

## 1. BPEL Hands-On: HelloWorld

This hands-on exercise is designed to introduce you BPEL development by creating, deploying, and running a simple "HelloWorld" example.

### Prerequisites

- SOA Suite 10.1.3.1.0 installation files.
- JDeveloper Studio 10.1.3.1.0 installation files.

### Contents

- [1.1 Installing the SOA Suite basic install option](#)
- [1.2 Installing and running JDeveloper](#)
- [1.3 Creating Oracle Application Server and Integration Server Connections](#)
- [1.4 Creating the HelloWorld process](#)
- [1.5 Deploying the HelloWorld process](#)
- [1.6 Running the HelloWorld process](#)

#### 1.1 Installing the SOA Suite basic install option

1. Select **Basic Install** on the Oracle Application Server SOA Suite 10.1.3.1.0 Installation window.

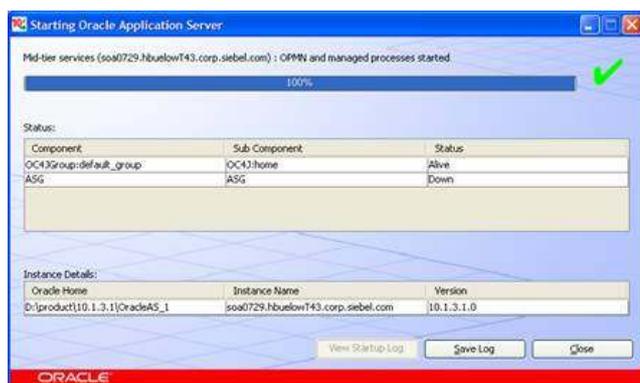
**Note:** If you want to use Oracle BPEL Process Manager samples, enter **welcome1** in the AS Administrator Password field for the **oc4jadmin** user. If you enter a different password, some samples require modification in order to work.

**Note:** Always **Uninstall** an existing version of Oracle SOA Suite before installing a new one or you may have to modify samples for non-default port numbers in order for them to work.

2. Click **Install**.

3. Click **Exit** when installation completes.

4. Select **Start > All Programs > Oracle - Oracle\_Home > Start SOA Suite** to start the Oracle SOA Suite and all the parts of the default group, including the BPEL Process Manager Server, Enterprise Service Bus Server, Rule Author, Web Services Manager, and etc. When the server has started, you will see the status **Alive** for the default group in the window:



5. Click Close to close the window.

**Note:** If you see a status of **Init**, simply **Close** the window and select **Start > All Programs > Oracle - Oracle\_Home > Start SOA Suite** again to get an updated status until you see **Alive**. This doesn't start another instance of the server.

**Note:** Ignore the status line for ASG.

#### 1.2 Installing and starting Oracle JDeveloper

1. Unzip Oracle JDeveloper Studio 10.1.3.1.0 into a location on your host.

2. Start Oracle JDeveloper by double-clicking on `JDev_Oracle_Home\jdeveloper.exe` or create a shortcut for this program and start it from your shortcut.

### 1.3 Creating Oracle Application Server and Integration Server Connections

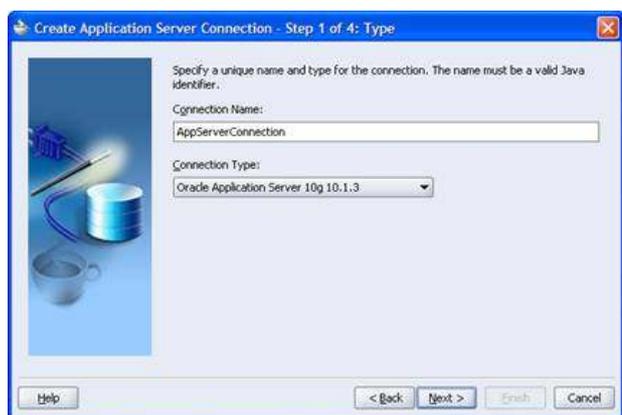
1. Select **Connection Navigator** from the **View** main menu in Oracle JDeveloper.
2. Right-click **Application Server**.



3. Select **New Application Server Connection**. The Create Application Server Connection wizard starts.



4. Click **Next** on the Welcome page.



5. Provide a meaningful name for connecting to the application server. Remember this name, as you will select it when creating an integration server connection in Step 19 below.
6. Select **Oracle Application Server 10g 10.1.3** from the Connection Type list.
7. Click **Next**.



