

SOA Suite 11g Technology Preview 3 – Update 1

Installation and quick start instructions

Recommended system specs: Windows XP or Linux with at least 2GB memory and 2GB hard disk space.

These instructions use the Windows nomenclature. Linux users should replace the commands with the appropriate Linux equivalent.

Contents

Update 1.....	1
Pre-install setup	1
Installation	2
Application Deployment	5
Creating the SOAHelloApp Application.....	7
Appendix A: Sample log screens	15

Update 1

This update includes a new version of the SOA Suite design- and runtime. Installing this requires you to start JDeveloper two times with the update step in between before you begin the SOA Suite database schema setup. If you do not start JDeveloper a second time before running the schema setup, you will not have updated scripts. Please follow the directions carefully.

Warning: applications that you built using a previous version of the technology preview may no longer work. Do not upgrade until you are prepared to rebuild your applications.

Pre-install setup

1. Database: Oracle Enterprise Database 10gR2 or Oracle Express Edition 10gR2
 - a. Oracle Express Edition 10.2.0.1 (Oracle XE) is a great choice for this because it has a small size and is easy to install. If you use the enterprise database, check your version. Oracle Enterprise Database 10gR1 (10.1.x) and 10.2.0.1 are NOT supported - you must apply the 10.2.0.3 patch for the Oracle Enterprise 10g 10.2.0.1 database.
 - b. Take note of the sys password and connection URL for later steps (e.g., stbbl20:1521:XE)
2. JDeveloper
 - a. Go to the Technical Preview 3 location and download **jdevstudio1111.zip**. This includes the Windows JDK used by JDeveloper. Linux users will download jdevstudiobase1111.zip

and get the Linux JDK 1.5.0_11 from the [Sun location](#). When running the first time, JDeveloper base version asks for the location of the Linux JDK.

3. SOA Suite update
 - a. Go to the SOA Suite 11g Technical Preview 3 location and download the SOA Suite design time and runtime update file: **pcbpel_bundle.zip**.
4. Set environment variables
 - a. Setting these environment variables is **REQUIRED** to work around bugs. You can use any location or choose these sample locations. Take note of these directories as the instructions below refer to them during installation. Set these using **Control Panel -> System -> Advanced -> Environment Variables**

```
set JOH=c:\jdevt3 - your JDeveloper unzip location – no spaces!
set JDEV_USER_DIR=c:\jdevinstance - no spaces!
set JAVA_HOME=c:\java - your Java SE 5 JDK if using base version – no spaces!
```

Installation

Installation has 3 parts:

- a) JDeveloper,
 - b) SOA infrastructure database schema, and
 - c) SOA infrastructure install.
5. Unzip the JDeveloper zip file from step 2 to **JOH** – making sure there are no spaces in this path
 - a. Open a command prompt and do the following


```
> mkdir %JOH%
> cd %JOH%
> unzip jdevstudio1111.zip
```
 6. Start JDeveloper
 - a. Start JDeveloper (%JOH%\jdev\bin\jdev.exe). This starts JDeveloper with a console window which shows informational and error messages. See Appendix A for a sample of this log window.
 - b. The first time you start JDeveloper, you must respond to some configuration prompts:
 - i. If you are using the base version, you are prompted for the J2SE JDK. Browse to and select java.exe in the %JAVA_HOME%\bin directory (no spaces in path).
 - ii. When you are prompted for the user type you should choose **Default Role**.
 - iii. When you are asked if you want to migrate from a previous version of JDeveloper you should select **No**.
 - c. Once you have JDeveloper running, you may be prompted about new updates. DO NOT update, except when explicitly told to do so in these instructions or tutorials.
 7. Update JDeveloper with the new SOA Suite design- and runtime.
 - a. In JDeveloper, select **Help/Check for Updates**.
 - b. Select **Next** and then select the radio button for **Install From Local File** and navigate to where you stored the **pcbpel_bundle.zip** in step 3.

