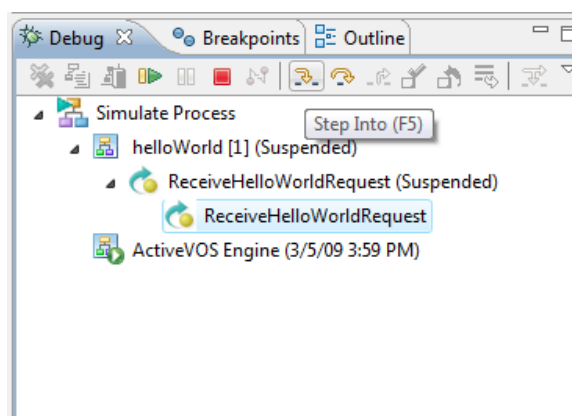
```
<hello:lastName>World</hello:lastName>
```

```
<hello:firstName>New</hello:firstName>
```

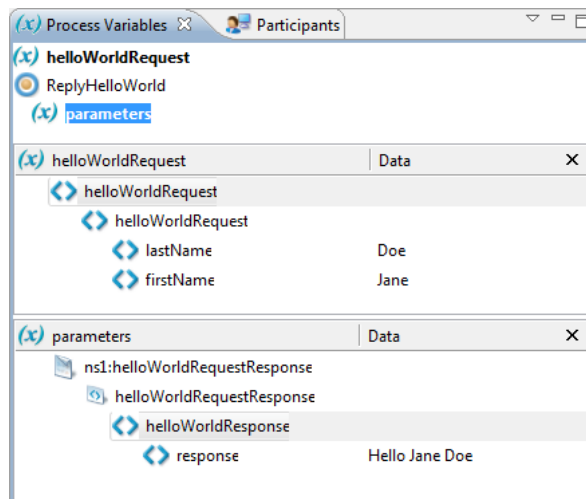
4. Save and close the file.

Step 2. Simulate the process

1. Click on a blank part of the Process Editor canvas to activate the main toolbar.
2. Select the Start Simulation icon.
3. When prompted, select the helloWorldRequest.xml sample for the input.
4. In the Debug view, select the Step Into icon to execute the receive, as shown.



5. Step into the assign activity.
6. Step into the reply.
7. In the Process Variables view, ensure your variables are open by right-mouse clicking on one of them and selecting Open All.
8. Right-mouse click on each variable and select View Data.
9. After the reply executes, your Process Variables view should look like the following.



Continue to Part 8: Deploy the Process to the Server.

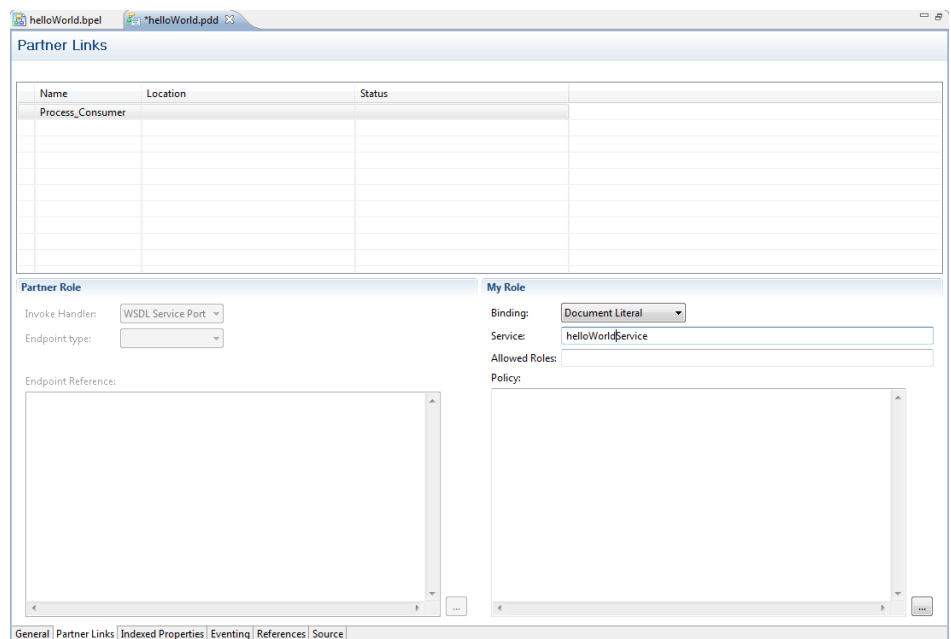
Part 8: Deploy the Process to the Server

To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

We will create a Process Deployment Descriptor (.pdd) file to provide access protocol information for the process. When the process is deployed, it becomes a Web service that clients can access.