8. Enter the following details:

| Field           | Value  |
|-----------------|--|
| Username        | oc4jadmin  |
| Password        | Enter the same administrator password you entered when installing Oracle Application Server SOA Suite. |
| Deploy Password | Select this check box; otherwise, you must authenticate each time you deploy a process.                |

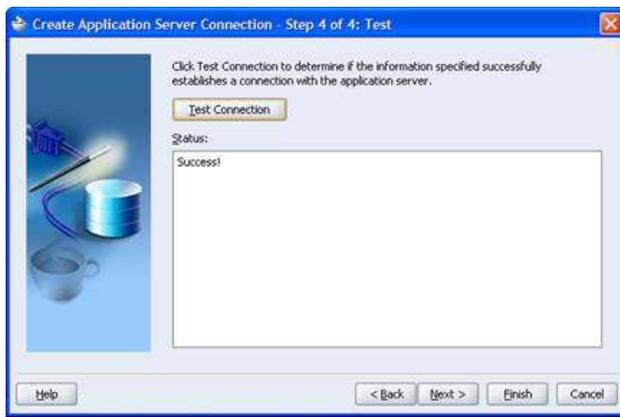
9. Click **Next**.



10. Enter the following details:

| Field                    | Value   |
|--------------------------|---|
| Host Name                | localhost   |
| OPMN Port                | Enter the same port value used by the opmn.requestport. This is likely port 6003. |
| Group/OC4J Instance Name | home  |
| Single Instance          | Select this radio button.   |
| Group                    | Leave this radio button unselected.   |

11. Click **Next**.



12. Test the connection by clicking **Test Connection**. If the connection is successful, the following message appears:

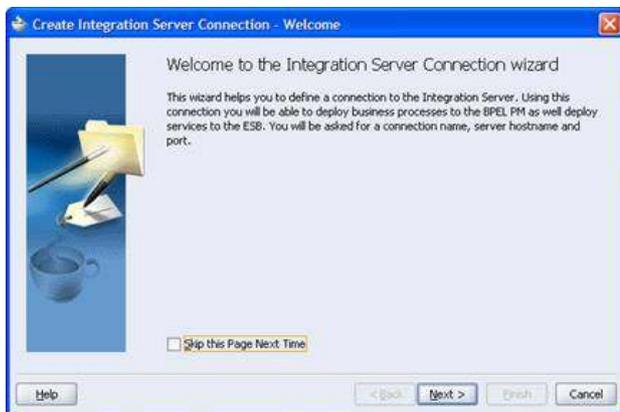
*Success!*

13. Click **Finish**.

14. Right-click **Integration Server** in the Connection Navigator.



15. Select **New Integration Server Connection**. The Create Integration Server Connection wizard starts. You use this integration server connection to deploy all your BPEL processes and to register your ESB service flows.

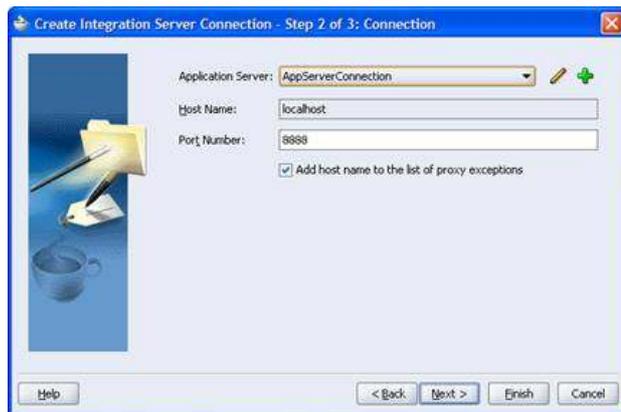


16. Click **Next** on the Welcome page.



17. Provide a meaningful name for connecting to the server.

18. Click **Next**.



19. Select the application server connection name you specified in Step 5 from the Application Server list.

20. Enter the following details:

| Field   | Value   |
|---|---|
| Host Name                                     | localhost   |
| Port  | Enter the same port value used by the http.port property. This is likely port 8888.   |
| Add host name to the list of proxy exceptions | Select this check box to add this host automatically to the list of proxy exceptions shown in the Exceptions field under Tools > Preferences > Web Browser and Proxy. The first time it is started, Oracle JDeveloper by default picks up these proxy exceptions that are set for your browser. |

21. Click **Next**.



22. Test the connection by clicking Test Connection. If the connection is successful, the following message appears:

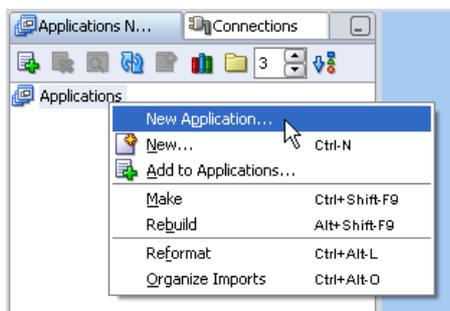
*Application Server: OK*  
*BPEL Process Manager Server: OK*  
*ESB Server: OK*

23. Click **Finish**.

Congratulations! You have now finished installing and setting up SOA Suite and JDeveloper and you are ready to build your BPEL process. Continue with the steps below to create and run the Hello World BPEL process.

