

B Managing the lifecycle of a SOA composite application

B	Managing the lifecycle of a SOA composite application	1
B.1	Introduction to SOA lifecycle.....	1
B.2	Lab overview	2
B.3	DEVELOPMENT environment.....	2
B.3.1	Surfacing binding properties at the composite level	2
B.3.2	Generating and validating a deployment plan.....	4
B.4	TEST environment	6
B.4.1	Compiling and packaging from the command-line with ANT	6
B.4.2	Deploying from the command-line with ANT	6
B.5	PRODUCTION environment	7
B.5.1	Deploying from Enterprise Manager	7
B.6	Challenge 1	8
B.7	Challenge 2	8
B.8	Challenge 3	8

B.1 Introduction to SOA lifecycle

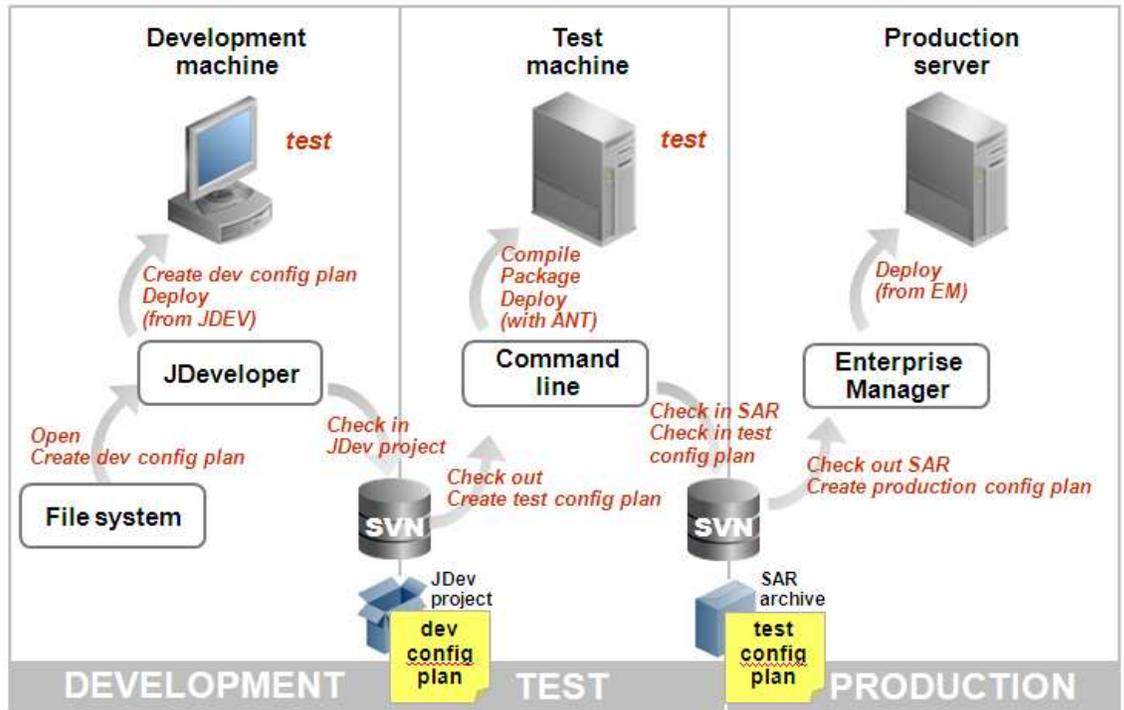
This lab exercise will give you a brief introduction to various techniques that can be used to manage the lifecycle of SOA composite applications.

Oracle SOA Suite provides multiple facilities to help in this area:

- a **variety of deployment methods**:
 - from JDeveloper, the development environment
 - from Enterprise Manager, the management & monitoring environment
 - from the command line, in an interactive fashion
 - from ANT or Python scripts
- the **ability to simultaneously deploy multiple versions of a given composite**, and specify a default version
- **configuration plans** to customize environment-specific values (ex: a web service URL that is different in the test environment than in the production environment)
- **built-in support in JDeveloper for version control systems** such as Subversion.

B.2 Lab overview

In this lab you will create a configuration plan which surfaces a property that can take on different values. This simulates the process of deploying with different values to your different environments: Development, Test, Production.