Step 1. Create a process deployment descriptor file

1. Select File>New>Deployment Descriptor to open the New Deployment Descriptor dialog.
2. Select helloWorld.bpel, and click Next.
3. Select the deploy folder to store your deployment descriptor, and click Finish to open the PDD Editor.
4. On the General tab, do not make any changes.
5. On the Partner Links tab, select the Process_Consumer partner link.
6. Change the Service name to helloWorldService, as shown.



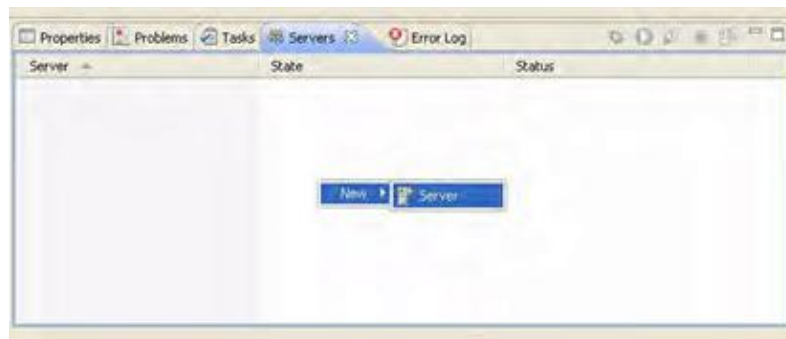
7. Leave the Allowed Roles field blank, leave the Policy tab blank.
8. Save and close helloworld.pdd.

Step 2. Starting the ActiveVOS Server

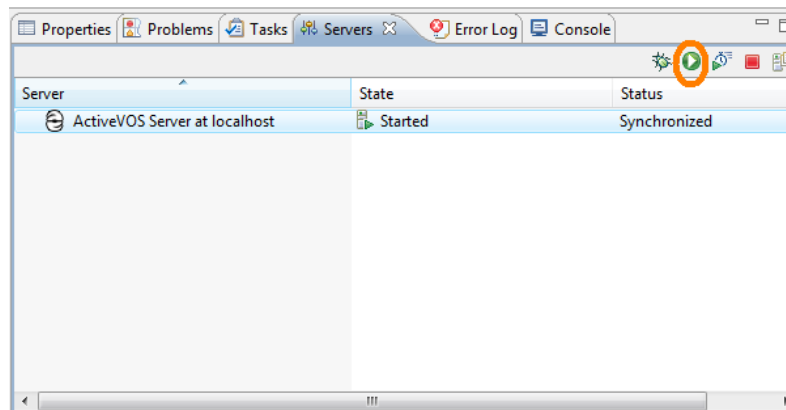
We will deploy helloworld.pdd and its resources to the server. To do so, we will start up the server.

The ActiveVOS Server consists of the ActiveVOS engine running under Apache Tomcat.