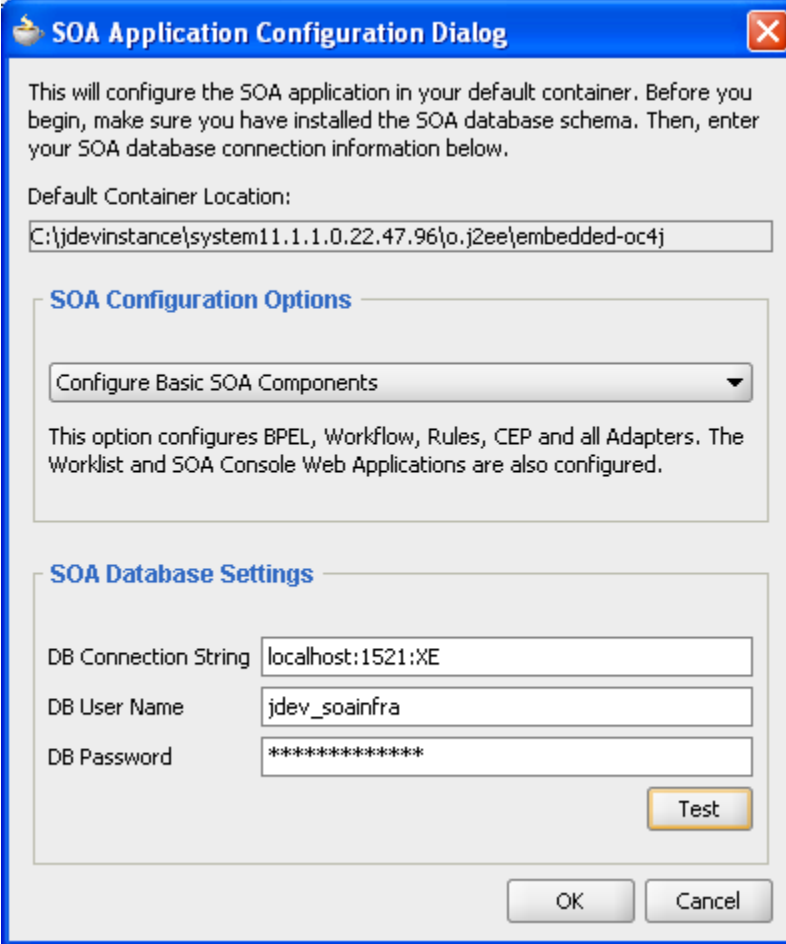
- c. Select **Next** and **Finish**.
 - d. When you are asked to restart JDeveloper, select **Cancel** and then exit JDeveloper using the **File/Exit** menu item. This is so you can restart JDeveloper yourself using the console.
 - e. Restart JDeveloper using the method in step 6. You **MUST** restart JDeveloper to process the new `pcbpel_bundle.zip` before moving on to the next step.
8. Create the SOA infrastructure database schemas
- After JDeveloper has started the second time you have the updated scripts to set up the database schemas. Open a command window to run these sql scripts. In the commands below, it is assumed your database bin directory is on your path (the default) to access the sqlplus tool.
- a. If you are updating from a previous version or reinstalling, it is recommended that you drop the existing user and schema first. Use the following to remove the user you created last time (either `adrs_soainfra` or `jdev_soainfra`), replacing `<sys-passwd>` with your password for the system user. This operation will take a few minutes to complete.


```
> sqlplus system/<sys-passwd>
SQL> drop user jdev_soainfra cascade; [or adrs_soainfra]
SQL> exit
```
 - b. Now create the user as follows, replacing `<sys-passwd>` with your password for the sys user and `<jdev-passwd>` with your own password (remember your password!). (If you are familiar with the previous version you will notice the user name change. This is because we no longer refer to this configuration as *adrs*.)


```
> cd %JOH%\rcu\integration\soainfra\sql
> sqlplus sys/<sys-passwd> as sysdba
@createuser_soainfra_oracle_all.sql jdev_soainfra <jdev-passwd>
SYSTEM TEMP
```
 - c. Next create the schema using your new user


```
> sqlplus jdev_soainfra/<jdev-passwd>
@createschema_soainfra_oracle_all.sql jdev_soainfra
```
9. Configure SOA to install the Integrated OC4J Server
- a. Start JDeveloper if it is not already started
 - b. Some users will need to set a proxy if the machine and the network configuration require a proxy. In JDeveloper, select **Tools -> Preferences, Web Browser and Proxy**, enter the proxy information, and select **Test Proxy**.
 - c. In JDeveloper, select **Tools -> Preferences, Run**, and then select **Integrated OC4J Server**.
 - d. Next, select **Tools -> Configure SOA...**
 - i. Complete the dialog for your configuration specifying the database location and the SOA schema user name and password you used in Step 8.
 - ii. The Basic configuration option is recommended.
 - iii. Be sure to use the Test button to test the database connection.

- iv. Select OK to begin the configuration



The image shows a Windows-style dialog box titled "SOA Application Configuration Dialog". It contains the following text and fields:

This will configure the SOA application in your default container. Before you begin, make sure you have installed the SOA database schema. Then, enter your SOA database connection information below.

Default Container Location:

SOA Configuration Options

This option configures BPEL, Workflow, Rules, CEP and all Adapters. The Worklist and SOA Console Web Applications are also configured.

SOA Database Settings

DB Connection String
 DB User Name
 DB Password

Buttons: Test, OK, Cancel

- e. This will take 8-10 minutes to run and when finished will end with "**Build Successful.**" The operation is disk intensive so for best results leave your machine alone during this time – you won't be able to do much else anyway.
- i. If this step seems to be taking a long time or fails, look in the log (will be at %JDEV_USER_DIR%\system11.1.1.0.22.47.94\o.j2ee\embedded-oc4j\log\startsoa.log) to see the error. The most usual issues are not setting the environment variables or with the database connection – correct any configuration issues and try the Configure SOA command again using the instructions in step f below.
 - ii. Even if you see errors in the log, if you get "Build Successful" and the SOA infrastructure was configured, then you can continue to Step 8. See Appendix A for a sample of this log window.
- f. If Configure SOA fails with "**Configuration of SOA Infrastructure has FAILED**"
- i. Look in the log (will be at %JDEV_USER_DIR%\system11.1.1.0.22.47.94\o.j2ee\embedded-oc4j\log\startsoa.log) for errors and correct any configuration problems. Check the release notes for help.

- ii. Using Task Manager, make sure the server is not running (kill the "java" process)
 - iii. Remove the instance directory at %JDEV_USER_DIR%
 - iv. Run the Configure SOA command again.
- g. If Configure SOA was previously successful and you want to reinstall, simply complete step f above.

Application Deployment

10. Deploy your own application (try the simple SOAHello application in the next section or use one of the tutorials at the download location). If you are redeploying, use step 12 below.

After you have created your own composite, you can deploy it to the server.

a. **Deploy method A – Run**

This method rebuilds the app but does not write the binary to disk. It uses the existing (defaulted) deployment plan. This method is for quick turnaround during development. It does not leave the application deployed on the SOA server when the run command terminates and ends the server session.

- i. Select **composite.xml** in your application and then select the **Run** button (green arrow) in the toolbar.

The Run command starts the server if it is not already started and deploys the application using the existing deployment plan. Watch the messages in the logs (there will be errors but you can ignore them unless they are specific to your application) until you see the server log show that your application was initialized. See Appendix A for a sample of this log window.

b. **Deploy method B – Deploy**

This method rebuilds the app and also writes the binary to disk. It also brings up the deployment plan dialog before deploying. This method is used when you want the application to stay deployed, for example when you are deploying a set of applications that work together. When the server is stopped and restarted, the applications will still be deployed.

- i. If the server is not started, press the Start Server Instance button on toolbar (circle with an arrow inside - usually it's next to the last button on the right) This button is disabled if you do not have an application open in JDeveloper or if you have not selected *Integrated OC4J* in step 9c . Wait for the server to completely start up. After you see, "DefaultServer started" wait a little more and you will see the final message "done deploying composites."
- ii. In the application menu - the down-arrow to the right of the application name - select **Deploy** and follow the menu to select **BundledOC4JServer**
- iii. When you see the **Deployment Plan** dialog, select OK to continue with deployment.