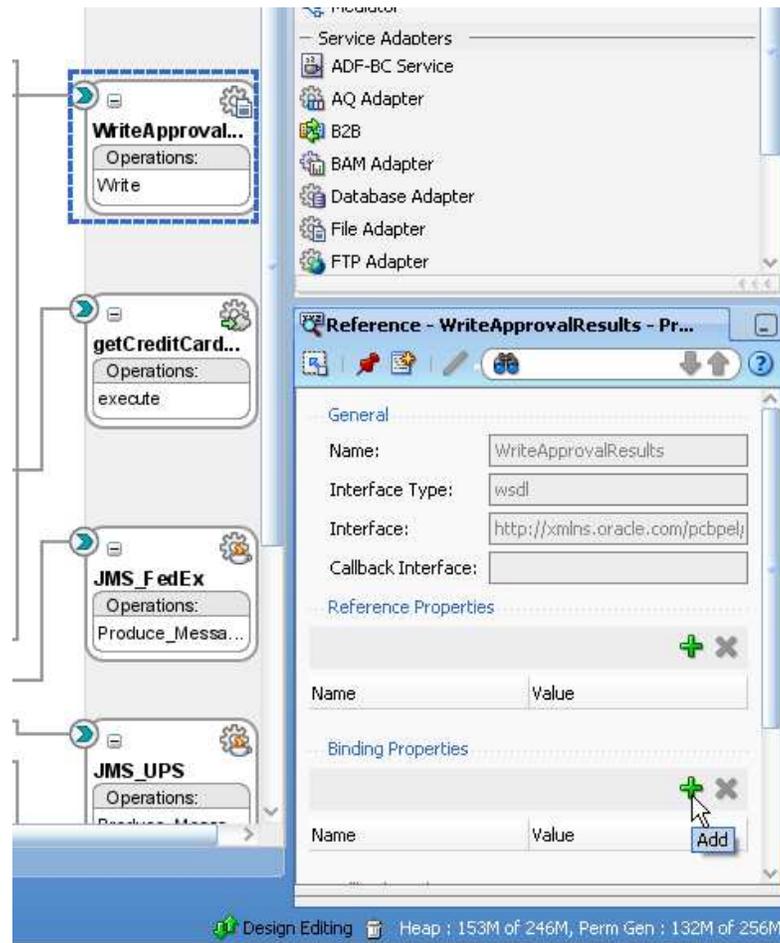


B.3 DEVELOPMENT environment

B.3.1 Surfacing binding properties at the composite level

1. In JDeveloper, open your previously completed POProcessing composite (or the prebuilt solution for chapter 9).
2. Open composite.xml
3. Select the WriteApproval reference (a file adapter)
4. View the properties using the **Property Inspector** in the lower right pane. You may need to enlarge or scroll the pane to see all of the sections. If you do not see the **Property Inspector**, select it from the **View** menu in the toolbar.

5. Using the **Property Inspector**, we are going to add a binding property.
6. Scroll to the **Binding Properties** section and select the **Add** icon.



7. Select **FileNamingConvention** for the property name and **test_%SEQ%.xml** for the value.
8. Click **OK**.



9. Check the source of composite.xml to see the result of this operation: you have surfaced a binding property in composite.xml.

```

<reference name="WriteApprovalResults"
  ui:wSDLLocation="WriteApprovalResults.wsdl">
  <interface.wSDL interface="http://xmlns.oracle.com/pcbpel/adapter/file/POProcessing/POProces
  <binding.jca config="WriteApprovalResults_file.jca">
    <property name="FileNamingConvention" type="xs:string" many="false"
      override="may">test_?SEQ?.xml</property>
  </binding.jca>

```

10. Save-all

B.3.2 Generating and validating a deployment plan

11. From the **Projects Explorer**, select composite.xml
12. Right-click on composite.xml and choose **Generate Config Plan**
13. Name your plan POProcessing_dev_cfgplan.xml
14. It should automatically open in text mode in JDeveloper.
15. Locate the reference for WriteApprovalResults and edit the FileNamingConvention property to **orderoutput_?SEQ?.xml**

```

<reference name="WriteApprovalResults">
  <!--Add search and replace rules for the binding properties-->
  <binding type="jca">
    <property name="FileNamingConvention">
      <replace>orderoutput_?SEQ?.xml</replace>
    </property>
  </binding>
</reference>