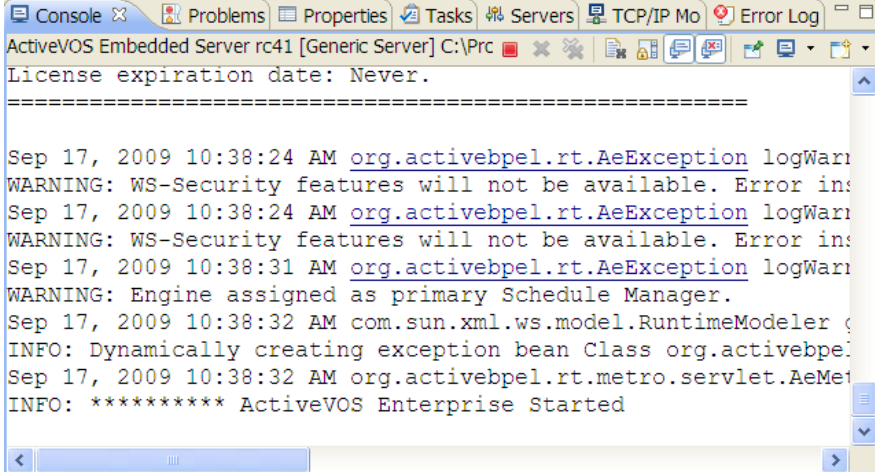
1. Select the Servers view in the lower right of the workspace.
2. Right-mouse click within the Servers view and select New>Server as shown in the illustration.



3. In the Server type list, select ActiveVOS Server, and select Finish.
4. Select the Start the Server button, as shown in the example.



5. As the server starts up, you see start up tasks scroll in the Console. Several files are deployed to the embedded server each time you start it. When the server is started, the message indicates this, as shown.

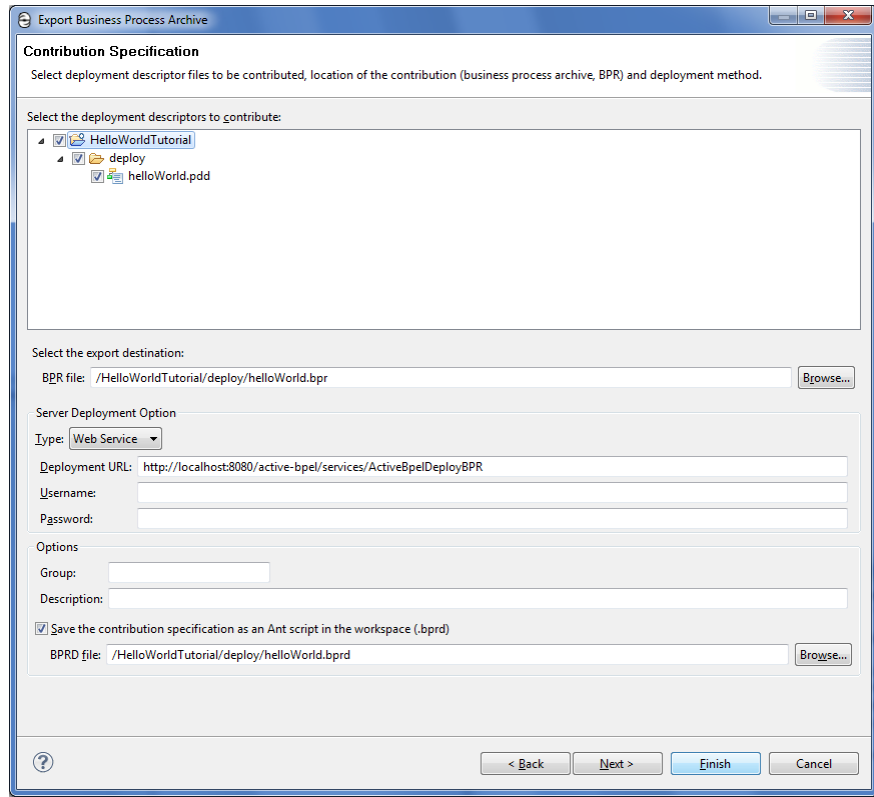


```
ActiveVOS Embedded Server rc41 [Generic Server] C:\Prc
License expiration date: Never.
=====
Sep 17, 2009 10:38:24 AM org.activebpel.rt.AeException logWarn
WARNING: WS-Security features will not be available. Error ins
Sep 17, 2009 10:38:24 AM org.activebpel.rt.AeException logWarn
WARNING: WS-Security features will not be available. Error ins
Sep 17, 2009 10:38:31 AM org.activebpel.rt.AeException logWarn
WARNING: Engine assigned as primary Schedule Manager.
Sep 17, 2009 10:38:32 AM com.sun.xml.ws.model.RuntimeModeler (
INFO: Dynamically creating exception bean Class org.activebpel
Sep 17, 2009 10:38:32 AM org.activebpel.rt.metro.servlet.AeMet
INFO: ***** ActiveVOS Enterprise Started
```