After deployment finishes, watch the messages in the logs (there will be errors but you can ignore them unless they are specific to your application) until you see the server log

show that your application was initialized – or go directly to the next step and view your application in the console. See Appendix A for a sample of this log window.

11. Check your application

- a. Start the SOA Console in your browser by opening <http://localhost:8988/SOAConsole>
- b. Run your application
 - i. Select application composite name in the left nav bar. If your composite name is long and extends past the vertical bar, drag the nav bar wider or you won't be able to select your composite
 - ii. Select the service for your composite from the Actions menu (usually "client"). A web services test page opens.
 - iii. Enter any input data and select Invoke.
 - iv. Go back to SOAConsole and use the top right Refresh button to refresh.
 - v. Select the composite instance to view the flow of your composite.

12. Redeploy your application

If you make changes and want to redeploy your application, **you must undeploy** the previous deployment first using method A or B here as appropriate and then follow step 10 above to deploy your application again.

- a. Undeploy method A - Run
 - i. If you used the Run command to deploy your application, open the Application Server Navigator from the View menu and navigate to the app.
 - ii. Right-click your application and select the Undeploy command
 - iii. Alternatively, you can simply terminate the Run by selecting the red box in the top of the run application log window. This also undeploys the application but you'll have to wait for the server to start up again when you deploy again.
 - iv. Go to the server log window and wait for the undeploy message
- b. Undeploy method B – Undeploy
 - i. If you used the Deploy command to deploy your application, open the Application Server Navigator from the View menu and navigate to the app.
 - ii. Right-click your application and select the Undeploy command
 - iii. Go to the server log window and wait for the undeploy message

Congratulations! You are now using the new 11g SOA infrastructure design and runtime. Now try the SOA Suite 11g Tech Preview 3 Sample Application to build an application using all of the components.

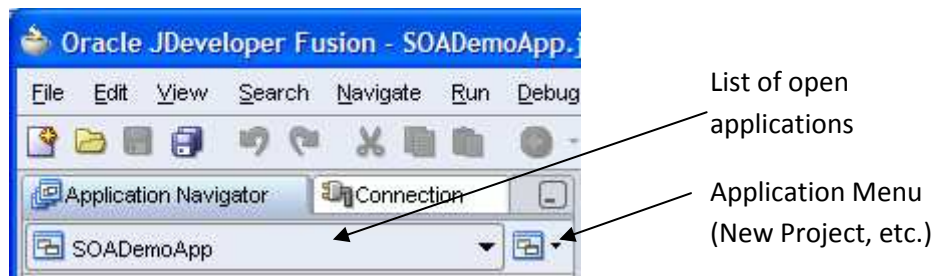
Creating the SOAHelloApp Application

In JDeveloper, perform these steps to create an application called "SOAHelloApp", which will contain the SOAHelloComposite project.

1. In Oracle JDeveloper, select **View > Application Navigator**.

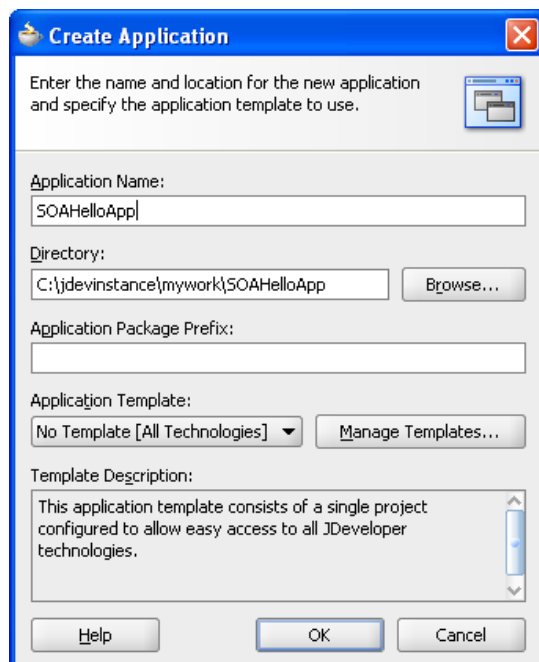
If there are no currently open applications in JDeveloper you will see two options in the Application Navigator window: New Application and Open Application. Select **New Application**.

Otherwise you will see an open application listed in the Application Navigator window and you will need to click the open applications drop-down list and select **New Application**.



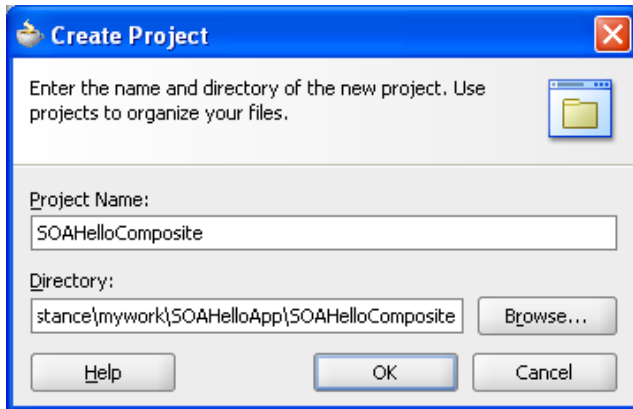
2. In the Create Application dialog, enter these values.

- **Application Name:** enter **SOAHelloApp**.
- **Directory Name:** Make sure there are no spaces in the path! JDeveloper will create this directory.
- **Application Package Prefix:** leave blank for this application.
- **Application Template:** select **No Template [All Technologies]**.

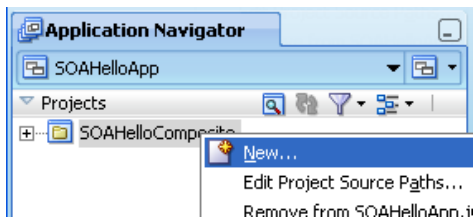


Click **OK**. A Create Project dialog opens.

