Practices for Lesson 8: Deploying and Testing SOA Composite

Chapter 8

Practices for Lesson 8

Practices Overview

In this practice, you deploy the HelloWorld composite application project to the SOA server and use the Enterprise Manager web application to test the service with a sample input. You then write an Ant script to undeploy the application. Finally, you extract the composite application files from the SAR file and examine them to learn the expected behavior of the application.

Practice 8-1: Deploying and Testing the HelloWorld Composite Application

Overview

In this practice, you use an Ant script to deploy the HelloWorld composite application project to the SOA server and use the Enterprise Manager web application to test the service with a sample input.

Assumptions

- You have successfully completed all previous practices.
- Oracle WebLogic Server is running.

Tasks

Deploying the HelloWorld Composite Application

1. Deploy the application.
   a. Open a terminal window.
   b. In the open terminal window, issue the following commands:

      ```
      $ cd /practices/practice08/scripts
      $ ./deployHelloWorld.sh
      ```

   c. When prompted, supply the username `weblogic` and the password for the `weblogic` user.

      The script runs and reports successful deployment of the application.

```
bash-4.1$ sh deployHelloWorld.sh
Buildfile: /u01/oracle/product/fmw/soa/bin/ant-sca-deploy.xml
     [echo] oracle.home = /u01/oracle/product/fmw/soa/bin/..

deploy:
    [input] skipping input as property serverURL has already been set.
    [input] skipping input as property sarLocation has already been set.
    [input] skipping input as property password has already been set.
[deployComposite] Processing sar=/practices/practice08/deploy/sca_HelloWorld_rev1.0.jar
[deployComposite] Adding sar file - /practices/practice08/deploy/sca_HelloWorld_rev1.0.jar
[deployComposite] INFO: Creating HTTP connection to host:soainternal.example.com, port:8080
[deployComposite] Enter username and password for realm 'default' on host soainternal.example.com:8080
[deployComposite] Authentication Scheme: Basic
[deployComposite] Username:
weblogic
[deployComposite] Password:

[deployComposite] INFO: Received HTTP response from the server, response code=200
[deployComposite] ---->Composite deployment produced 0 warning/severe messages
[deployComposite] ---->Deploying composite success.

BUILD SUCCESSFUL
Total time: 10 seconds
```

2. Examine the `deployHelloWorld.sh` script to understand what just happened.

```
# deployHelloWorld.sh

# variables store ant and ant script locations
MW_HOME=/u01/oracle/product/fmw
ANT_HOME=$MW_HOME/oracle_common/modules/org.apache.ant_1.9.2/bin
ANT_CMD=$ANT_HOME/ant
ANT_SCRIPT=$MW_HOME/soa/bin/ant-sca-deploy.xml
```

Note: The Ant command line utility uses the `ant-sca-deploy.xml` script to deploy the application. Both the Ant command line utility and the `ant-sca-deploy.xml` script are included in the Oracle SOA Suite installation.

```
# variables store ant script parameters
SERVER_URL=http://soainternal.example.com:8080
SAR_LOCATION=/practices/practice08/deploy/sca_HelloWorld_rev1.0.
jar
OVERWRITE=true
```

Note: The preceding lines provide the URL of the WebLogic Administration Server, the location of the composite application jar file, and the instruction to overwrite any composite of the same name and version number that is already deployed. (Version numbers are discussed in the next lesson titled “Managing the Composite Application Life Cycle.”)

```
# invoking the ant script
$ANT_CMD -f $ANT_SCRIPT -DserverURL=$SERVER_URL -
DsarLocation=$SAR_LOCATION -Doverwrite=$OVERWRITE
```

Note: Although this is not always a best practice, you could also set the username and password values as variables and pass them as parameters when calling the Ant command line utility. Because they have been omitted in the provided script, you are prompted for these values at run time. If you were providing the username and password as parameters, you would add `USER` and `PASSWORD` to the preceding script parameter assignments, and you would append the parameters to the command line in the script invocation.

```
# additional script parameter assignments
# USER=weblogic
# PASSWORD=<password>

# additional script parameters passed at invocation
# -Duser=$USER -Dpassword=$PASSWORD
```

Practice 8-2: Testing the Application

Overview

In this section, you use a web browser to access Oracle Enterprise Manager Fusion Middleware Control and initiate a test of the deployed HelloWorld application.

Assumptions

- You have successfully completed practice 8-1.
- Oracle WebLogic Server is running.

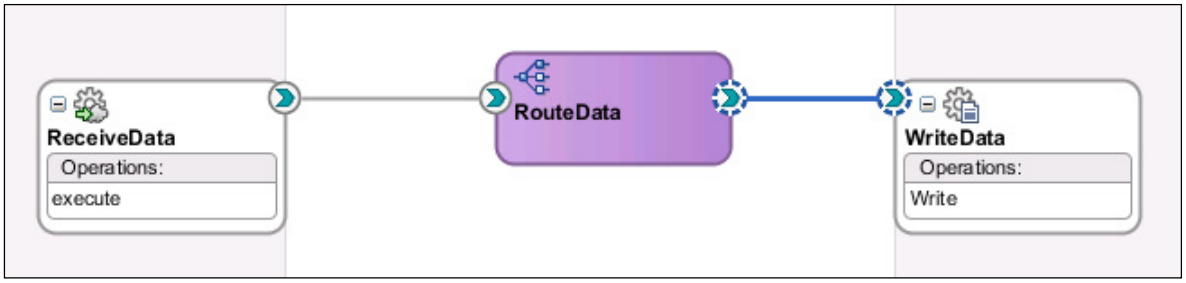
Tasks

1. Familiarize yourself with the composite application.

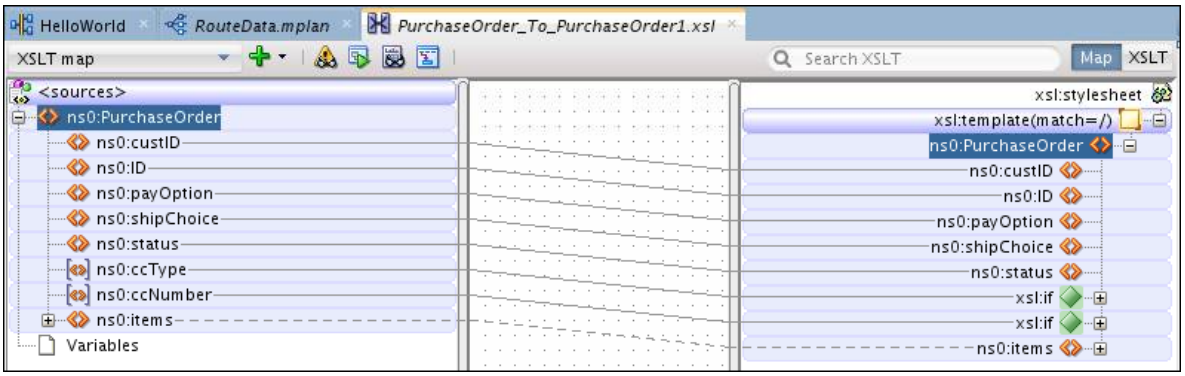
   The following screenshot is of the `composite.xml` file that is part of the HelloWorld application. This file describes the entire composite assembly of services, service components, and references. There is one `composite.xml` file for each SOA project.

   The left swimlane is for services (such as web services, REST adapters, or JCA adapters) that provide an entry point to the SOA composite application. The right swimlane is for references that send messages to external services in the outside world, such as web services or JCA adapters. The center swimlane is for components such as BPEL processes, business rules, human tasks, Oracle Mediators, and spring components.

   In this example, a client connects to the application by invoking the *execute* operation of the ReceiveData service. The data is passed to the RouteData mediator, which transforms the data and routes it its next destination. In this case, the destination is a file adapter, WriteData, which writes the data to a file.