Step 3. Create a business process deployment archive

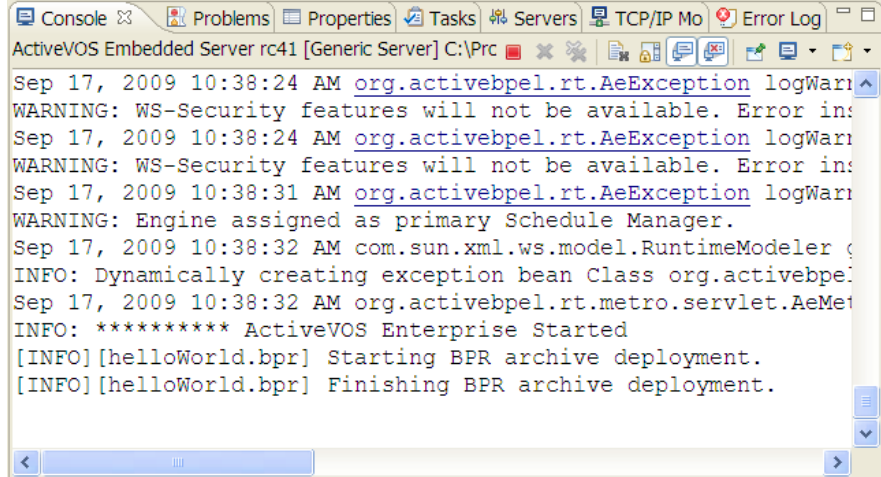
To deploy your process to the ActiveVOS Server, you will add your project as a 'contribution'. Doing so will automatically add all relevant files to an archive file (.bpr file), which is similar to a Web archive file.

1. Select the HelloWorldTutorial Project from the Project Explorer View
2. Right Mouse and select Export>Orchestration>Business Process Archive File and click Next.
3. Select the helloworld.pdd file to include in the archive, as shown in the following example.



4. For the export destination BPR file, browse to the deploy folder and name the .bpr file helloworld.bpr. Your path should be similar to the following:
designer\workspace\helloWorldTutorial\deploy\helloworld.bpr
5. Select Web Service in the Type field. The embedded engine is automatically filled in for the Deployment URL. Selecting this option automatically deploys your .bpr file to the ActiveVOS server upon completion of the export.
6. Select the check box next to Save the contribution specification..., and browse to the deploy folder.
7. Name the BPRD file helloWorld.bprd. This file is an Ant script which you can run to re-deploy the bpr file whenever you modify a .bpel or .pdd file.
8. Select Finish. A deployment details dialog shows the results.

Your BPR file has been automatically deployed to the server, as indicated by the information dialog. You can also see the results in the Console.



```
ActiveVOS Embedded Server rc41 [Generic Server] C:\Prc
Sep 17, 2009 10:38:24 AM org.activebpel.rt.AeException logWarn
WARNING: WS-Security features will not be available. Error ins
Sep 17, 2009 10:38:24 AM org.activebpel.rt.AeException logWarn
WARNING: WS-Security features will not be available. Error ins
Sep 17, 2009 10:38:31 AM org.activebpel.rt.AeException logWarn
WARNING: Engine assigned as primary Schedule Manager.
Sep 17, 2009 10:38:32 AM com.sun.xml.ws.model.RuntimeModeler (
INFO: Dynamically creating exception bean Class org.activebpe
Sep 17, 2009 10:38:32 AM org.activebpel.rt.metro.servlet.AeMet
INFO: ***** ActiveVOS Enterprise Started
[INFO][helloWorld.bpr] Starting BPR archive deployment.
[INFO][helloWorld.bpr] Finishing BPR archive deployment.
```

Continue to Part 9: Run the Process on the Server.