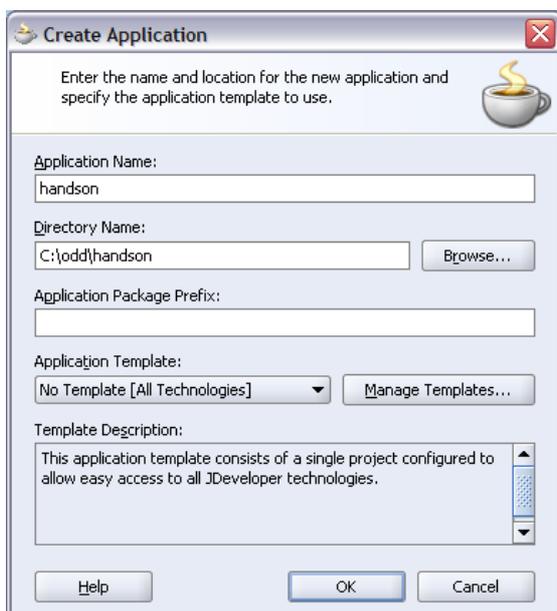
## 1.4 Creating the HelloWorld process

1. Open JDeveloper.
2. In the Applications Navigator, right-click **Applications** and select **New Application....**

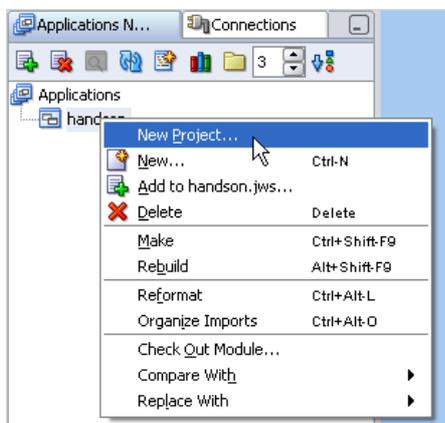


3. Set the following fields, leaving all other fields as they are:

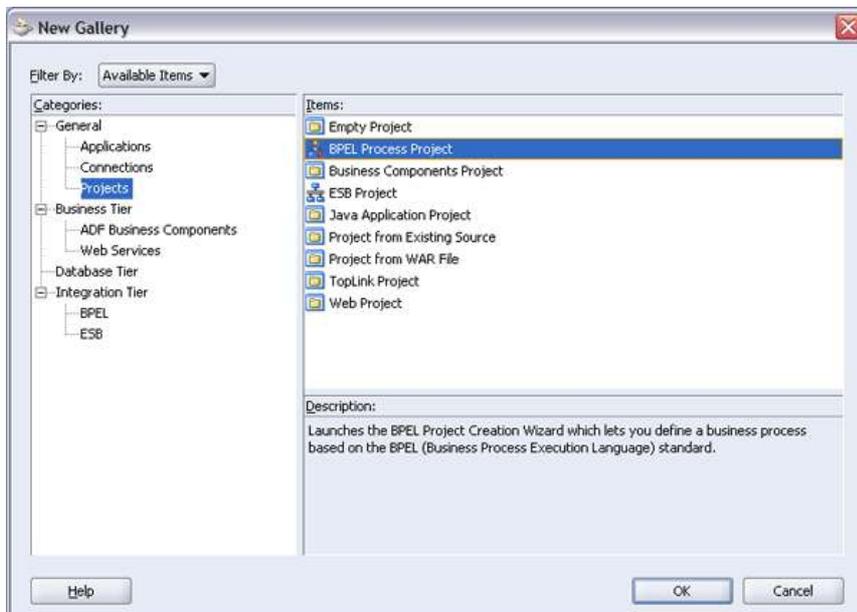
| Field            | Value   |
|------------------|---|
| Application Name | handson   |
| Directory Name   | C:\odd\handson or pick a directory of your choice. (You'll want to make sure the final directory name in the path is the same as the Application Name, though.) |



4. Select **OK**.
5. In the **Create Project** window that appears, select **Cancel**. That would create a generic project, but we want a BPEL project. If you inadvertently created a new project, it won't cause any harm. Just ignore it.
6. Right-click **HelloWorld** and select **New Project...**. The **New Gallery** window will appear.