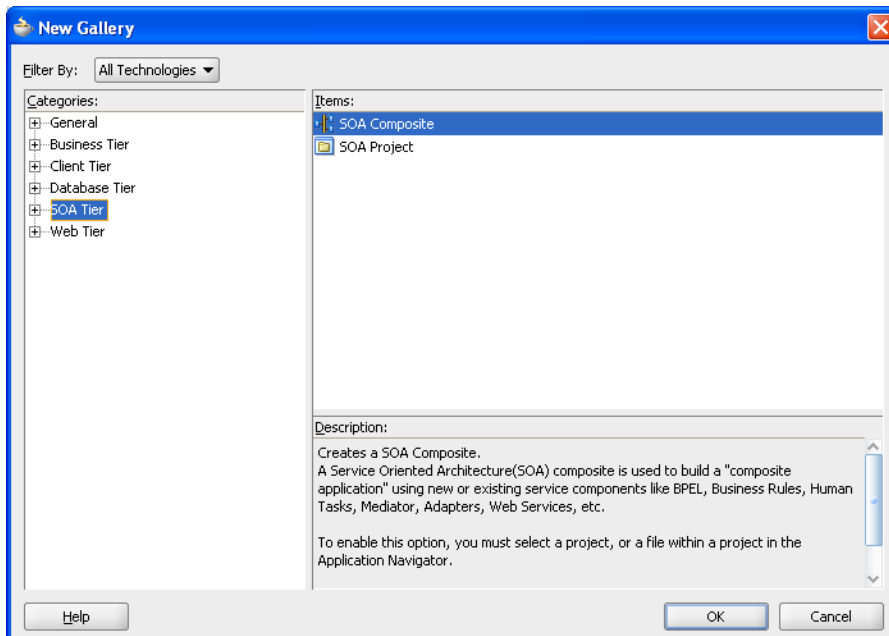
3. In the Create Project dialog, enter **SOAHelloComposite** for the project name and click **OK**. This creates an empty project.



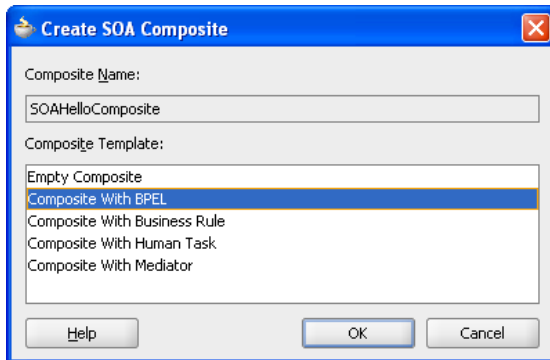
4. Right-click on the new project and select **New...**



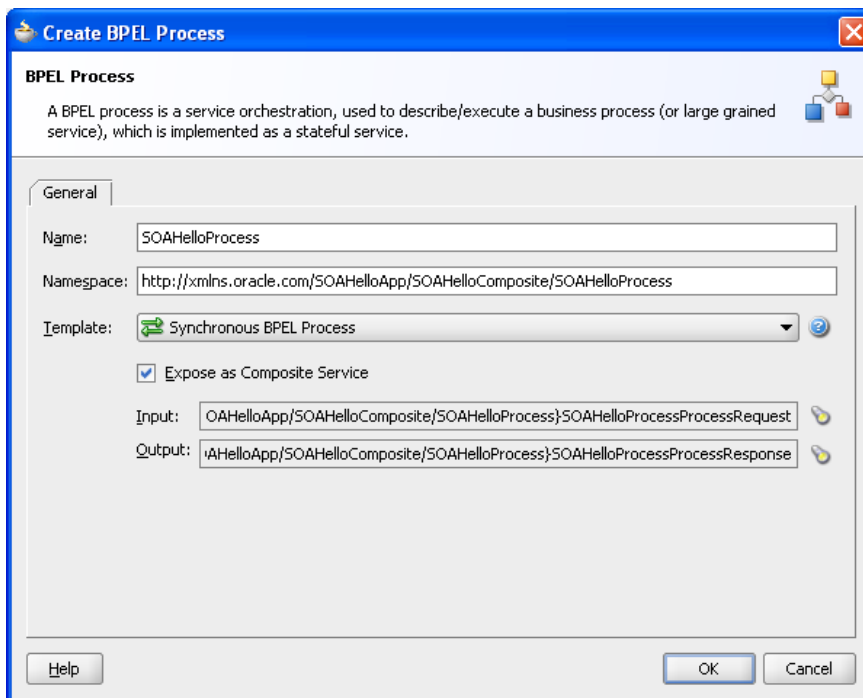
5. Select **SOA Tier** and **SOA Composite** in the New Gallery. Select **OK**. This opens the **Create Composite** dialog.



6. In the **Create Composite** dialog, select **Composite with BPEL** and select **OK**. This opens the **Create BPEL Process** dialog.



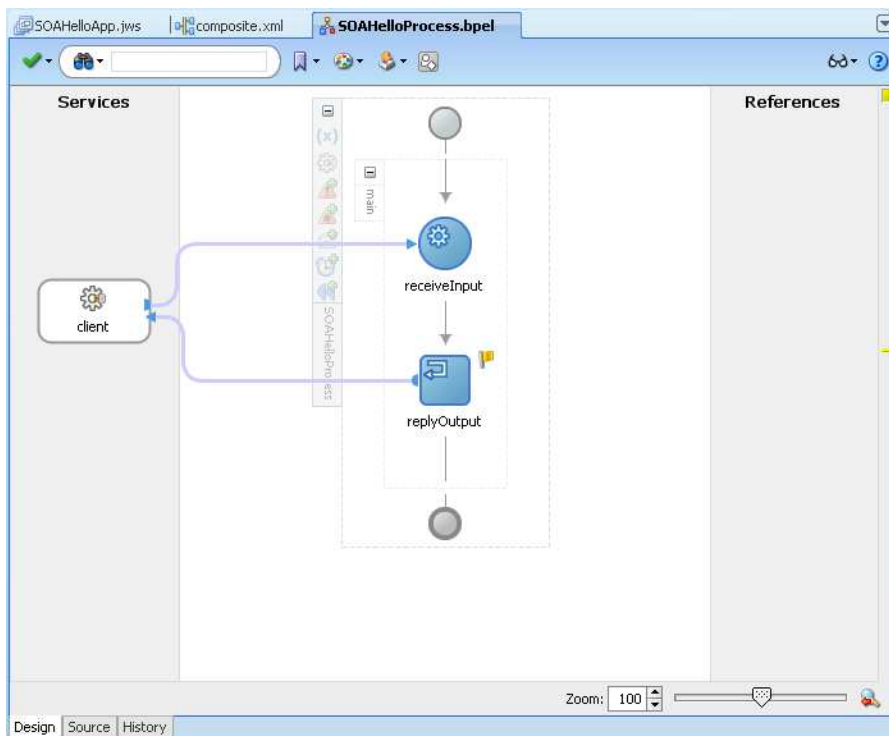
7. Enter the name, **SOAHelloProcess**, and choose **Synchronous BPEL Process** from the list of process types. Make sure **Expose as Composite Service** is checked.