Part 9: Run the Process on the Server

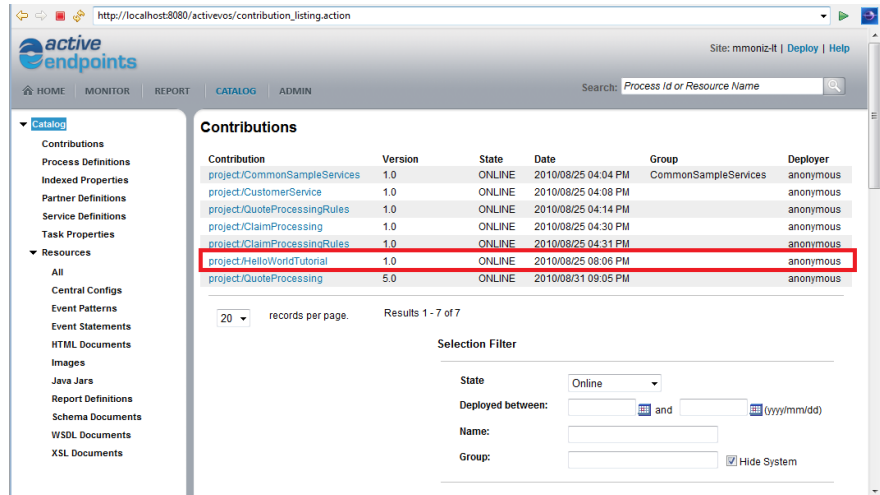
To start at the beginning of this tutorial, see Hello World: Build a Process from Scratch.

After deploying the process to the ActiveVOS Server, you can run the process by sending in a client request.

Step 1. Open a browser to view the ActiveVOS Console

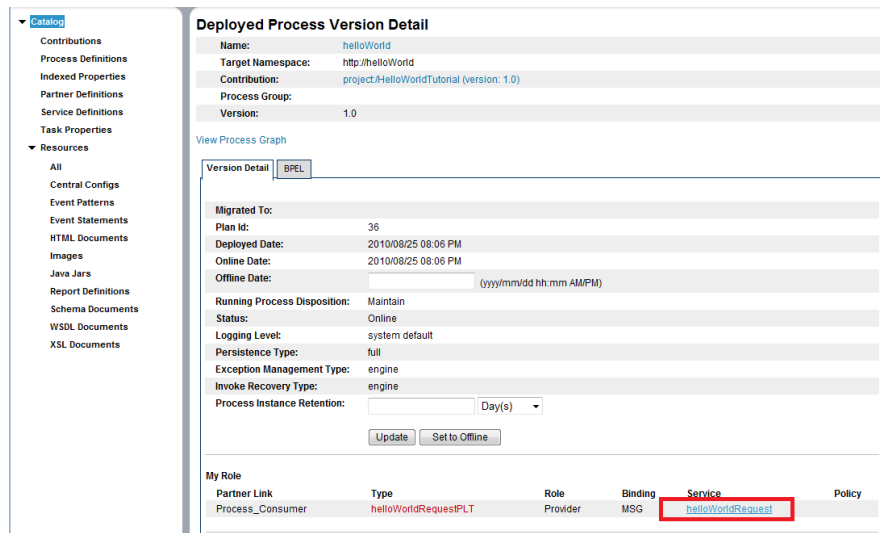
You can view deployment details for your processes in the ActiveVOS Console.

1. On the Designer menu bar, select the ActiveVOS Console icon.
2. The ActiveVOS Console opens, using a URL of <http://localhost:8080/activevos>.



You can view the deployed process. To do so, in the Server Console, select Catalog, and then select Contributions You will see the listing for the HelloWorldTutorial.

3. Select the link for HelloWorldTutorial. From here you can see all of the artifacts that were deployed with this contribution.
4. Select the link for the process HelloWorld to view the process details.