```

16. In addition, in the wsdlAndSchema section, add the PhysicalDirectory jca property (this is the directory that you configured for the file adapter to write to) and replace it with another directory, for instance "C:\tmp\out".

```

104 <wsdlAndSchema name="approveLargeOrder.wsdl|FulfillmentProcess.wsdl|getStatusByC
105 <jca:property name="PhysicalDirectory">
106 <replace>c:\tmp\out</replace>
107 </jca:property>
108 <searchReplace>
109 <search></search>
110 <replace></replace>
111 </searchReplace>
112 </wsdlAndSchema>

```

Note that you can restrict the scope of this global replace to a single file by only leaving WriteApprovalResults_file.jca in the list of files to consider for this search & replace.

Now it is time to validate your new config plan to ensure it works as expected:

17. Right-click on composite.xml
18. Select **Validate Config Plan**
19. Select OK in the **Composite Configuration Plan Validator** dialog.
20. The report.log opens automatically. You should see the following:

```

Reference [ WriteApprovalResults ]
  Reference Bindings
    Binding [ jca ]
  Property [ FileNamingConvention ]
    Old [ test_%SEQ%.xml ]
    New [ orderoutput_%SEQ%.xml ]
Reference [ getCreditCardStatus ]
  Reference Bindings
    Binding [ ws ]
Attribute name=port
  No change in old and new value http://oracle.
Attribute name=location
  No change in old and new value http://localho
Reference [ JMS_USPS ]
  Reference Bindings
    Binding [ jca ]
Reference [ JMS_UPS ]
  Reference Bindings
    Binding [ jca ]
Reference [ JMS_FedEx ]
  Reference Bindings
    Binding [ jca ]
---End Match for composite [ POProcessing ] in config plan---
Checking for replacement in wsdl and schema files
WSDL WriteApprovalResults_file.jca JCA properties
Property [ PhysicalDirectory ]
  Old Value [ c:\temp ]
  New Value [ c:\tmp\out ]

```

21. Finally, deploy from JDev and attach this deployment. Give this new deployment a different revision ID and ensure the behavior is as expected (i.e. the file adapter now writes all message to c:\tmp\out with a name orderoutput_<number>.xml).

B.4 TEST environment

B.4.1 Compiling and packaging from the command-line with ANT

Administrators typically do not want to work with IDEs and prefer to operate from the command-line or with scripts. Let's take the role of an administrator in this section and exclusively work from the command-line.

22. First, open a command prompt and set your environment for command-line deployment via ANT.

```
set PATH=C:\Oracle\Middleware\jdev_5361\modules\org.apache.ant_1.7.0\bin;%PATH%
```

23. Navigate to the following directory

```
cd C:\Oracle\Middleware\jdev_5361\jdeveloper\bin
```

24. Look at the available ANT commands:

```
dir ant-sca*
```

```
ant-sca-compile.xml    ant-sca-deploy.xml    ant-sca-mgmt.xml
ant-sca-package.xml   ant-sca-test.xml      ant-sca-upgrade.xml
```

Hint:

use `ant -f <command> help` to get usage information for a given command.

25. Compile and package POProcessing using the `ant-sca-package` command. This command also sets the version number. You will need to have `JAVA_HOME` set to run ant.

For instance (but adjust this to your own environment and application paths):

```
ant -f ant-sca-package.xml
-DcompositeDir=C:\po\POProcessing\POProcessing
-DcompositeName=POProcessing
-Drevision=6-cmdline
-Dscac.application.home=C:\po\POProcessing
```

If successful you should see a "BUILD SUCCESSFUL" message.

26. Check for the `[jar]` message that will tell you where the script is writing the resulting SAR file:

```
[jar] Building jar: C:\demo\end2end-105-
POProcessing\po\solutions\ch9\POProcessing\POProcessing\deploy\sca_POProce
ssing_rev6-cmdline.jar
```

B.4.2 Deploying from the command-line with ANT

Now deploy this SAR file from the command-line using `ant-sca-deploy.xml`.