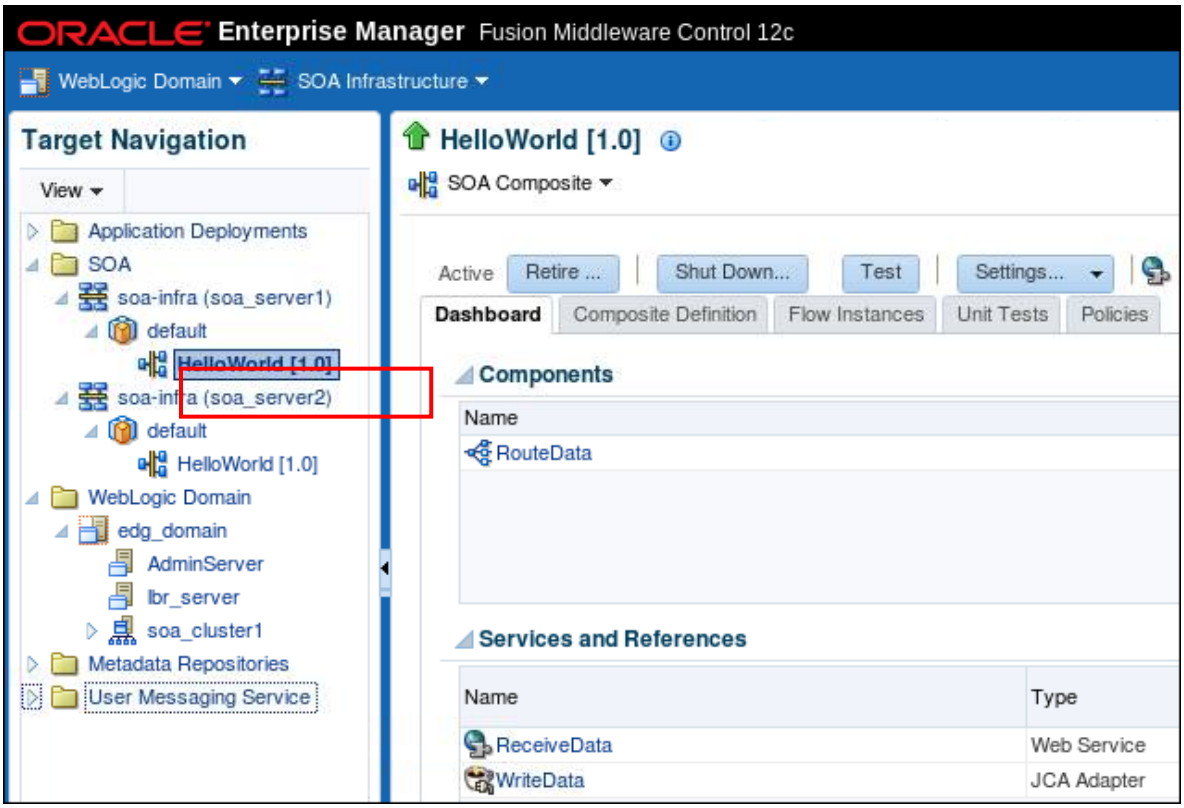
   The following screenshot shows the transformation in the mediator. In this simple example, the format of the incoming and outgoing data is identical. Data from each incoming field is mapped to its corresponding node in the destination message.



2. Log in to the Oracle Enterprise Manager Fusion Middleware Control application by performing the following steps:

a. Open a web browser and provide the URL http://admin.example.com:8080/em.
   Tip: You may want to bookmark this URL, because you will use it again in this course.
b. Log in with the username `weblogic` and the `weblogic` user password.

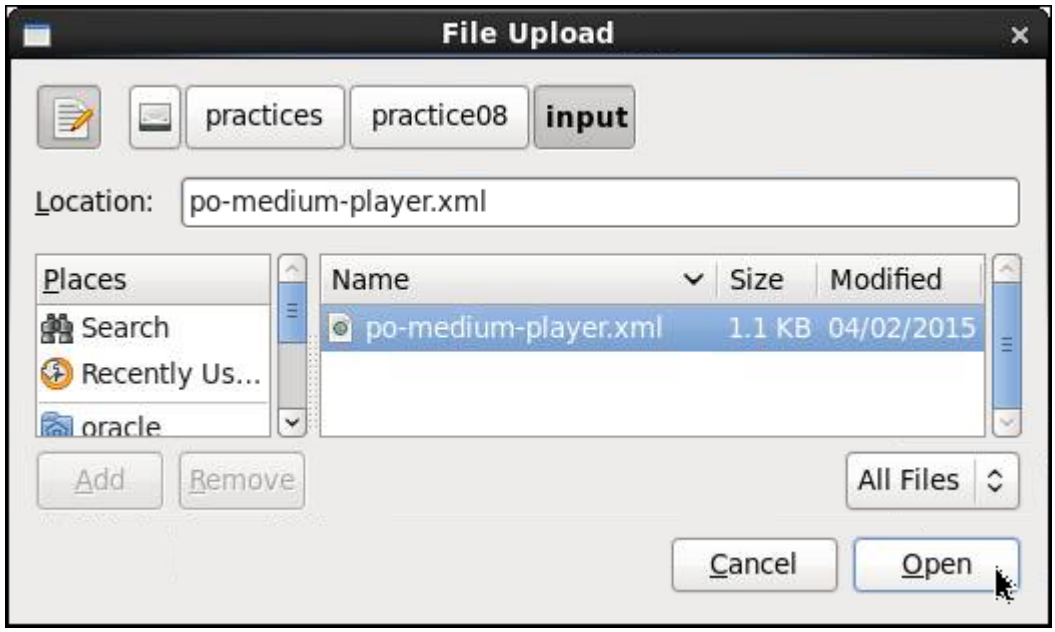
3. In the Target Navigation pane, expand the SOA > soa-infra (soa_server1) > default nodes in the tree and click the “HelloWorld [1.0]” link.



4. To initiate a test of the HelloWorld composite, perform the following steps:
   a. On the “HelloWorld [1.0]” home page, click Test.

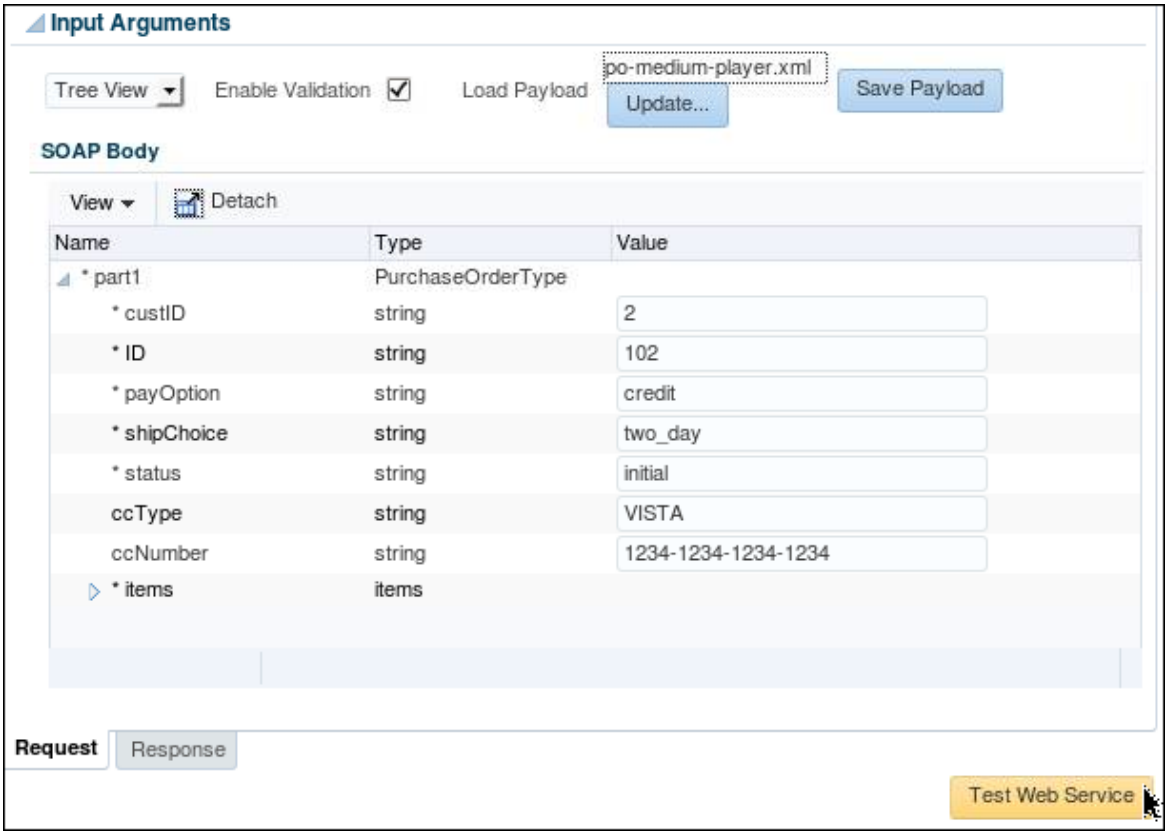
   b. On the Test Web Service page, scroll down to the Request tab’s Input Arguments section and click the Browse button.

c. In the File Upload dialog box, navigate to the `/practices/practice08/input` folder, select `po-medium-player.xml`, and click Open.



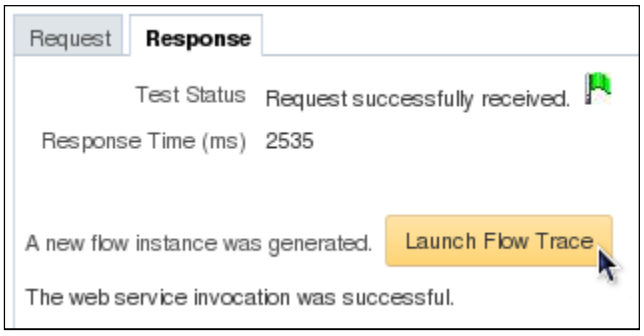
The initially supplied XML test data is replaced by the contents of the file.

