

Custom Approval Process for Self-Register User Scenario

1. Overview

This sample illustrates how customer can create their custom approval process and use it for approval process of Self-Register User requests.

This sample is developed on Oracle Identity Manager 11g (11.1.1.3.0)

2. Scenario

This sample illustrates Self-Register User approval scenario. The task at request level of approval should be assigned to SYSTEM ADMINISTRATORS role. The task at operation level of approval should be assigned to Organization administrators depending on the organization value provided at Request level of approval.

3. Requirements/Pre-Requisites

You will need the following software installed on your machine before you start working on this use case.

1. Oracle Identity Manager (OIM) Managed Server 11g (11.1.1.3.0).
2. JDeveloper 11g (11.1.1.3.0) with Oracle SOA Composite Editor extension.

From [here](#) download the zip file that contains the files that are required for this example.

You will also need to perform the following tasks from OIM Admin console:

1. Create an organization FINANCE.
2. Create a role FINANCE_APPROVERS.
3. Create an end-user Danny Crane.
4. Assign FINANCE_APPROVERS role to Danny Crane.
5. Assign FINANCE_APPROVERS role to Administrative roles for FINANCE organization
6. Create a user Shirley Schmidt.
7. Assign SYSTEM ADMINISTRATORS role to Shirley Schmidt.

You will also need to perform the following tasks from Oracle Enterprise Manager (EM) console:

1. Store the credentials of Shirley Schmidt in CSF. These credentials are read in Java Embedding activity that will be used while creating custom composite. (Refer to Appendix B for details on how to store credentials in CSF).

4. Acronyms

BEAHOME	Path to middleware directory in OIM installation
OIMHOME	Path to OIM home directory (e.g <BEAHOME>/Oracle_IDM1)
SOAHOME	Path to SOA home directory (e.g <BEAHOME>/Oracle_SOA1)
WLS_DOMAIN	Weblogic domain name (e.g base_domain)
SOA_SERVER	Name of SOA managed server

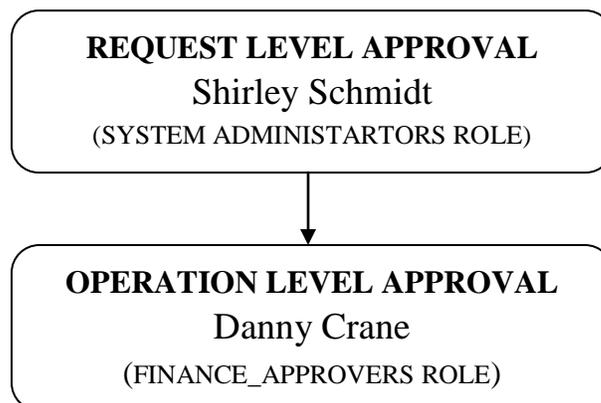
5. Designing custom approval process

1. Who should be the approver?

At operation level of approval, the approvers need to be the administrators of the organization to which the user is registered.

2. Where this approval process should be used?

For this scenario, the custom approval process will be used for all Self-Register User type requests at Operation level of approval.



6. Developing custom approval process

By default, Self-Register User tasks are assigned to xelsysadm at operation level of approval. As per the scenario, all the Self-Register User tasks at request level must be assigned to SYSTEM ADMINISTRATORS role; and all the Self-Register User tasks at operation level must be assigned to the administrator of that organization to which the user is assigned during Request level of approval.

For request level of approval, we can use the DefaultRoleApproval composite that is shipped along with OIM. DefaultRoleApproval approval process assigns task to SYSTEM ADMINISTRATORS role.

For operation level of approval, we need to create a custom approval process. To create a custom approval process, we need to perform the following steps:

1. Creating a SOA composite application
2. Modifying SOA composite in JDeveloper
3. Deploying SOA composite to Oracle SOA Managed Server
4. Registering SOA composite with OIM Managed Server
5. Creating an Approval Policy in OIM
6. Testing of custom approval process

The following sections will take you through each of these steps in detail.

Note: In case you commit any mistakes while executing these steps and wish to re-run the sample, refer to Appendix C on how to cleanup and re-run.