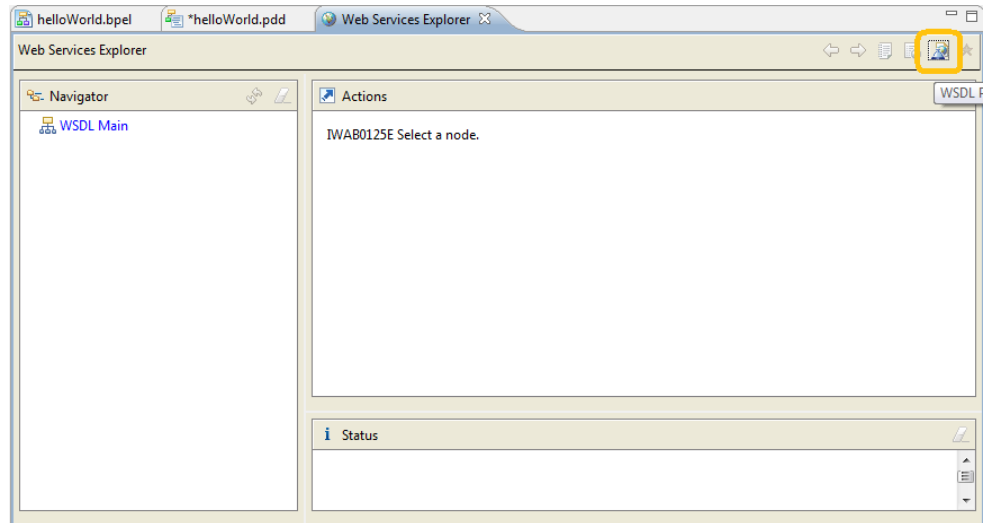


5. Right mouse on the HelloWorldRequest service, and copy the shortcut for the WSDL file to your clip-board.

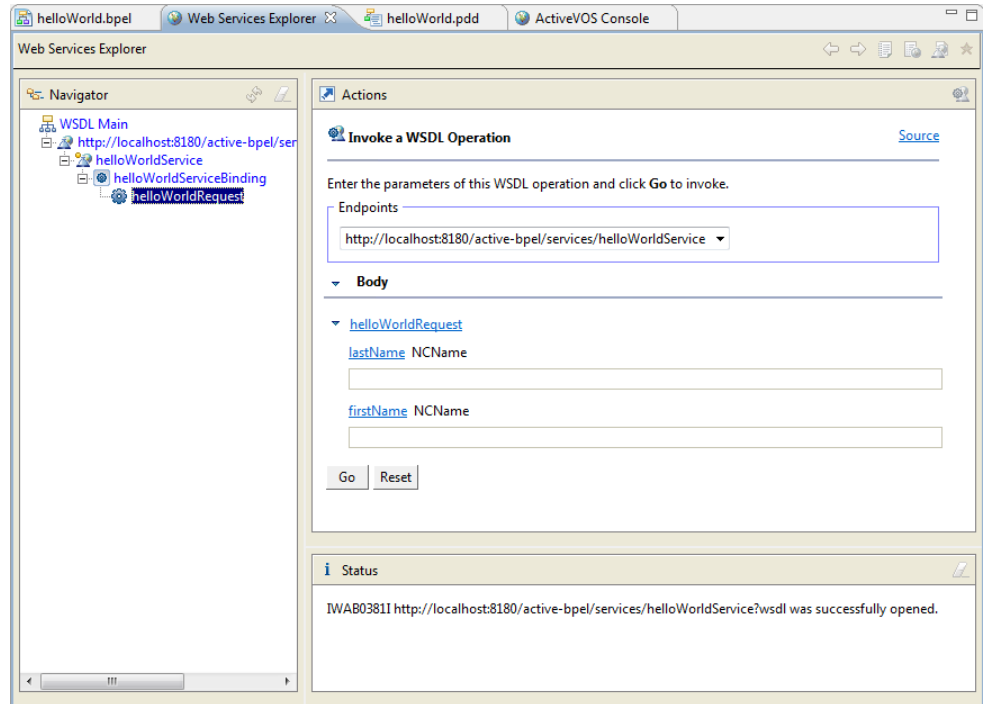
Step 2. Launch the Web Services Explorer

We need to paste the HelloWorldService URL into the Web Services Explorer in order to send a request to the process.