5. On the Test Web Service page, scroll to the bottom of the page and click Test Web Service.



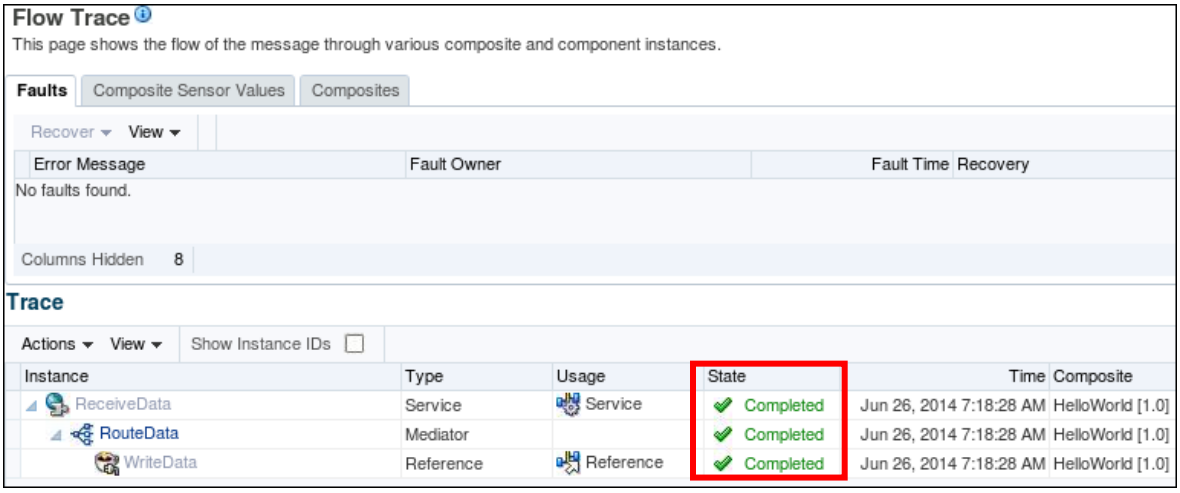
Note: This action sends the XML data as an input message to the HelloWorld composite application. It might take several seconds for the page to refresh with the response that is displayed on the Response tab. Wait until the page is refreshed.

6. On the “HelloWorld [1.0]” Response tab page, click the Launch Flow Trace button to view the results of the asynchronous composite application.



Note: If your web browser blocks the pop-up window, perform steps to correct allow pop-ups.

7. On the Flow Trace page, verify that the ReceiveData, RouteData, and WriteData components have a Completed state, indicating that the application executed successfully.



8. Close the Flow Trace window.
9. Use your web browser to open the output file `/practices/practice08/files/output/hardcoded/order_1.xml`.
10. Verify that the XML file contains data that matches the following screenshot:

```
−<ns0:PurchaseOrder>
    <ns0:custID>2</ns0:custID>
    <ns0:ID>102</ns0:ID>
    <ns0:payOption>credit</ns0:payOption>
    <ns0:shipChoice>two_day</ns0:shipChoice>
    <ns0:status>initial</ns0:status>
    <ns0:ccType>VISTA</ns0:ccType>
    <ns0:ccNumber>1234-1234-1234-1234</ns0:ccNumber>
  −<ns0:items>
    −<ns0:item>
        <ns0:productId>SKU304</ns0:productId>
        <ns0:productName>Music Player 8Gb</ns0:productName>
        <ns0:price>140</ns0:price>
        <ns0:quantity>10</ns0:quantity>
      </ns0:item>
    −<ns0:item>
        <ns0:productId>SKU303</ns0:productId>
        <ns0:productName>Music Player 4Gb</ns0:productName>
        <ns0:price>99</ns0:price>
        <ns0:quantity>10</ns0:quantity>
      </ns0:item>
    </ns0:items>
</ns0:PurchaseOrder>
```

11. Minimize or close your web browser.

Practice 8-3: Undeploying an Application with an Ant Script

Overview

In this practice, you complete and run an Ant script to undeploy an application.

Assumptions

This practice assumes that you have completed Practice 8-1 successfully.

Tasks

1. Use gedit to open the file `/practices/practice08/scripts/undeployHelloWorld.sh`.
2. Update the script to assign values to the five missing script parameters.

```
# script parameters
ACTION=undeploy
SERVER_URL=
COMPOSITE_NAME=
REVISION=
USER=
PASSWORD=
```

   Hint: There are several ways to learn the revision number of the application. It can be found as part of the name of the SAR file that you deployed earlier in this practice. You can also get the number and learn more about the application in Enterprise Manager.
3. Save your changes and close gedit.
4. From a terminal window, execute the `/practices/practice08/scripts/undeployHelloWorld.sh` script.
5. Optional: Use Enterprise Manager to verify that the application has been undeployed.

Practice 8-4: Exploring the Composite Application Files

In this section, you extract the composite application configuration files from the SAR file. You then examine these files to learn about the expected behavior of the application.

Tasks

1. Extract the application files from the SAR file.
   a. Open a terminal window.
   b. Enter the following commands:

      ```
      $ cd /practices/practice08/deploy
      $ unzip sca_HelloWorld_rev1.0.jar -d v1
      $ cd v1
      $ ls -lt
      ```

      Note: The `-d` command-line option in the `unzip` command creates the `v1` subfolder and extracts the contents of the `.zip` file into that subfolder. A directory listing similar to the one that follows is displayed:

      ```
      bash-4.1$ ls -lt
      total 36
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 Adapters
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 Mediators
      drwxr-xr-x 3 oracle oinstall 4096 May 12 00:27 SCA-INF
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 Schemas
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 testsuites
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 Transformations
      drwxr-xr-x 2 oracle oinstall 4096 May 12 00:27 WSDLs
      -rw-r--r-- 1 oracle oinstall 2705 Feb 17 09:53 composite.xml
      -rw-r--r-- 1 oracle oinstall  704 Feb 16 17:04 measurements.xml
      ```

2. Use your web browser to open the `composite.xml` file. Examine the contents of the file to answer the following questions:
   - What file would you examine to know the data format that is expected when the composite is invoked?
   - What file would you open to examine the transformation performed by the mediator?
   - What file would you examine to learn about the data format that is written to file by the composite?
   - What file would you examine to learn about the directory where data is written by the composite?
3. Locate the files that you have identified within the directory hierarchy. Open each of the files you have named and confirm your answer.

Practices for Lesson 9: Managing the Composite Application

Chapter 9

Overview of Practices for Lesson 9

Practices Overview

In this practice, you perform the following tasks:

- Create a partition
- Deploy a SOA bundle
- Use the `ant` command to determine what applications are deployed
- Unzip the `jar` files to learn how each application is configured
- Manage the life cycle of the deployed composites
- Undeploy the applications and delete the partition

Practice 9-1: Creating a Partition

Overview

In this practice, you perform the following tasks:

- Create a partition
- Deploy a SOA bundle
- Use the `ant` command to determine what applications are deployed

The `antPartitions.sh` Script

Many of the tasks in this practice ask you to issue `ant` commands. To reduce the amount of typing required, two scripts have been provided. One deploys a SOA bundle. The other provides menu-based access to a variety of partition management actions. Both scripts show you the `ant` command line and each of the parameter values to be passed before they execute it.

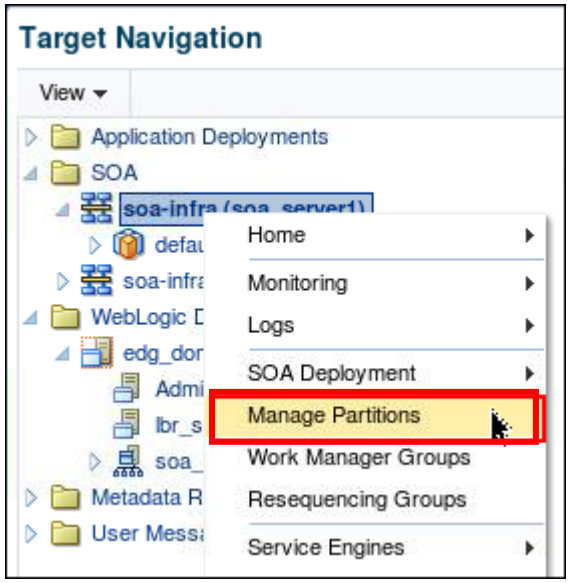
Assumptions

- You have successfully completed all previous practices.
- Oracle WebLogic Server is running.

Tasks

Creating a Partition

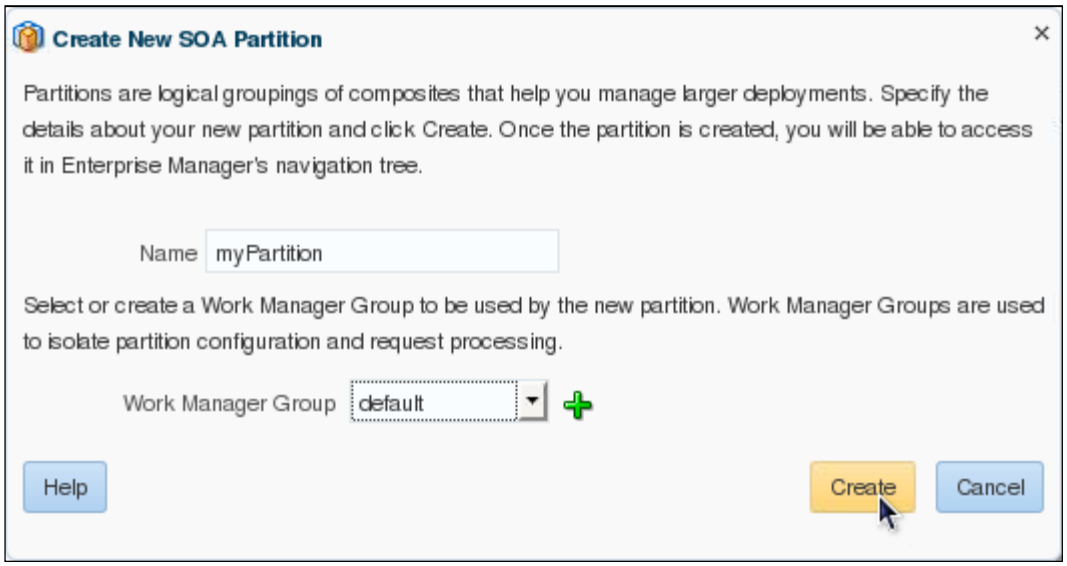
1. Create a partition.
   a. Open Enterprise Manager at http://adminvh.example.com:7001/em.
   b. Right-click soa-infra (soa_server1) and select Manage Partitions.