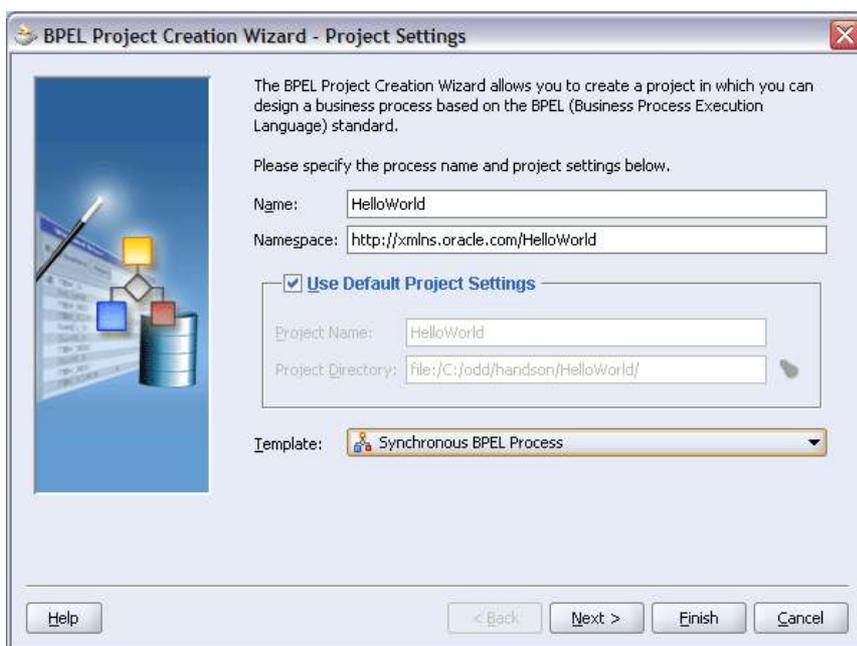


7. In the **Categories** section, select **General > Projects**. In the **Items** section, select **BPEL Process Project** and press **OK**.



8. Set the following fields, leaving all other fields as they are:

| Field    | Value                    |
|----------|--------------------------|
| Name     | handson                  |
| Template | Synchronous BPEL Process |



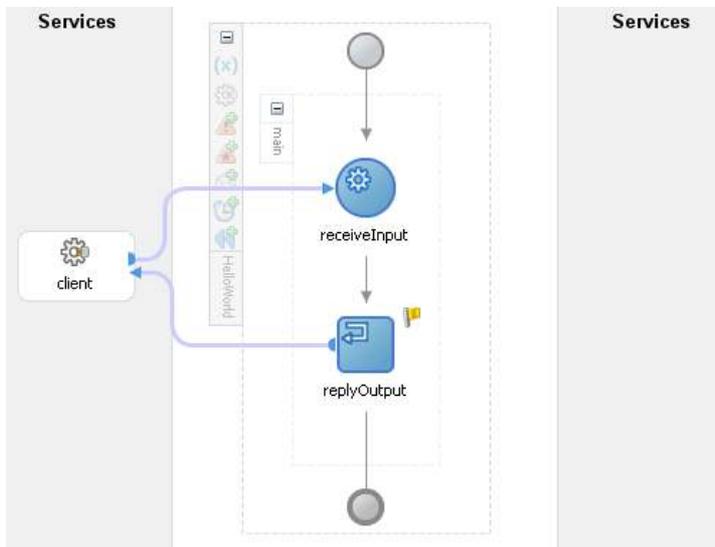
9. Select **Next**.

10. The second page of this wizard lets you specify the input and output schema definitions of your BPEL process. In this case we will take the default, which is a single string for input and a single string for output.

Select **Finish** to close the wizard and create your project.