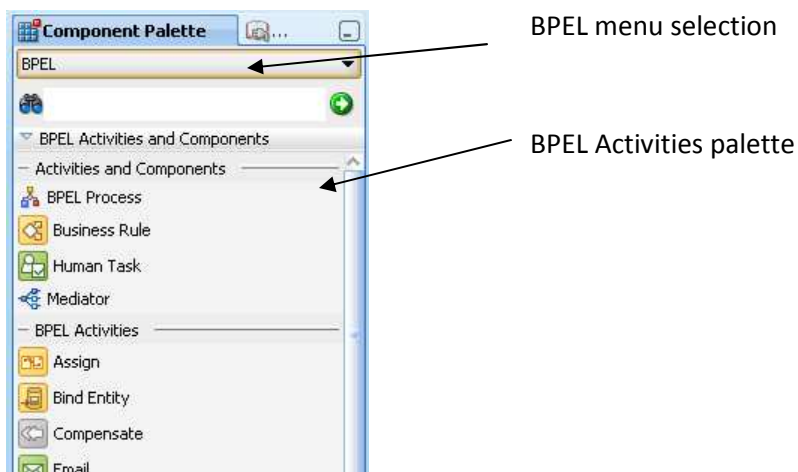
The input and output schemas can be specified as defaulted. The defaulted schemas are a string value for both input and output. Select **OK**.

8. Complete the **SOAHelloProcess** BPEL process
 You see a synchronous process with a partner link coming into a receive activity and a reply activity returning to the partner. The partner name is client and this is the operation name when your process is called. A wsdl file also gets created with all of this information.

There is a small warning icon next to the replyOutput activity. This warning is telling you that you have not set your output parameter yet (double-click the icon to see the error message).

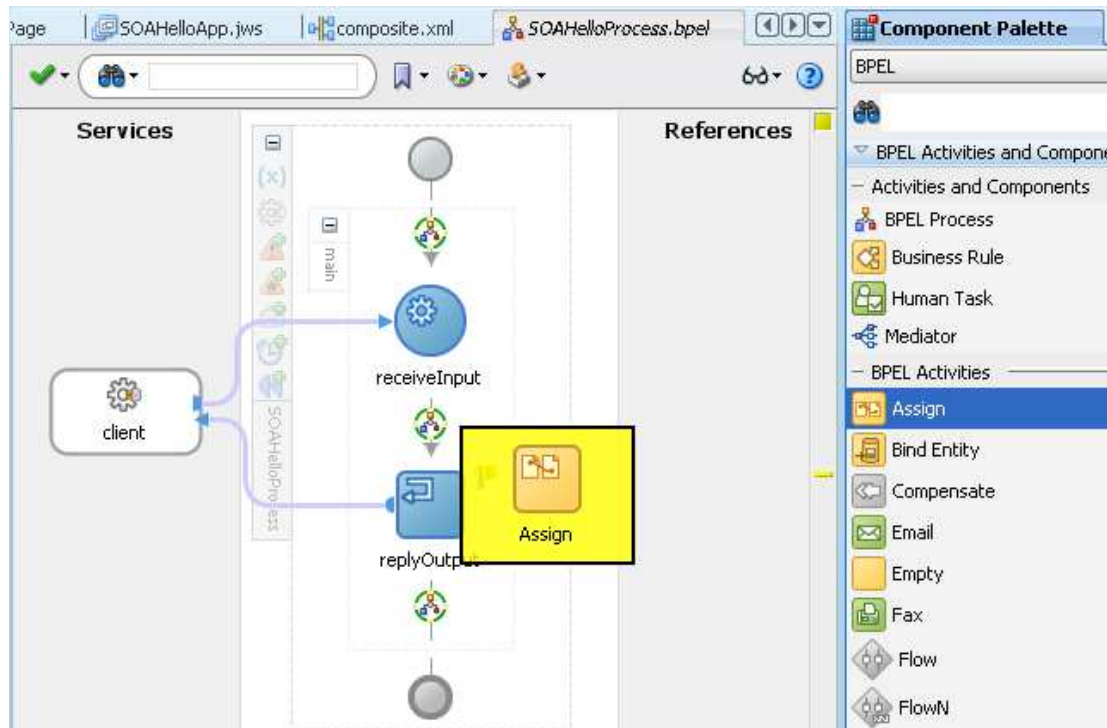


Select the BPEL component from the Components Palette menu on the right (Select View -> Components to show the palette if it is not open). Select the right-pointing arrow next to BPEL Activities to open the activities palette (might be at the bottom of the Component Palette when closed).