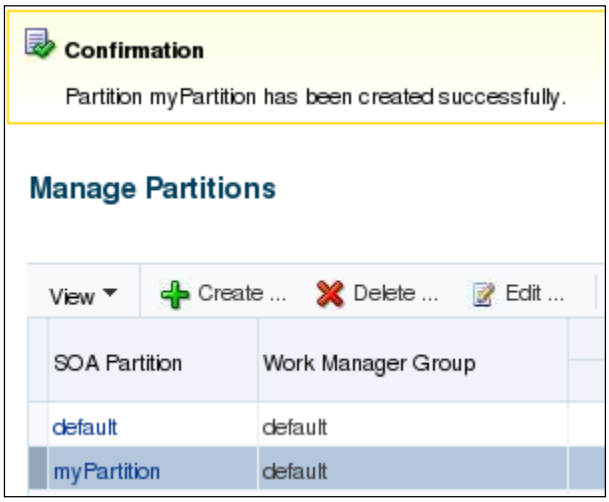
   c. Click Create.
      The Create New SOA Partition dialog box is displayed.
   d. In the Name field, enter `myPartition`.
   e. Accept the `default` work manager group.
      Note: A work manager is an Oracle WebLogic Server entity that represents a logical thread pool. It is similar to a queue in which work items line up for processing. You can

define priorities for the work to be processed by work managers. Work managers manage thread pools internally and automatically, providing for optimal scheduling thereby.

f. Click Create.



The new partition is displayed in both the navigator under soa-infra and the SOA Partition column of the Manage Partitions page.



You can now deploy composites to this partition by selecting “Deploy to This Partition” from the Deployment drop-down list or by right-clicking a specific partition in the navigator and selecting *Deploy to This Partition*. When a composite is deployed to a partition, it is displayed below the partition in the navigator. After it is deployed, a composite cannot be transferred to a different partition.

Note: You can also create partitions with the Oracle WebLogic Scripting Tool (WLST) and `ant` commands. Most of the remaining activities in this practice use `ant` commands.

2. Use the `antPartitions` script to list the partitions on this host.

   a. In a Terminal window, enter the following commands:

   ```
   $ cd /practices/practice09/scripts
   $ ./antPartitions.sh
   ```

Note: The `antPartitions` script displays a menu of partition management options.

b. Select the List Partitions option.

c. Provide the `weblogic` user password when prompted.

The script lists the partitions on this host.

```
folderMgrTask:
     [java] calling FolderManager.initConnection(), m_platform= weblogic, m_host
=soavh01.example.com, m_port=8001, m_user=weblogic
     [java] Connecting to: service:jmx:t3://soavh01.example.com:8001/jndi/weblog
ic.management.mbeanservers.runtime
     [java] connection initiated
     [java] folderMBean=oracle.soa.config:name=soa-infra,j2eeType=FolderLifecycl
eConfig,Application=soa-infra
     [java] Following 2 partitions are currently available on the platform:
     [java]
     [java] 1. myPartition
     [java] 2. default
     [java]
```

Deploying a SOA Bundle to a Partition

In this step, you deploy a composite bundle to the new partition. You invoke a command script, which in turn invokes an Ant script that actually performs the deployment.

3. Use the `antDeploy.sh` script to deploy a bundle of composite applications to the new partition.

   a. In a Terminal window, enter the following commands:

   ```
   $ cd /practices/practice09/scripts
   $ ./antDeploy.sh
   ```

   b. Provide the `weblogic` username and password when prompted.

   The script deploys the bundle.

```
[deployComposite] ---->Composite deployment produced 0 warning/severe messages
[deployComposite] ---->Deploying composite success.

BUILD SUCCESSFUL
Total time: 15 seconds
```

Listing the Applications in the Partition

4. Use the `antPartitions` script to determine which composite applications were just deployed. Determine which version of the application is the default.

   a. In a Terminal window, enter the following commands:

   ```
   $ cd /practices/practice09/scripts
   $ ./antPartitions.sh
   ```

   Note: The `antPartitions` script displays a menu of partition management options.

   b. Select the List Composites in Partition option.

   c. Provide the `weblogic` password when prompted.

   The script lists the composites that are deployed to the new partition, as well as some information about their state.

Practice 9-2: Examining the Contents of the SAR Files

In this practice, you perform the following:

- Unzip `jar` files to learn how each application is configured
- Manage the life cycle of the deployed composites
- Undeploy the applications and delete the partition

Assumptions

- You have successfully completed practice 9-1.
- Oracle WebLogic Server is running.

Tasks

Unzipping the SOA Bundle

1. In a Terminal window, enter the following commands:

   ```
   $ cd /practices/practice09
   $ unzip deploy/mybundle.zip -d tmp
   ```

   Note: The `jar` files for each of the deployed composites are extracted from the `mybundle.zip` file into the `tmp` subfolder. The `-d` option of the `unzip` command creates the subfolder to which the files are extracted.

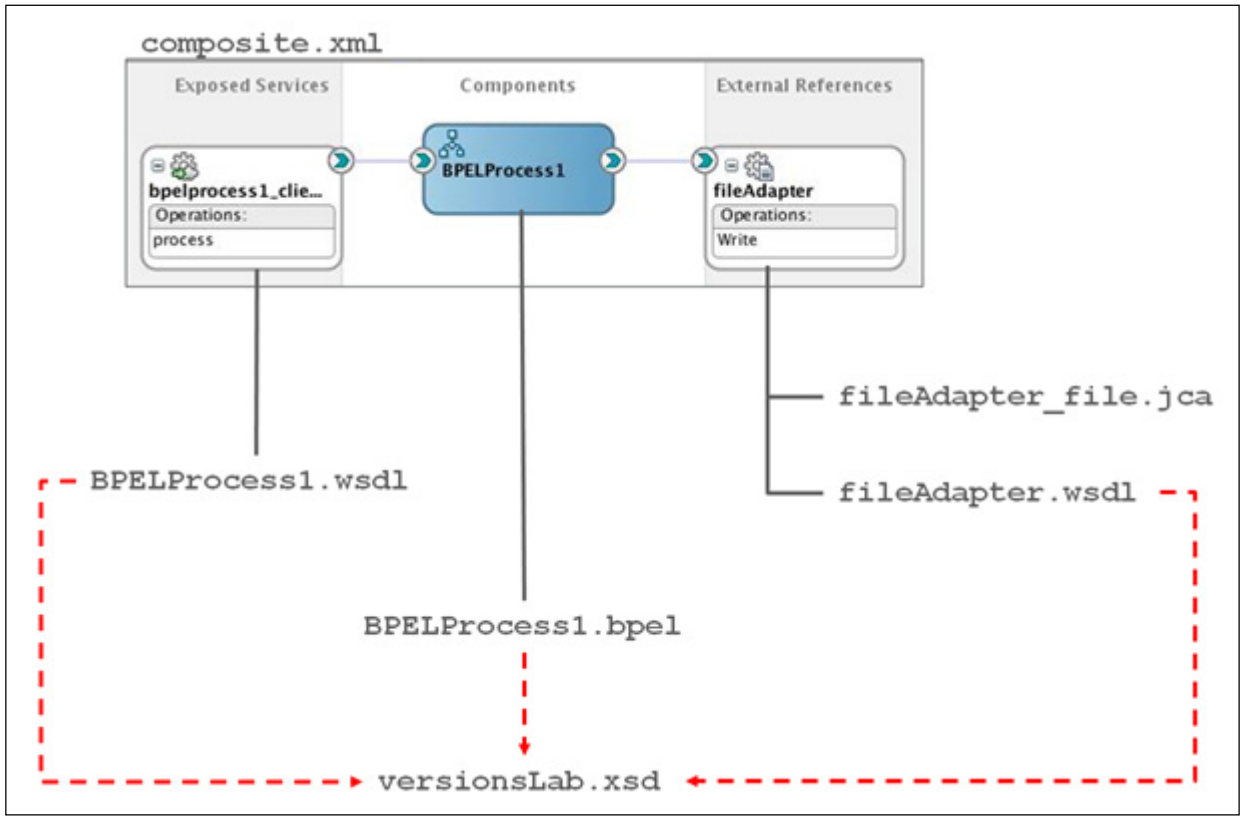
Unzipping Each of the SAR Files

2. In the same Terminal window, enter the following commands:

   ```
   $ cd tmp
   $ unzip sca_Versioning_rev1.0.jar -d 1
   $ unzip sca_Versioning_rev2.0.jar -d 2
   $ unzip sca_Versioning_rev3.0.jar -d 3
   ```

   Note: The contents of each of the `jar` files are extracted to the subfolders `1`, `2`, and `3`, respectively.

The file adapter in each of the applications writes the output to a different location. Each composite has used a different method to configure this behavior. In the next task, you use the following diagram as reference to explore the contents of each of the SAR files. The composite applications that you deployed are represented by the model at the top of the diagram. The bottom portion of the diagram lists the files that were created with each component. The red dashed lines indicate references among the files, and from that, the underlying relationships among the components.