27. Use `ant-sca-deploy.xml` to deploy previously packaged SAR file. Adjust the following command to your own environment:

```
ant -f ant-sca-deploy.xml
-DserverURL=http://localhost:8001
```

```

-DsarLocation=C:\po\POProcessing\POProcessing\deploy\
sca_POProcessing_rev6-cmdline.jar
-Doverwrite=true
-Duser=weblogic
-Dpassword=welcome1
-DforceDefault=true
-Dconfigplan=C:\po\POProcessing\POProcessing\
POProcessing_dev_cfgplan.xml

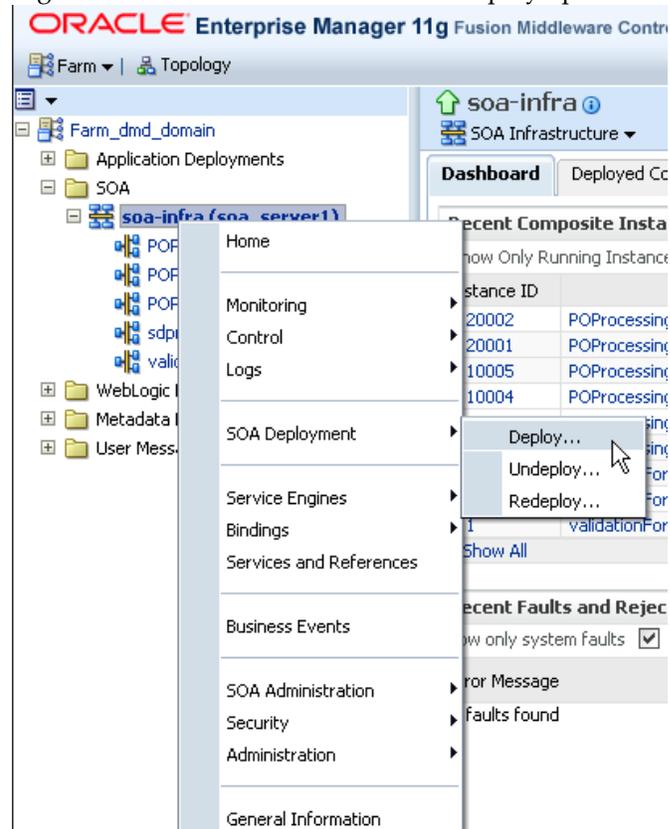
```

B.5 PRODUCTION environment

B.5.1 Deploying from Enterprise Manager

In this section we will explore the deploy options from Enterprise Manager.

Right-click on soa-infra and select the deploy option:



Point EM to your local SAR file and configuration plan:

Select Archive ?

This wizard lets you create a runtime environment for SOA composite applications. Once this operation is performed, these bundle containing revisions of multiple SOA composites can be deployed.

Specify the archive or expanded directory and deployment plan to deploy a single revision of a SOA composite. Or specify .

Archive or Exploded Directory

You can deploy a Service archive (SAR) or a ZIP file containing one or more Service archives (SARs). You can also deploy a running. Ensure that the revision information for each SOA composite is provided in its application package.

Archive is on the machine where this web browser is running.

Archive or exploded directory is on the server where Enterprise Manager is running.

Deployment Plan

The deployment plan is a file that contains the deployment settings for a SOA composite revision.

No external deployment plan is required.

Deployment plan is on the machine where this web browser is running.

Deployment plan is on the server where Enterprise Manager is running.

B.6 Challenge 1

Repeat some of the above commands using WLST instead of ANT.

- 1) go to your Oracle home, under common/bin
(ex: C:\as11wl\home_LOAD16F\Oracle_SOA1\common\bin)
- 2) Enter ./wlst.cmd to start the interactive WLST environment

B.7 Challenge 2

Create ANT scripts to:

1. **create the pre-requisites for the class:** JDBC datasources, JMS destination and connection factory, etc.
2. **deploy in one pass the complete end2end-105 lab** (chapter 9)

Then forward the result to your instructor ;-)

B.8 Challenge 3

Add Subversion in the mix to source-control JDev projects and jar files.

- Download the VisualSVN server here:
<http://www.visualsvn.com/server/download/>
- Check out the JDeveloper cue card for working with Subversion (no extra software required)
- Download the standalone client for Subversion here (for operating from the command-line):
<http://www.sliksvn.com/en/download>