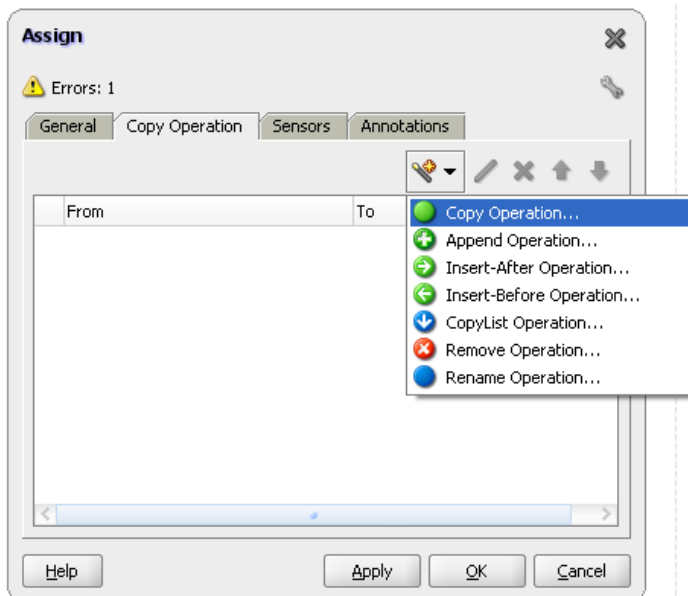
- Click on the Assign activity and drag it to the BPEL process just after the receiveInput activity. You will know you have it located properly when you are right over the small empty circle and

the Assign activity highlights in yellow.



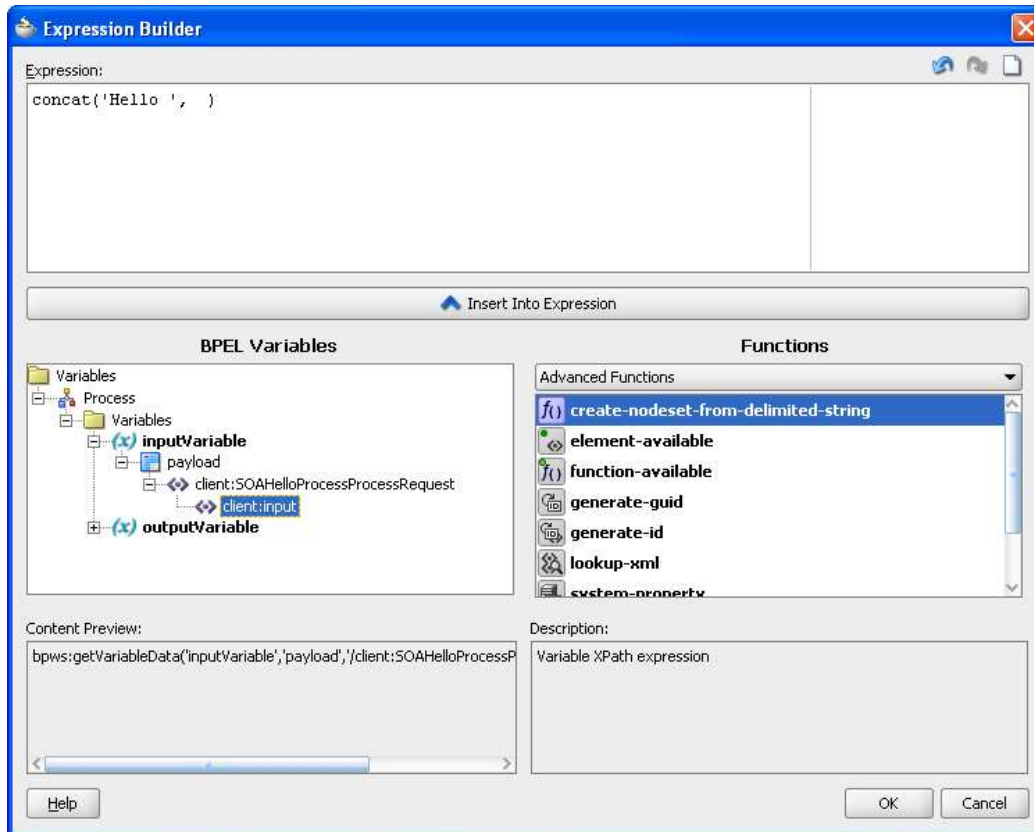
10. Double-click on the Assign activity to open the dialog.

Select "Copy Operation" from the drop-down menu to open the Create Copy Operation dialog.



11. On the left side of the dialog, select "Expression" from the Type drop-down. In this expression, you are going to concatenate the string, "Hello" to the input data string (your name, e.g., "Heidi") so that the resulting string is "Hello Heidi".

12. Select the box with the pencil at the upper right of the Expression box. This opens the Expression Builder. Enter the string: `concat('Hello ',)`
13. With the cursor just before the closing parenthesis, select the input variable to be used as the second part of the concat operation. Double click the variable part or select it and select the **Insert Into Expression** button to move the variable into the expression along with the correct XPath syntax for obtaining the variable value.

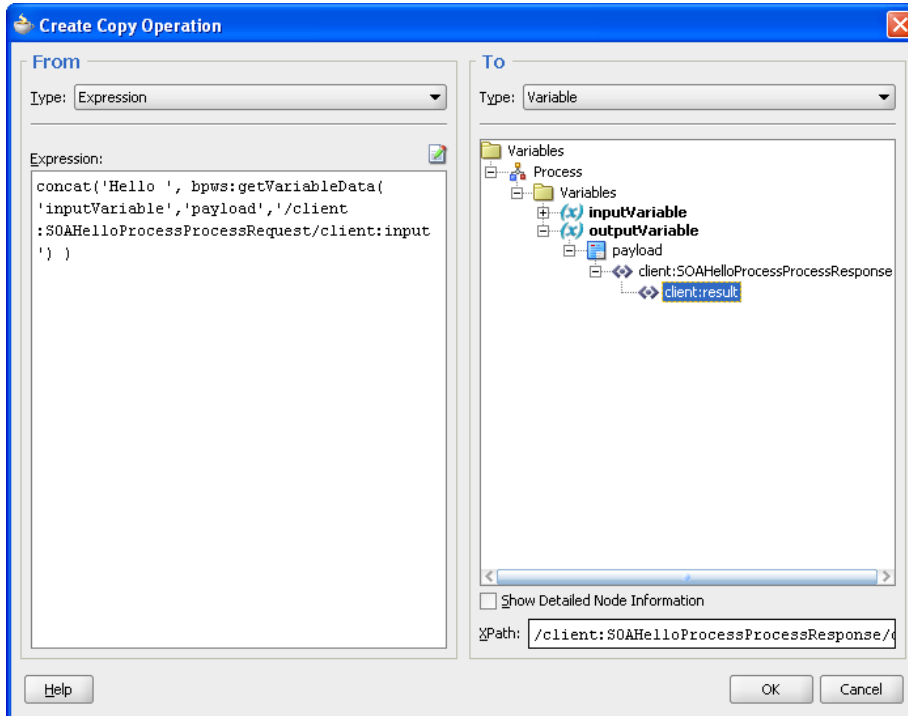


The result should look like this:

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

Select **OK**.