The Files

- The `BPELProcess1.wsdl` file describes how to call the composite application. It includes a reference to the `versionsLab.xsd` file, which describes the input and output message formats of the composite.
- The `BPELProcess1.bpel` file includes the definition of the business process. It also includes a reference to the `versionsLab.xsd` file, which describes the input and output message formats of the business process.
- The `fileAdapter_file.jca` file contains the adapter implementation details. In the case of a file adapter that is configured to write data, the details include an output directory name and a file naming pattern, as well as other information.
- The `fileAdapter.wsdl` file includes the abstract WSDL information about the service that the adapter provides. It also references the `versionsLab.xsd` file, which describes the input and output message formats of the adapter.
- A `composite.xml` file is automatically created when you create a SOA project. This file describes the entire composite assembly of services, service components, references, and wires.

- When a file adapter is defined in a composite, an `<import>` statement is added to the `composite.xml` file, which makes the `fileAdapter.wsdl` file contents available.
- A `<reference>` element is also added to the `composite.xml` file. It aggregates references to the adapter `.jca` and `.wsdl` files, as well as the binding information in place of the concrete information in a WSDL.

- A `configurationPlan.xml` document (not shown) can be used at deployment time to specify values that may vary among the development, test, and production environments. In this example, the values might include the output directory of the file adapter.

3. Use gedit and the preceding diagram to explore the contents of each of the SAR files that you just extracted. Determine the following:
   - Which composite writes its output to a hard-coded path name? What directory will it write to?
   - Which composite uses a logical name for the path that is resolved at the composite level? What directory will it write to?
   - Which composite uses a configuration plan to specify the path name? What directory will it write to? How does the configuration plan leverage the logical name?

Verifying the Output Directory of Each Application

4. Log in to the Oracle Enterprise Manager Fusion Middleware Control 12*c* application and test the applications:
   a. Open a web browser and provide the URL http://admin.example.com:8080/em.
   b. Log in with the username `weblogic` and the `weblogic` user password.
   c. Navigate to the test console for the first of the deployed applications.
   d. Provide any text that you wish as input.
   e. Verify that the application writes to the output directory that you anticipated in the previous task.

Forcing Version 2 of the Application To Be the Default

5. To force version 2 of the application to be the default version, perform the following steps:
   a. Make a copy of the `antDeploy.sh` script.

```
$ cd /practices/practice09/scripts
$ cp antDeploy.sh myantDeploy.sh
```

   b. Use gedit to modify the `myantDeploy.sh` script to redeploy version 2 and force it to be the default version.
      1) In the `SET ANT PARAMETERS` section of the script, modify the value of `SAR_LOCATION`.

```
SAR_LOCATION=/practices/practice09/tmp/sca_Versioning_rev2.0.
jar
```

      2) In the `SET ANT PARAMETERS` section of the script, uncomment the `FORCE_DEFAULT` variable assignment.

```
FORCE_DEFAULT=true
```

```
38  # ================================================================
39  # SET ANT PARAMETERS
40  # ================================================================
41    ANT_SCRIPT=$MW_HOME/soa/bin/ant-sca-deploy.xml
42    SERVER_URL=http://soainternal.example.com:8080
43    SAR_LOCATION=/practices/practice09/tmp/sca_Versioning_rev2.0.jar
44    OVERWRITE=true
45    USER=weblogic
46    PARTITION=myPartition
47    FORCE_DEFAULT=true
48  }
```

3) In the `INVOKE ANT` section of the script, add the following text to the command line:

`-DforceDefault=true`

```
76  # ================================================================
77  # INVOKE ANT
78  # ================================================================
79
80  invokeScript()
81  {
82    echo "You are about to execute the following command:"
83    display
84    echo ---------------------------------------
85    prompt "Press any key when ready: "
86    read ANYKEY
87    $ANT_CMD -f $ANT_SCRIPT -DserverURL=$SERVER_URL -DsarLocation=$SAR_LOCATION -Doverwrite=
      $OVERWRITE -Dpartition=$PARTITION -Duser=$USER -DforceDefault=$FORCE_DEFAULT
88  }
```

c. Save your work and close gedit.

d. Invoke the script to redeploy the application.

```
$ ./myantDeploy.sh
```

e. Provide the `weblogic` username and password when prompted.

The script deploys the bundle.

```
[deployComposite] Username:
weblogic
[deployComposite] Password:

[deployComposite] INFO: Received HTTP response from the server, response code=20
0
[deployComposite] ---->Composite deployment produced 0 warning/severe messages
[deployComposite] ---->Deploying composite success.

BUILD SUCCESSFUL
Total time: 13 seconds
```

f. Use the List Composites in Partition option of the `antPartitions` script to verify that version 2 of the application is now the default.

```
     [java] Following 3 composites are currently deployed to the platform, in pa
rtition: myPartition.
     [java]
     [java] 1. Versioning[3.0], partition=myPartition, mode=active, state=on, is
Default=false, deployedTime=2015-03-09T15:43:23.708Z
     [java] 2. Versioning[2.0], partition=myPartition, mode=active, state=on, is
Default=true, deployedTime=2015-03-10T11:01:13.332Z
     [java] 3. Versioning[1.0], partition=myPartition, mode=active, state=on, is
Default=false, deployedTime=2015-03-09T15:43:21.874Z
     [java]
```

Managing the Applications in the Partition

6. Use the `antPartitions.sh` script to perform the following:
   a. Shut down all the composites in the partition `myPartition`
   b. Retire all the composites in the partition `myPartition`
   c. Activate all the composites in the partition `myPartition`
   d. Undeploy all the applications and delete `myPartition`

```
     [java] connection initiated
     [java] folderMBean=oracle.soa.config:name=soa-infra,j2eeType=FolderLifecycl
eConfig,Application=soa-infra
     [java] Partition (myPartition) is successfully deleted.
```

7. Use the `antPartitions.sh` script to verify successful deletion of the partition.

```
[java] Following 1 partitions are currently available on the platform:
[java]
[java] 1. default
[java]
```

Practices for Lesson 10: Administering the SOA Run Time

Chapter 10

Practices for Lesson 10

Practices Overview

In this practice, you use an Ant script to deploy a pair of composite applications to the SOA server. A second script generates a series of data files that provide input to the applications. You use the Enterprise Manager web application to monitor the sensors, events, analytics, faults, and more.

Practice 10-1: Deploying the Composite Applications

Overview

In this practice, you use an Ant script to deploy a pair of composite applications to the SOA server. You then use a shell script to generate input data for the deployed applications to process.

Assumptions

- You have successfully completed all previous practices.
- Oracle WebLogic Server is running.

Tasks

Deploying the Composite Applications

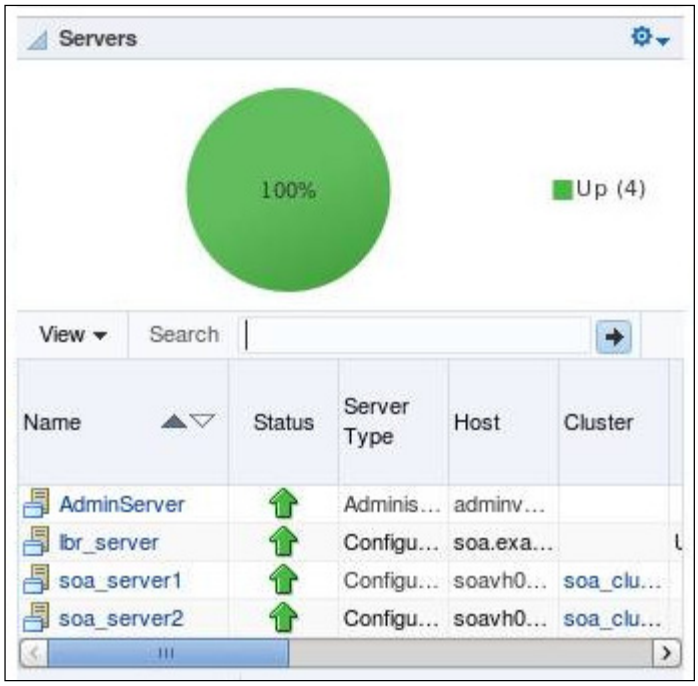
1. Deploy the applications.
   a. Open a Terminal window.
   b. In the open Terminal window, issue the following commands:
      ```
      $ cd /practices/practice10/scripts
      $ ./deployApps.sh
      ```
   c. When prompted, supply username `weblogic` and the password for the `weblogic` user.
      (You will be prompted once for each of the two applications.) The script runs and reports successful deployment of the applications.

Creating the Data Script

2. Execute the following commands in a Terminal window:
   ```
   $ cd /practices/practice10/scripts
   $ nohup ./makeData.sh &
   ```
   This script creates an input file for the deployed application every 10 seconds for the next 10 minutes, for a total of 60 inputs. By design, every 20th input generates a fault. In the remainder of this practice, you monitor the various behaviors and statistics of the applications and their respective BPEL and mediator components. You observe the faults, sensors, analytics, and events generated by these applications and components.