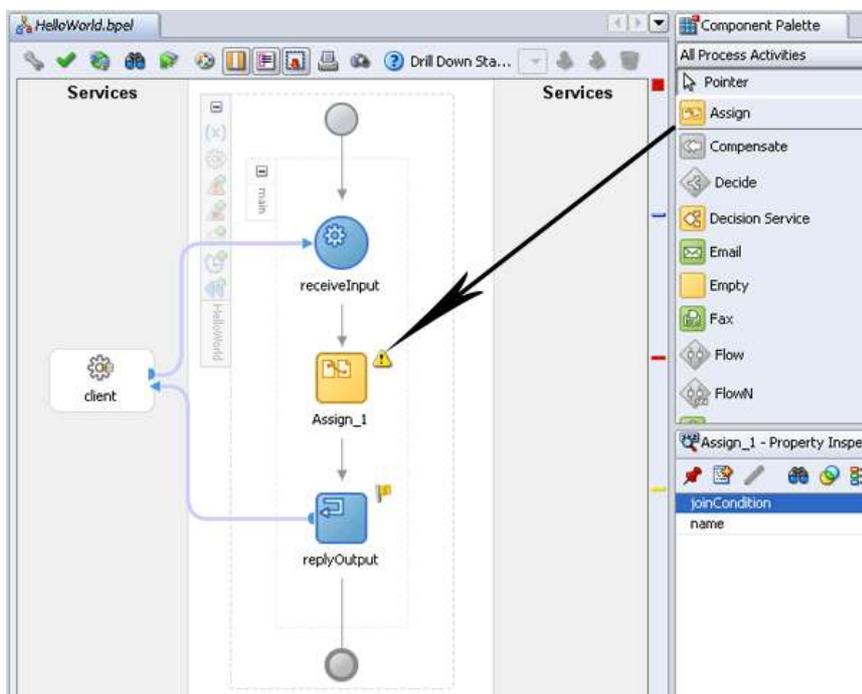
11. A skeleton BPEL process has been created for you. The **client** partnerlink in the Services "swimlane" represents the client calling this process. The **receiveInput** receive activity is the start of our process, and accepts the input from the client invoking this process. The **replyOutput** reply activity is the end of the process and returns the output from the process to the client invoking it. (This is optional, BPEL processes are not obliged to return data.)

All of our changes will be between these start and end activities.

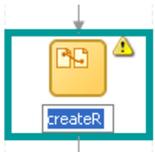


12. This application will take some text input, and prepend "Hello " to it, and return the result.

In the Component Palette, select **Process Activities** from the poplist. Drag-and-drop an assign activity onto the diagram view, between **receiveInput** and **replyOutput**.

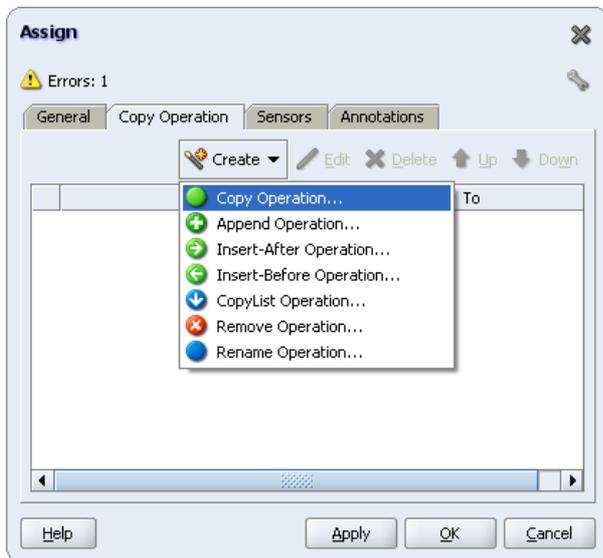


13. Double-click the text under the assign activity (*not* the icon itself) where it says "Assign\_1" to edit it.  
 14. Enter "createReturnString" and press Enter to rename this activity.



15. Double-click the assign activity to open the Assign dialog.  
 16. We want to assign the input to the process to the return result, after concatenating "Hello " to the front.

On the **Copy Operation** tab, select **Copy Operation...** from the **Create** poplist.



17. Change the From **Type** poplist to Expression. This lets us enter an XPath expression.

18. Enter the following expression:

```
concat("Hello ",
bpws:getVariableData('inputVariable','payload','/client:HelloWorldProcessRequest/client:input'))
```

That is all one line, even though it may appear wrapped here.

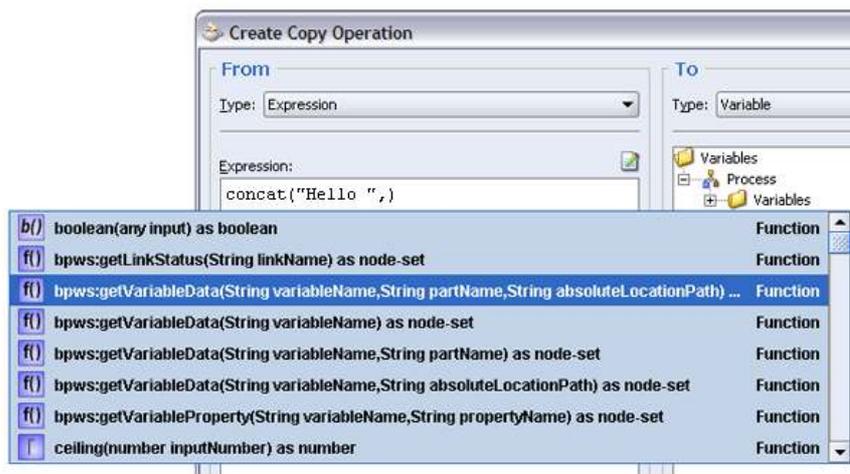
**Hint:** You can use JDeveloper's auto-completion feature instead of typing the entire text by using Ctrl-Space to bring up a list of valid methods and variable names. For example, put the cursor in the **Expression** field:

- Type "con".
- Press Ctrl-Space.
- Select the concat() method.
- Type:

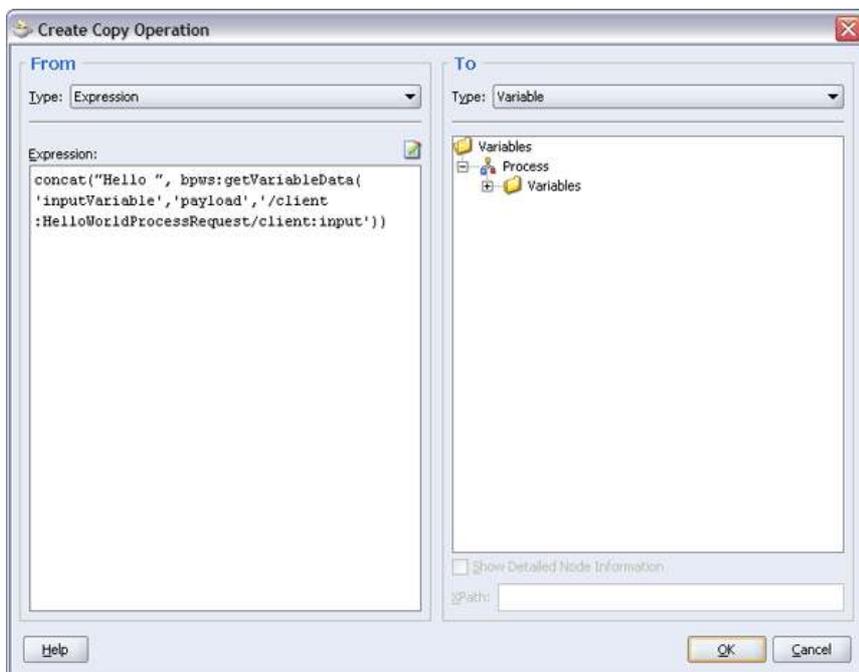
"Hello ",

Make sure you include the quotes and the comma. If the auto-complete appears, you can press Escape to hide it.

- Press Ctrl-Space again (if the list automatically reappears, then you don't need to press Ctrl-Space.)
- Select the first reference to the bpws:getVariableData() method.

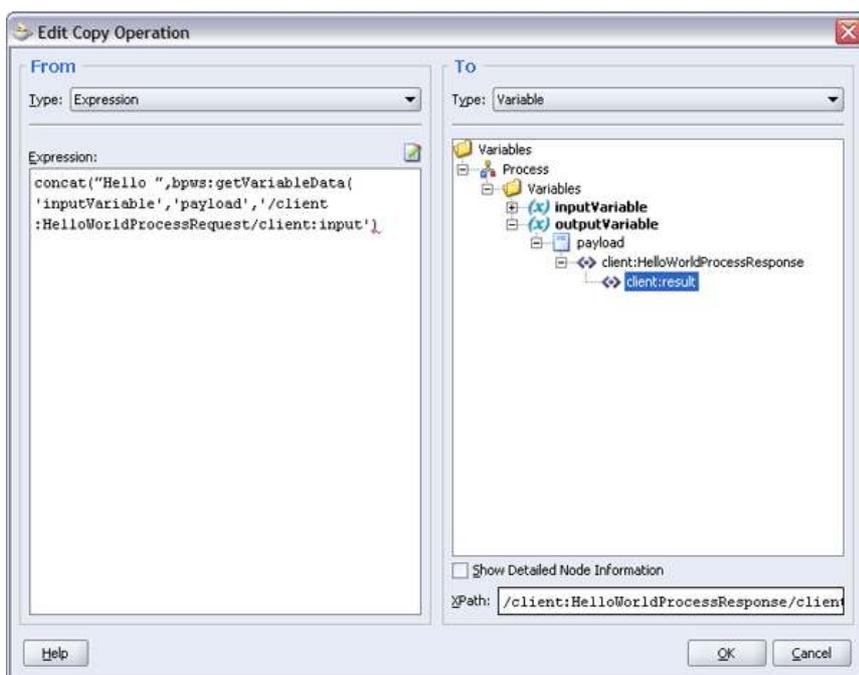


- If the list doesn't reappear, press Ctrl-Space.
- Select **inputVariable**.
- Keep repeating this, pressing Ctrl-Space (if the list doesn't automatically reappear) and select the following values, one after the other: **payload**, **/client:HelloWorldProcessRequest**, and **client:input**.