1. From the Designer Run menu, select Launch the Web Services Explorer.
2. Select WSDL Page as shown.



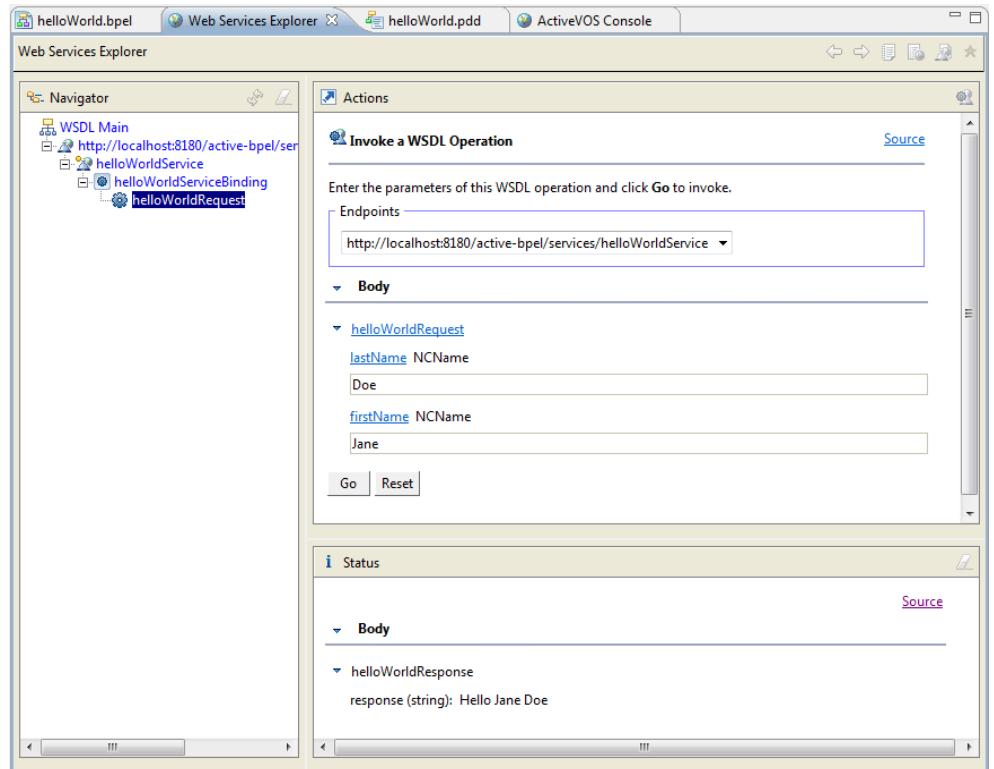
3. Select WSDL Main.
4. Paste the URL into the Web Services Explorer WSDL URL field as shown:



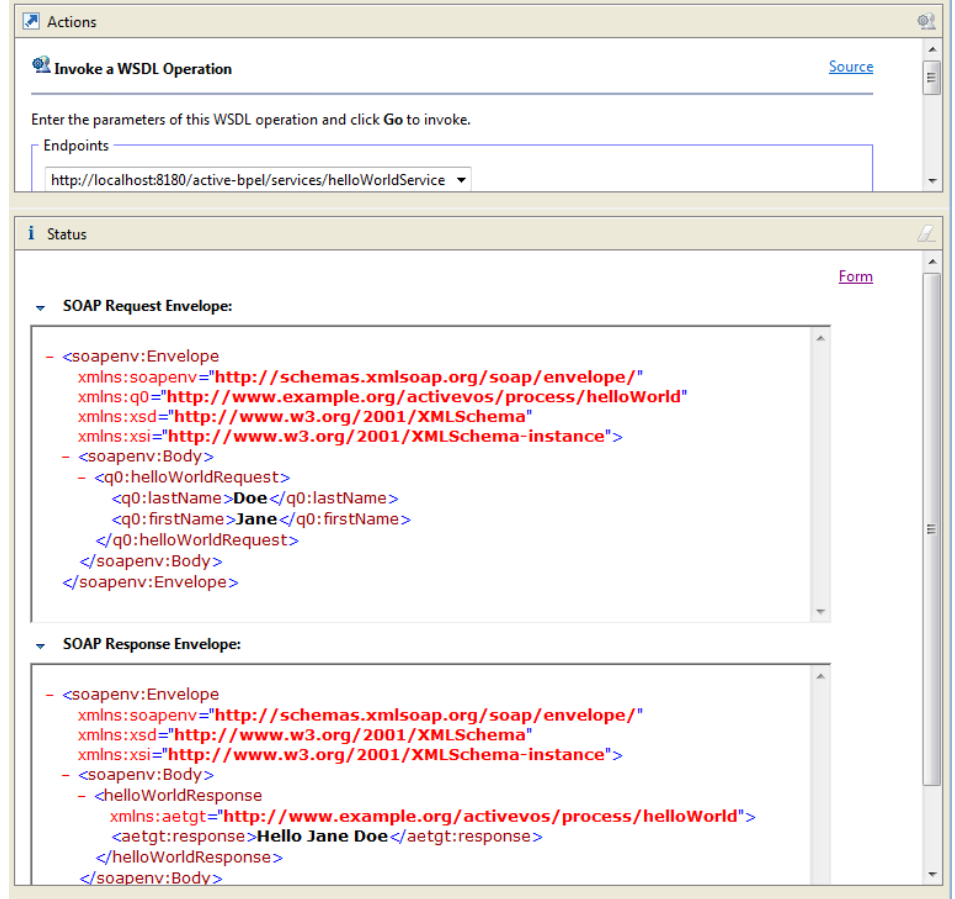
Step 3. Send a request to the process

You can make the ActiveVOS server create an instance of this process by sending a message to the service endpoint; that is, the endpoint displayed in the Web Services Explorer after you paste in the WSDL URL.

1. In the Web Services Explorer, select Go under the WSDL URL field.
2. Select the helloWorldRequest link.
3. Fill in the message parts: Last Name and First Name.
4. Select Go, which is located at the bottom of the message. The Response is shown.



5. Click the Source link, and then double-click the Status box to see the SOAP request and response sent and received.



The screenshot shows the 'Actions' tab in the Active Endpoints console. The 'Invoke a WSDL Operation' section has an 'Endpoints' dropdown menu set to 'http://localhost:8180/active-bpel/services/helloWorldService'. Below this, the 'Status' tab is active, displaying the SOAP request and response XML. The SOAP Request Envelope contains a `helloWorldRequest` with `lastName` 'Doe' and `firstName` 'Jane'. The SOAP Response Envelope contains a `helloWorldResponse` with a `response` of 'Hello Jane Doe'.

```

SOAP Request Envelope:
- <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://www.example.org/activevos/process/helloWorld"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <q0:helloWorldRequest>
  <q0:lastName>Doe</q0:lastName>
  <q0:firstName>Jane</q0:firstName>
</q0:helloWorldRequest>
</soapenv:Body>
</soapenv:Envelope>

SOAP Response Envelope:
- <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <helloWorldResponse
  xmlns:aetgt="http://www.example.org/activevos/process/helloWorld">
  <aetgt:response>Hello Jane Doe</aetgt:response>
</helloWorldResponse>
</soapenv:Body>
  
```

- In the ActiveVOS Console, select the Browser Back button to go back to the Console home page, if necessary. Select Monitor from the menu, and then select Active Processes. Notice that the process instance has completed.

Continue to Hello World Summary.

Hello World Summary

In the Hello World Tutorial, we have covered the following concepts and tasks for building a process from scratch, without starting with WSDL interfaces:

- Determine the input and output data needed for the process
- Create XML data files for the process data
- Use the Interface wizard to generate a WSDL for the process
- Build out the process, create sample data, and simulate
- Deploy the process to the server and run it

Next Steps

We recommend that you proceed to build the “Where Am I Connected From Hello World” sample.

About Active Endpoints

Active Endpoints (www.activevos.com) is the leading developer of visual orchestration systems. VOS empowers line of business project teams to create applications using services and industry standards, making their businesses more agile and effective. Active Endpoints’ ActiveVOS promotes mass adoption of SOA-enabled applications by focusing on accelerating project delivery time with a standards-based, easy to use system. Active Endpoints is headquartered in Waltham, MA with development facilities in Shelton, CT.

To find out how Active Endpoints can help your business, visit <http://www.activevos.com>, call +1 781 547 2900 and press 1 for Sales, or email us at info@activevos.com.