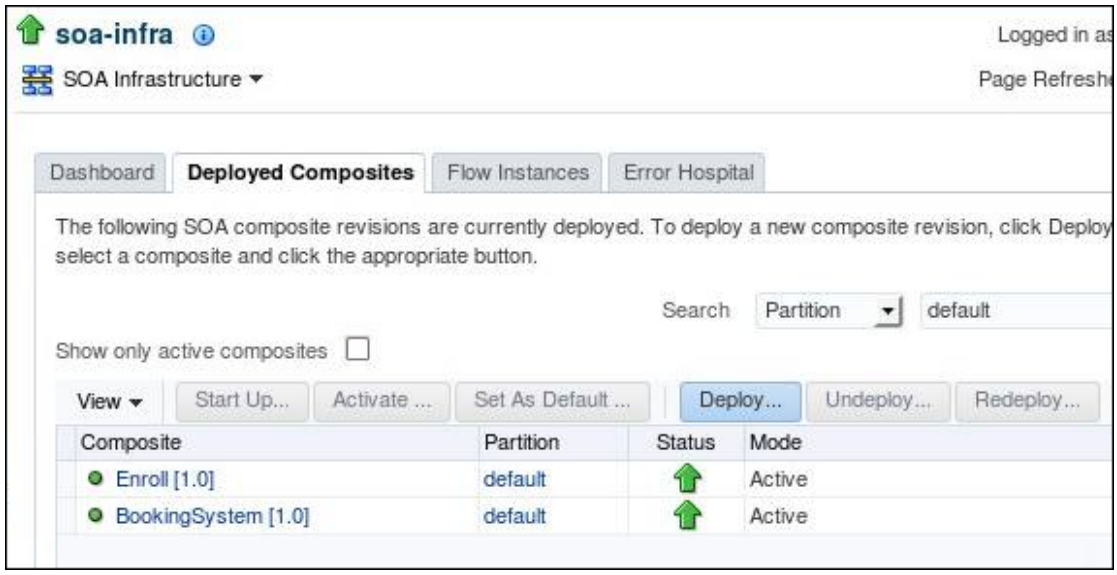
Verifying That All Servers Are Running

3. Complete the following steps:
   a. In a web browser window or on a tab page, sign in to Oracle Enterprise Manager Fusion Middleware Control (URL http://admin.example.com:8080/em) with the `weblogic` user credentials.
   b. In the Target Navigation pane, navigate to WebLogic Domain > edg_domain.
   c. On the edg_domain page, examine the Servers pane and verify that all the servers are up.



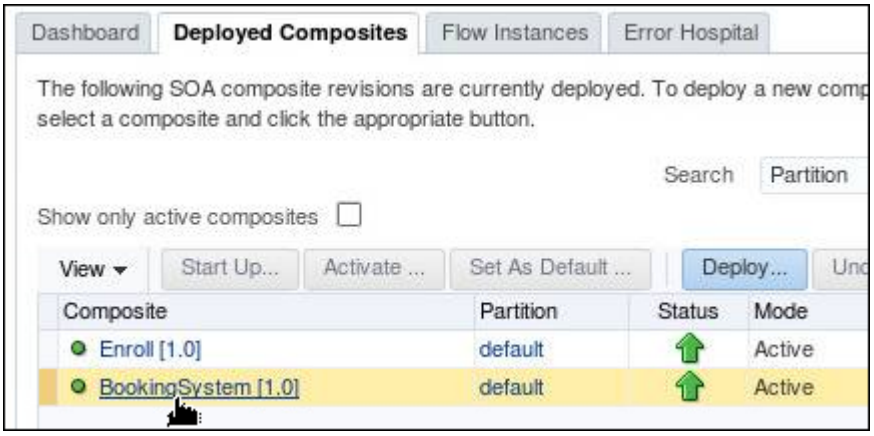
Verifying That Deployed Applications Are Active

4. Complete the following steps:
   a. In the Target Navigation pane, navigate to SOA > soa-infra (soa_server1).
   b. On the soa-infra page, click the Deployed Composites tab.
   c. Verify that all the composites listed in the Composite column are Active in the Partition that is named default.

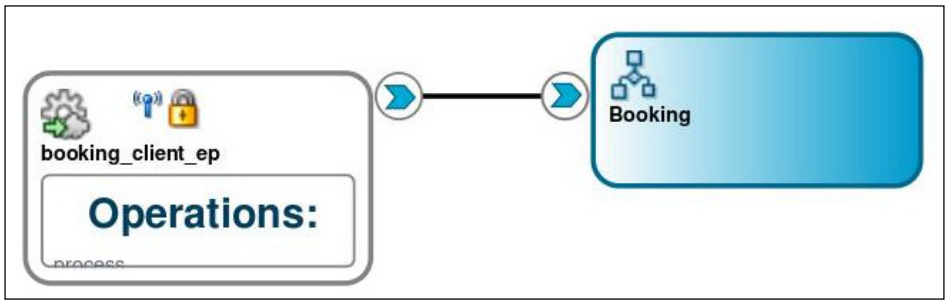


Examining the Definition of the BookingSystem Application

5. Complete the following steps:
   a. Click the BookingSystem [1.0] composite.



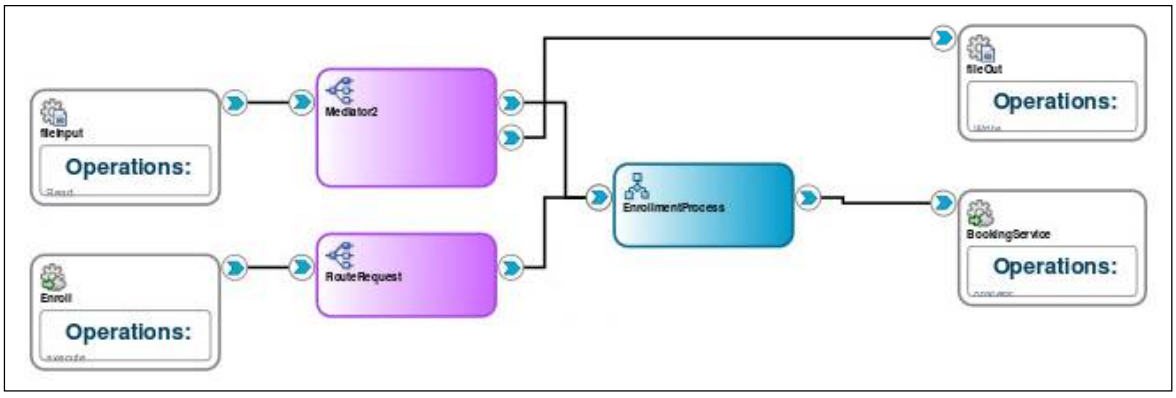
b. On the BookingSystem page, click the Composite Definition tab.



The `BookingSystem` project consists of a BPEL process that provides a simple mechanism to simulate a training enrollment booking system. It exposes a service, which receives enrollment-related messages. In most cases, it responds with positive confirmations, but when provided with certain data values, it returns a fault.

Examining the Definition of the Enroll Application

6. Repeat the previous steps to display the composite definition for the Enroll [1.0] composite.



The `Enroll` project uses a file adapter (fileInput) to read enrollment messages from a specified directory. The messages are then passed via a mediator (Mediator2) to a BPEL process (EnrollmentProcess). The process then invokes the BookingSystem application (BookingService). When a response is received from the BookingSystem project, it is passed to the mediator (Mediator2), which passes the message to an outbound file adapter (fileOut) that writes the message to a file.

Note that the Enroll entry point, which is shown in the diagram, provides an operation that is exposed through one of the web service interfaces for this composite. You do not use this interface during this practice.

Examining the Configuration of the Enroll Application

7. Examine the application configuration by performing the following steps:
   a. Display the Dashboard tab of the Enroll application home page.
   b. Use the links in the application's Services and References entries to determine the input directory for the `fileInput` file adapter.
   c. Determine the file naming convention for the `fileOut` file adapter. How many messages will be written to each file?

Examining the Flow Trace of a Successful Instance

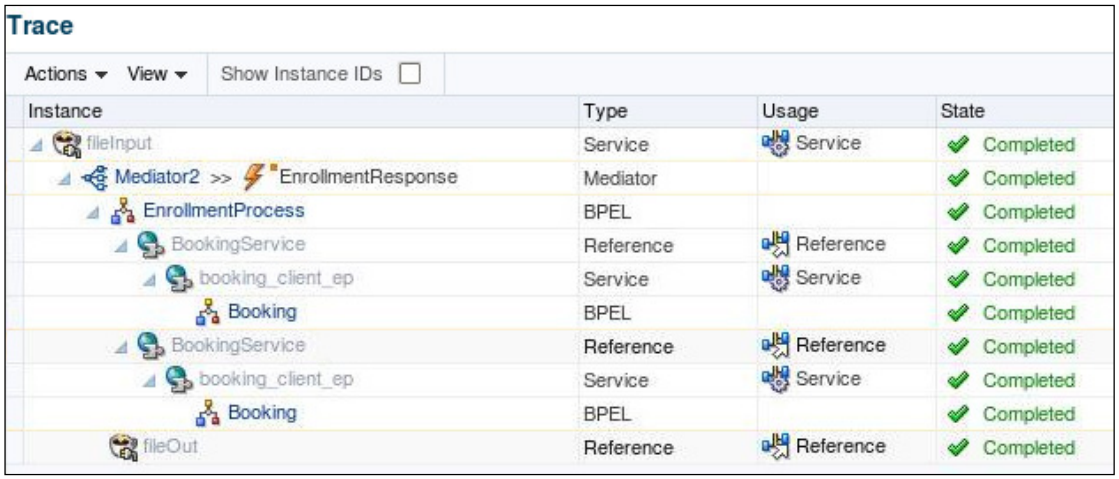
Several minutes should have passed since you initiated the script to pass instances to the Enroll application (step 2). A number of instances should have succeeded, and a few should have raised faults by now. In the next tasks, you examine both the successful and faulted instances of the application.