14. On the right side of the copy dialog, leave the Type drop-down set to Variable and navigate the outputVariable until you get to the result element. You'll see this in the XPath field at the bottom: `/client:SOAHelloProcessResponse/client:result`

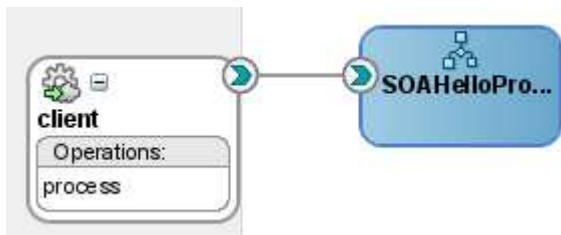


Now the string you created on the left gets copied to the output parameter and this will be sent back to the caller of your application.

Select **OK** to close the **Create Copy Operation dialog**.

Select **OK** to close the **Assign dialog**.

15. Select the green arrow at the top of the BPEL Process window to validate your process. All of the small warning icons should go away but if not, you can double-click the icon to read the warning or error message. Correct any problems and validate your process again.
16. Now return to the composite view of your application. You can select the composite.xml tab at the top of the JDeveloper working area or double-click the composite.xml file in the Applications Navigator to see this view.



You are done. Save all of the files and follow the **Application Deployment** section above to deploy and run your application.

17. When you see the web service test page, you see that your application takes one parameter. Enter a name in the field and then select **Invoke**.

- The application executes and the results are returned to the browser window because the application was created as synchronous. The result XML will look like this (select “Formatted XML” to see the results in a formatted view).

Test Result

View: [Formatted XML](#) | [Raw XML](#)

[Return to Test Page](#)

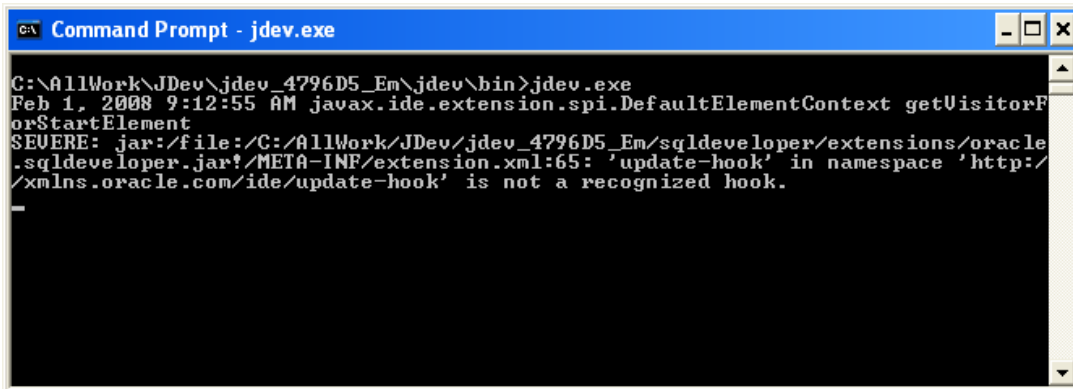
```
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <env:Header>
    <wsa:ReplyTo>
      <wsa:Address
        xmlns:wsa="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
      </wsa:ReplyTo>
    </env:Header>
    <env:Body>
      <SOAHelloProcessProcessResponse
        xmlns="http://xmlns.oracle.com/SOAHelloApp/SOAHelloComposite/SOAHelloProcess">
        <result>Hello Heidi</result> </SOAHelloProcessProcessResponse>
      </env:Body>
    </env:Envelope>
```

- Now use the SOA Console to view the instance details. Refresh the page using the refresh button on the upper right. You’ll see a new instance id show up in the list of instances. Select this id to open the instance details. You can view the BPEL audit trail by selecting the BPEL process in the flow table.

Congratulations! You are done! Now try one of the tutorials to see some of the other components in SOA Suite 11g – Mediator, Human Task, Rules, CEP, etc.