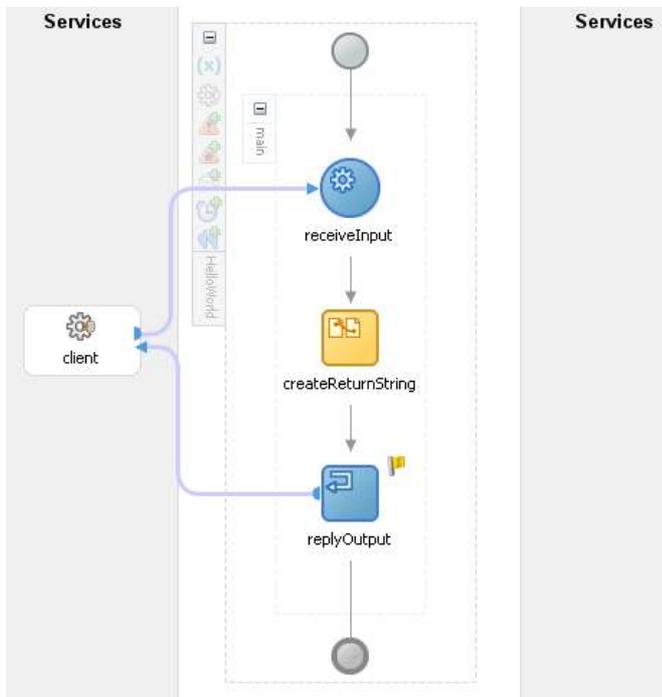
This is an XPath expression that says concat "Hello " to the contents of the input variable to this process. This may seem confusing now, but it will become second nature after building a few BPEL processes.

19. On to the **To** side of the dialog, select **Variables > Process > Variables > outputVariable > payload > client:HelloWorldProcessResponse > client:result**.



In summary, this dialog is copying the result of the expression from the **From** side into the variable specified in the **To** side. In this case, the target variable is the variable this process returns to the client that invoked it.

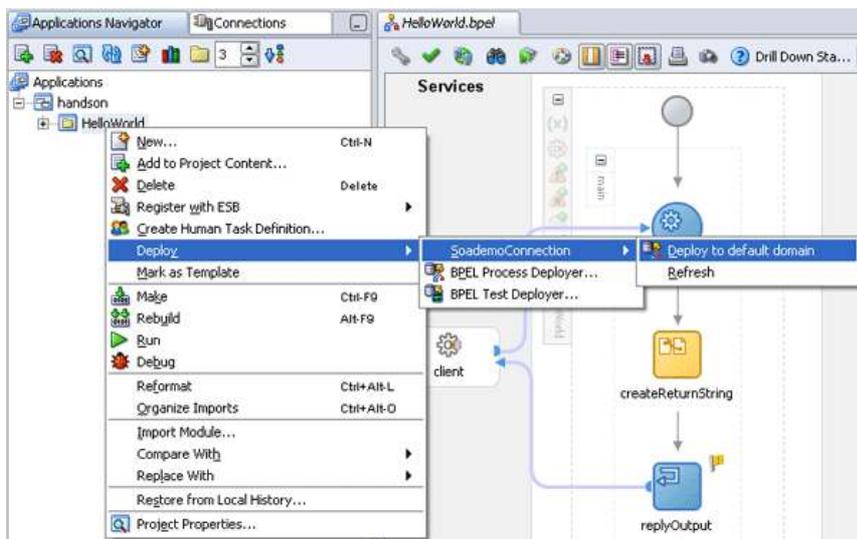
20. Press **OK** to close the **Edit Copy Operation** dialog.
21. Press **OK** to close the **Assign** dialog. Note that sometimes the **OK** button may not be visible, and you'll need to scroll down to see it. Don't press the X button because that will close the dialog without saving.
22. Select **File > Save**.
23. You have now created a BPEL process that is ready to be deployed and run.