8. Locate the composite application instance and view the Flow Trace page.
   a. Click the Flow Instances tab of the Enroll application home page.

      Note: Click Search to populate the Flow Instances tab, if empty, and click the Search icon to close the Search Options panel. Alternatively, click the Recent Instances link above the Search Results table.
   b. Click an Instance ID with a Flow State of Completed.

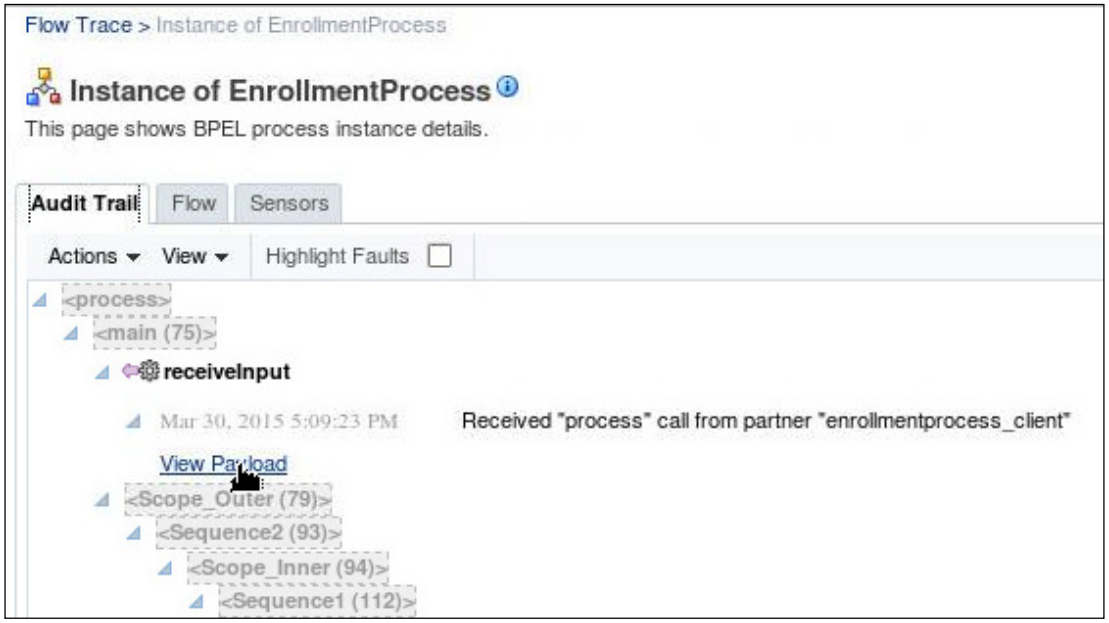
Note: Click the Recent Instances link or the Search (magnifying glass) icon as needed to display the most current instances.

The Trace pane for the selected instance is displayed.



c. In the Trace pane, click EnrollmentProcess.

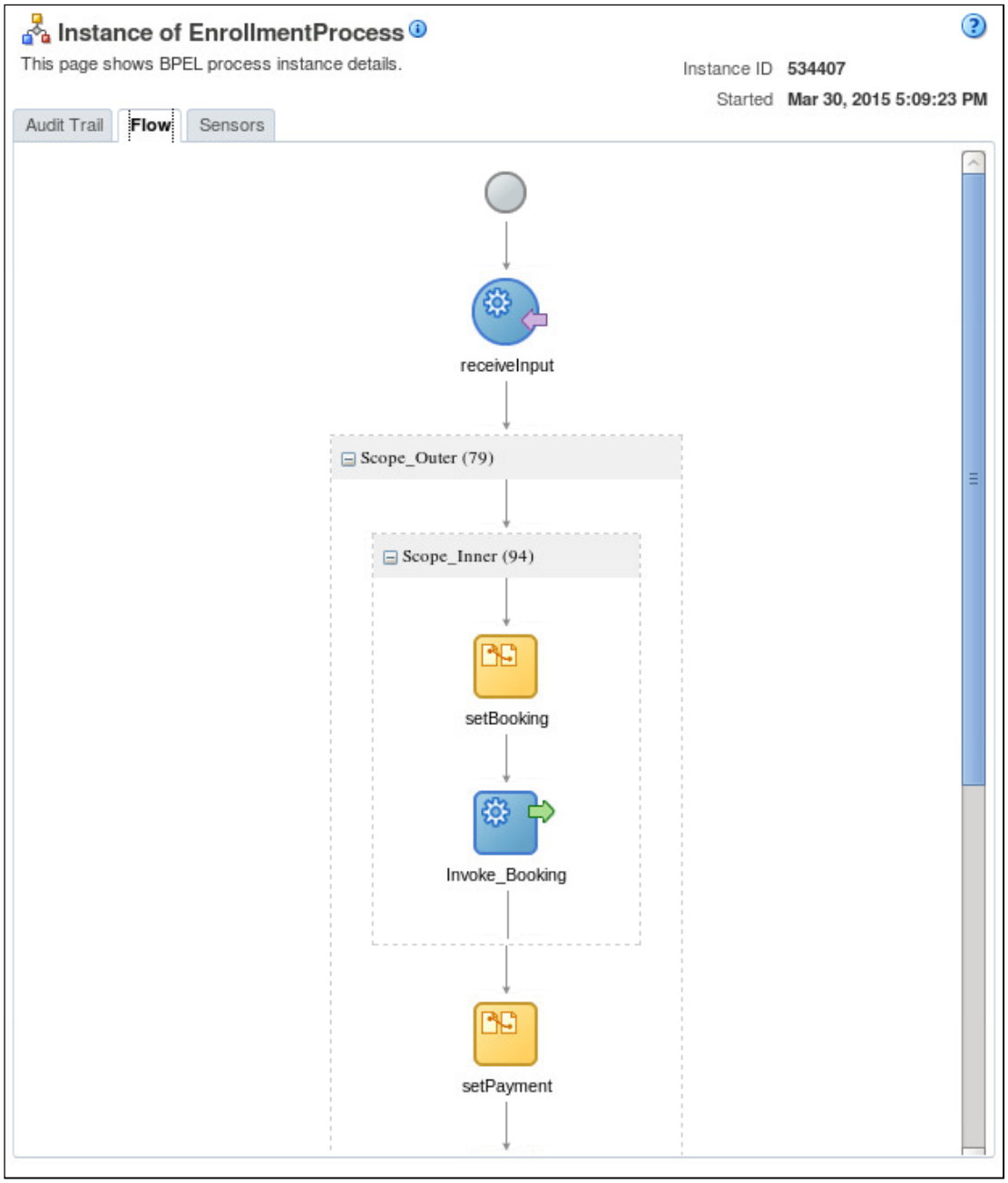
d. From the application instance Flow Trace page, display the incoming payload.



- What is the incoming message type?
- What is the ID number of the message?

Click the Flow Tab to display the BPEL process instance details.

A portion of the instance graphic is displayed as follows. Individual process activities are represented in the model.



e. Return to the Flow Trace page.

f. Click the Composite Sensor Values tab.

- What is the value of the incomingMsg Sensor?
- Compare this to the message payload that you viewed earlier. Which field of the message is the sensor tracking?

g. Click the Faults tab.

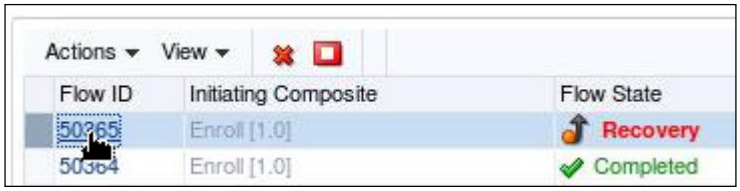
- What is the name of the Event published by Mediator2?

  (You discuss events in more detail in the lesson titled “Administering Business Events.”)

h. Close the Flow Trace page.

Examining the Flow Trace of a Faulted Instance

9. Complete the following steps:
   a. Return to the Enroll application home page.
   b. Click the Flow Instances tab.
   c. Click an Instance ID with a Flow State of **Recovery**.

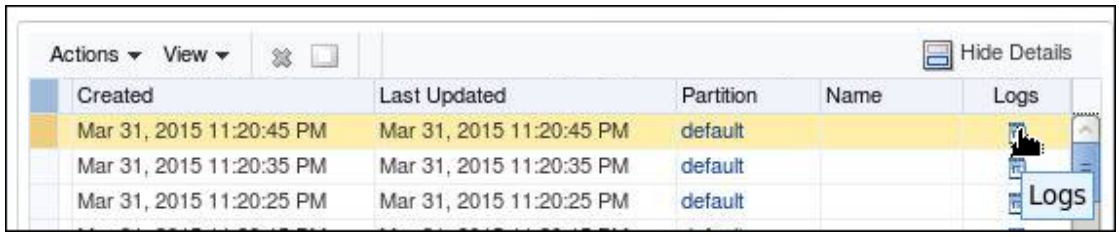


- What is the error message number?