Appendix A: Sample log screens

1. Starting JDeveloper

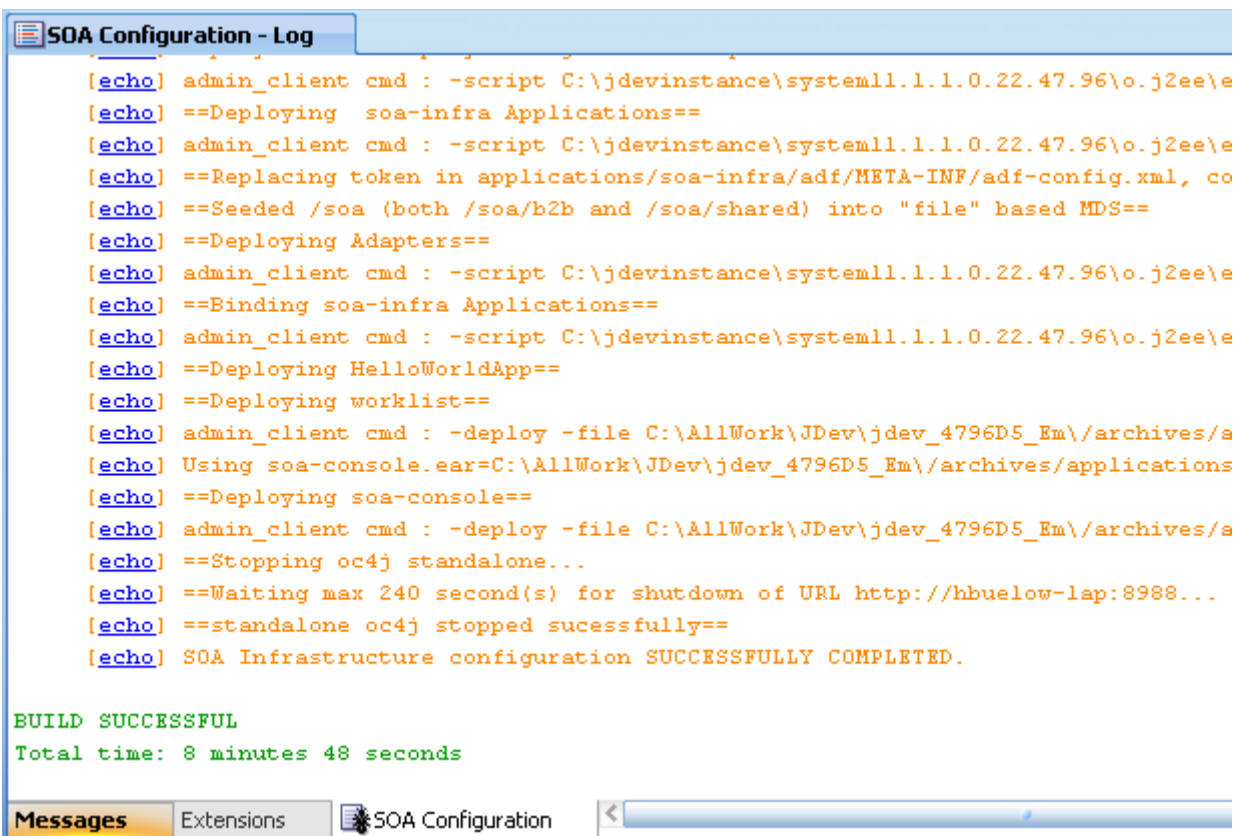


```

C:\AllWork\JDev\jdev_4796D5_Em\jdev\bin>jdev.exe
Feb 1, 2008 9:12:55 AM javax.ide.extension.spi.DefaultElementContext getVisitorForStartElement
SEVERE: jar:/file:/C:/AllWork/JDev/jdev_4796D5_Em/sqldeveloper/extensions/oracle.sqldeveloper.jar!/META-INF/extension.xml:65: 'update-hook' in namespace 'http://xmlns.oracle.com/ide/update-hook' is not a recognized hook.

```

2. SOA configuration



```

[echo] admin_client cmd : -script C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\
[echo] ==Deploying soa-infra Applications==
[echo] admin_client cmd : -script C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\
[echo] ==Replacing token in applications/soa-infra/adf/META-INF/adf-config.xml, co
[echo] ==Seeded /soa (both /soa/b2b and /soa/shared) into "file" based MDS==
[echo] ==Deploying Adapters==
[echo] admin_client cmd : -script C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\
[echo] ==Binding soa-infra Applications==
[echo] admin_client cmd : -script C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\
[echo] ==Deploying HelloWorldApp==
[echo] ==Deploying worklist==
[echo] admin_client cmd : -deploy -file C:\AllWork\JDev\jdev_4796D5_Em\archives/a
[echo] Using soa-console.ear=C:\AllWork\JDev\jdev_4796D5_Em\archives/applications
[echo] ==Deploying soa-console==
[echo] admin_client cmd : -deploy -file C:\AllWork\JDev\jdev_4796D5_Em\archives/a
[echo] ==Stopping oc4j standalone...
[echo] ==Waiting max 240 second(s) for shutdown of URL http://hbuelow-lap:8988...
[echo] ==standalone oc4j stopped successfully==
[echo] SOA Infrastructure configuration SUCCESSFULLY COMPLETED.

BUILD SUCCESSFUL
Total time: 8 minutes 48 seconds

```

3. Application deployment using Run

The screenshot displays the 'SOAHelloApp DefaultServer - Log' window with a 'Property Inspector' tab. The log text shows the following sequence of events:

```

copy the archive to C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\embedded-oc4j\applications\SOAHelloApp.ear
Initialize C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\embedded-oc4j\applications\SOAHelloApp.ear begins...
Unpacking SOAHelloApp.ear
Done unpacking SOAHelloApp.ear
Initialize C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\embedded-oc4j\applications\SOAHelloApp.ear ends...
Starting application : SOAHelloApp
Initializing ClassLoader(s)
Initializing EJB container
Loading connector(s)
Starting up resource adapters
Committing ClassLoader(s)
Started application : SOAHelloApp
Binding web application(s) to site default-web-site begins...
Binding web application(s) to site default-web-site ends...
Application Deployer for SOAHelloApp COMPLETES. Operation time: 3047 msec
Elapsed time for deployment: 14 seconds
---- Deployment finished. ---- Feb 1, 2008 9:29:51 AM
Run startup time: 14922 ms.
[Application SOAHelloApp deployed to Server Instance DefaultServer]

```

Below the log, there are two tabs for 'Messages'. The first tab shows the following messages:

```

08/02/01 09:30:05 BPELServiceEngine=> deploy component SOAHelloApp/SOAHelloProcess
08/02/01 09:30:05 BPELServiceEngine=> loading component SOAHelloApp/SOAHelloProcess
08/02/01 09:30:07 WARNING: ExecutorService.inform Unknown event state3 Ignored
08/02/01 09:30:07 WARNING: ExecutorService.inform Unknown event state4 Ignored
08/02/01 09:30:11 Compiled class(es) "C:\jdevinstance\system11.1.1.0.22.47.96\o.j2ee\embedded-oc4j\deployed-comp
08/02/01 09:30:11
08/02/01 09:30:11 BPELServiceEngine=> initing SOAHelloApp/SOAHelloProcess
08/02/01 09:30:21 WARNING: ExecutorService.inform Unknown event state3 Ignored
08/02/01 09:30:21 WARNING: ExecutorService.inform Unknown event state4 Ignored

```

The second 'Messages' tab is currently empty.