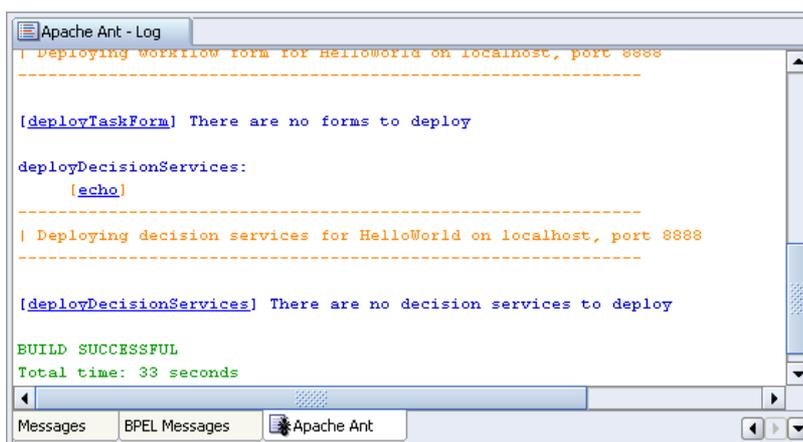
## 1.5 Deploying the HelloWorld process

You don't run a BPEL process from within JDeveloper, but rather deploy it to the server and run it from there.

1. Right-click the **HelloWorld** project name and select **Deploy > Your connection name > Deploy to default domain**.



2. The Messages pane will show you progress and let you know when deployment is complete in the Apache Ant sub-tab. The first time after starting the server it can take longer while everything initializes for the first time.



- Once deployed, you can now run your BPEL process.

## 1.6 Running the HelloWorld process

The BPEL Control is used to manage, monitor, and administer your BPEL server and the processes deployed. You can also run your BPEL process from there. In a real development environment, you would have some kind of screen or script to launch the BPEL process, but until you've created those, you can launch a BPEL process from the BPEL Control.

- Open the BPEL Control using one of the following methods. If prompted to login, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.
  - From the Windows Start Menu, select **All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > BPEL Control**.
  - From the SOA Welcome Page, select **BPEL Control** (the link is on the right-hand side).
  - Enter the URL directly into your browser: `http://<host>:<port>/BPELConsole`. For example:  
`http://localhost:8888/BPELConsole`

- The **Dashboard** tab shows all of the BPEL processes deployed to this server. You should see **HelloWorld** in the list.

| Deployed BPEL Processes |          | In-Flight BPEL Process Instances |                 |  |
|-------------------------|----------|----------------------------------|-----------------|--|
| Name                    | Instance | BPEL Process                     | Last Modified ↑ |  |
| DHLShipment             |          |                                  |                 |  |
| HelloWorld              |          |                                  |                 |  |
| SOAOrderBooking         |          |                                  |                 |  |
| SelectManufacturer      |          |                                  |                 |  |
| TaskActionHandler       |          |                                  |                 |  |
| TaskManager             |          |                                  |                 |  |

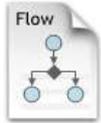
- Select **HelloWorld** to open the **BPEL Process** administration screen. You can do administration activities related to this process, inspect its deployment descriptors, WSDL file, etc. By default, the **Initiate** sub-tab comes up which lets you launch this process.
- In the **Initiate** sub-tab you can specify many different options, including setting the WS-Security and WS-Addressing headers. In this case, we only want to set the input to the process and run it.

Expand the **Payload** section, if it's not already, and enter your name in the **input** field.

- Press **Post XML Message**. After a few moments, the output from the process will be displayed. It will say "Hello " plus the name you entered.

```
<HelloWorldProcessResponsehttp://xmlns.oracle.com/HelloWorld>
<resulthttp://xmlns.oracle.com/HelloWorld>Hello Dupree</result>
</HelloWorldProcessResponse>
```

- Click **Visual Flow** see a visual flow of the process instance.



[Visual Flow](#)

7. The visual flow looks similar to the BPEL Designer. A BPEL process could have different paths, but at runtime, it will be resolved and you can see that path your process took. In this case, we have a simple process and there is only the one path.
8. Click on the **receiveInput** receive activity. This opens a popup window that shows the XML data received as input from the client that invoked this process. As you can see, name you entered when you launched the process is there in the `<input>` element.



9. Close the XML audit trail window.
  10. Click on the **createReturnString** assign activity. In the audit trail popup window, you can see the results of the assign copy operation in the `<result>` element: "Hello " is concatenated with your input.
  11. Close the XML audit trail window.
  12. Click on the **replyOutput** reply activity to see the output that was returned to the client. When you launched the process you were the client which is why you saw the output in step 5, above.
- Congratulations, you just built, deployed, and ran a HelloWorld BPEL process!