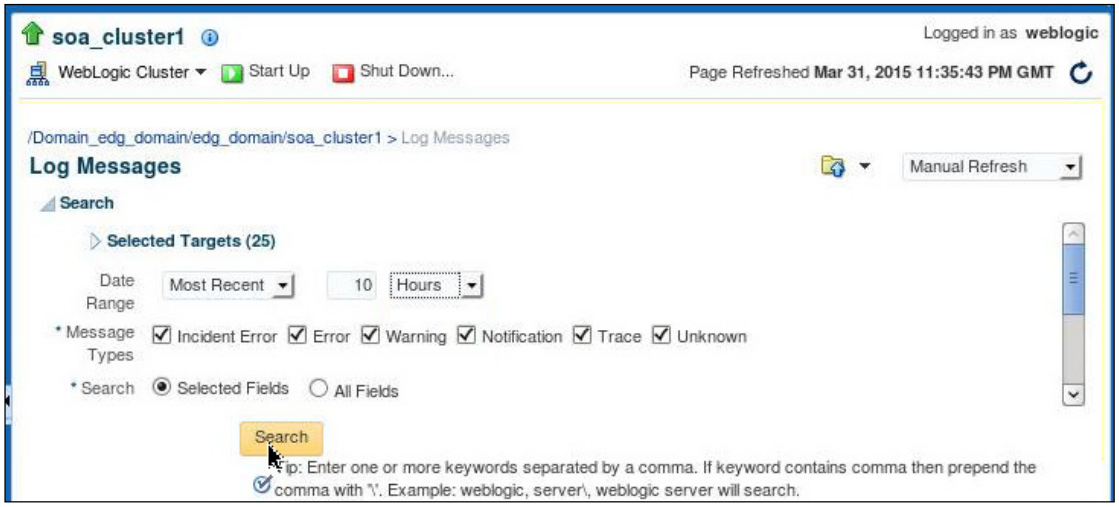
   d. Abort the instance.

Viewing the Log Files for a Given Instance

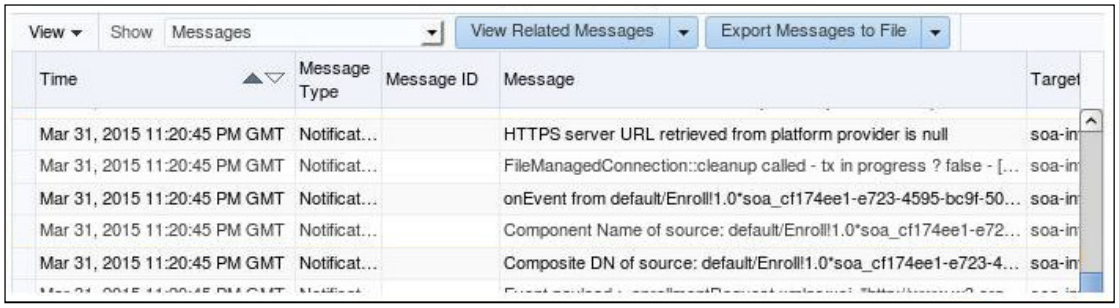
10. Complete the following steps:
    a. On the Enroll Dashboard page, click the Flow Instances tab.
    b. For a selected instance, scroll the table all the way to the right to find the Logs icon.



    c. Click the icon to display the logs search page.
    d. Modify the search criteria (specifically the date range), and click Search.



e. Explore the resulting list of log messages.



Practice 10-2: Exploring the Features of the Error Hospital

Overview

In this practice, you explore additional features of the Error Hospital for management of error conditions.

Assumptions

- You have successfully completed Practice 10-1.

Tasks

Generating Additional Faults

1. Shut down the BookingSystem project.
   a. In the Enterprise Manager Target Navigation pane, expand soa-infra > default.
   b. Select the BookingSystem project.
   c. In the BookingSystem pane, click Shut Down.

   d. In the Confirmation dialog box, click Yes.

   The project is shut down.

2. Execute the following commands in a Terminal window:

```
$ cd /practices/practice10/scripts
$ nohup ./makeData.sh &
```

   As before, this script creates an input file for the deployed application every 10 seconds for the next 10 minutes, for a total of 60 inputs. This time, because the BookingSystem application is unreachable, every message generates a remote fault. In subsequent steps, you use the bulk recovery and bulk abort features to address these faults.

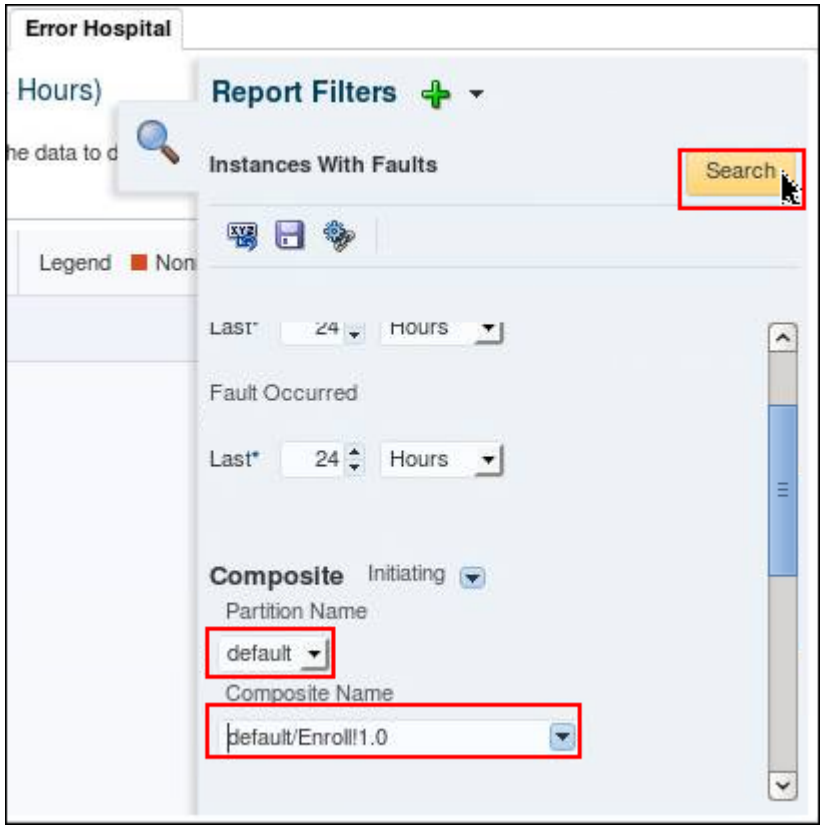
Accessing the Error Hospital

3. Complete the following steps:
   a. In the Enterprise Manager Target Navigation Pane, expand SOA > soa-infra.

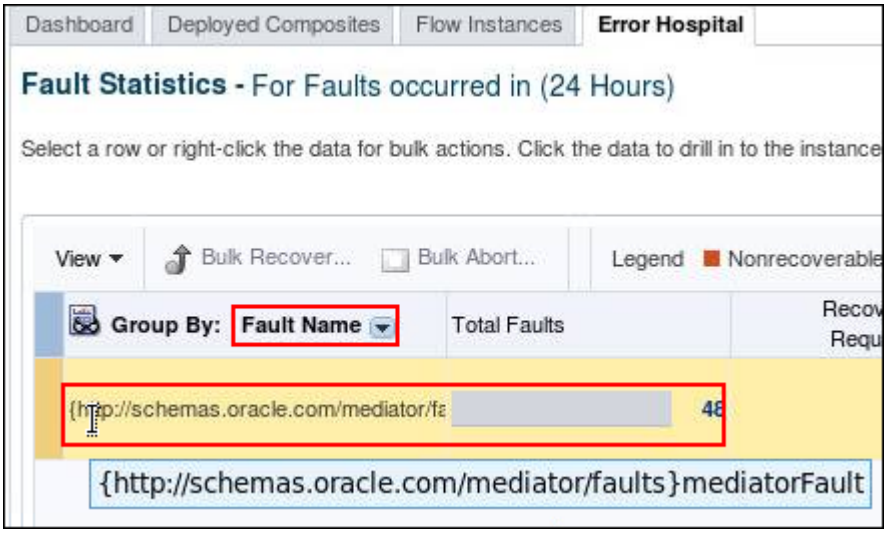
b. Click soa-infra (soa_server1).
c. Click the Error Hospital tab.

Viewing Faults

4. Set and apply report filters:
   a. Select the following options:
      - Partition Name: default
      - Composite Name: default/Enroll!1.0
   b. Click Search.



5. Confirm that there is a single entry for the Fault Name: {http://schemas.oracle.com/mediator/faults}mediatorFault.
   Note: If required, select Fault Name from the Group By drop-down menu.



Invoking Bulk Recovery

6. Restart the BookingSystem project and recover the faults.
   a. In the Enterprise Manager Target Navigation pane, expand soa-infra > default.
   b. Select the BookingSystem project.
   c. In the BookingSystem pane, click Start Up.

   d. Click Yes to confirm the Start Up action.
   e. Return to the Error Hospital page, and initiate a search for faults (if required).
   f. Select Bulk Recovery.

   Note: A majority of the faults are shown as recovered. (These are the remote faults generated by shutting down the BookingSystem application.) The remaining faults, which were originally thrown by the BookingSystem application, remain. The following example shows that 57 faults were recovered and 6 remain.