6.1. Creating a SOA composite application

OIM provides a helper utility for creating custom SOA composites. This utility creates a template SOA project that adheres to all the necessary standards.

Create a JDeveloper application for custom SOA composite by running the helper utility:

1. Set up the environment (for Linux machines)
 - `cd <BEAHOME>/wlserver_10.3/server/bin`
 - `bash`
 - `source setWLS.env.sh`
2. Run the utility by executing following commands:
 - `cd <OIMHOME>/server/workflows/new-workflow`

- `ant -f new_project.xml`
3. Enter the JDeveloper application name (SelfRegistrationApprovalApp) when the following prompt is displayed:
Please enter application name
 4. Enter the JDeveloper project name (SelfRegistrationApproval) when the following prompt is displayed:
Please enter project name
 5. Enter the name of the ADF binding service (SelfRegistrationApprovalService) for the composite when the following prompt is displayed:
Please enter the service name for the composite. This needs to be unique across applications

The following screenshot (Figure 1) shows creation of SelfRegistrationApprovalApp.

```

Shell - Konsole
Session Edit View Bookmarks Settings Help
ade:[ abhshsha_iam3 ] [abhshsha@adc2100686 new-workflow]$ ant -f new_project.xml
Buildfile: new_project.xml

new_project:
  [unzip] Expanding: /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/ApprovalApp.zip into /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template
  [input] Please enter application name
SelfRegistrationApprovalApp
  [input] Please enter project name
SelfRegistrationApproval
  [input] Please enter the service name for the composite. This needs to be unique across applications
SelfRegistrationApprovalService
  [mkdir] Created dir: /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp
  [copy] Copying 30 files to /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp
  [copy] Copied 21 empty directories to 5 empty directories under /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp
  [move] Moving 1 file to /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp
  [move] Moving 28 files to /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp/SelfRegistrationApproval
  [move] Moving 1 file to /scratch/abhshsha/shiphome/beahome/Oracle_IDM1/server/workflows/new-workflow/process-template/SelfRegistrationApprovalApp/SelfRegistrationApproval

BUILD SUCCESSFUL
Total time: 13 seconds
ade:[ abhshsha_iam3 ] [abhshsha@adc2100686 new-workflow]$

```

Figure 1: New SOA Composite Application

The new application is created in the `OIMHOME/server/workflows/new-workflow/process-template/` directory. You can now open the new application in JDeveloper for modification.

6.2. *Modifying SOA composite in JDeveloper*

The JDeveloper application created by the helper utility in previous section is actually a SOA application. It has a composite which by default assigns task to user “xelsysadm”. We will have to modify this SOA composite to assign task to the administrators of that organization to which the user is assigned in Request level of approval. To modify the newly created SOA composite follow these steps:

1. Open the application (SelfRegistrationApprovalApp.jws under SelfRegistrationApprovalApp directory) in JDeveloper.
2. Open composite.xml file under SelfRegistrationApproval→SOA Content. Define a property bpel.preference.oimurl at the following location in composite.xml

```
<component name="ApprovalProcess">  
<implementation.bpel src="ApprovalProcess.bpel"/>  
<property name="bpel.preference.oimurl">t3://oim_host:oim_port </property>  
</component>
```

This property will store OIM’s url which will be used later in Java Embedding activity in the composite to login to OIM using OIMClient. Adding this property in composite.xml will allow us to change its value at runtime (from EM console) rather than hard coding it. For this sample we will set the actual value of oimurl at the time of testing the composite.

3. Open ApprovalProcess.bpel. Switch to design view. Click on “(x)” inside the approval process to see variables in this approval process (see Figure 2).
4. Click on “+” to add a variable (see Figure 2). Give name as oimurl, choose type as “Simple Type” and set it to String. This variable will read OIM url from the property mentioned in step 2 and it can be used in Java code.
5. Add another variable orgadmin, choose type as “Simple Type” and set it to String. This variable will be used in Java Embedding Activity to store the Organization Administrators, the value for which can be obtained using OIM APIs.

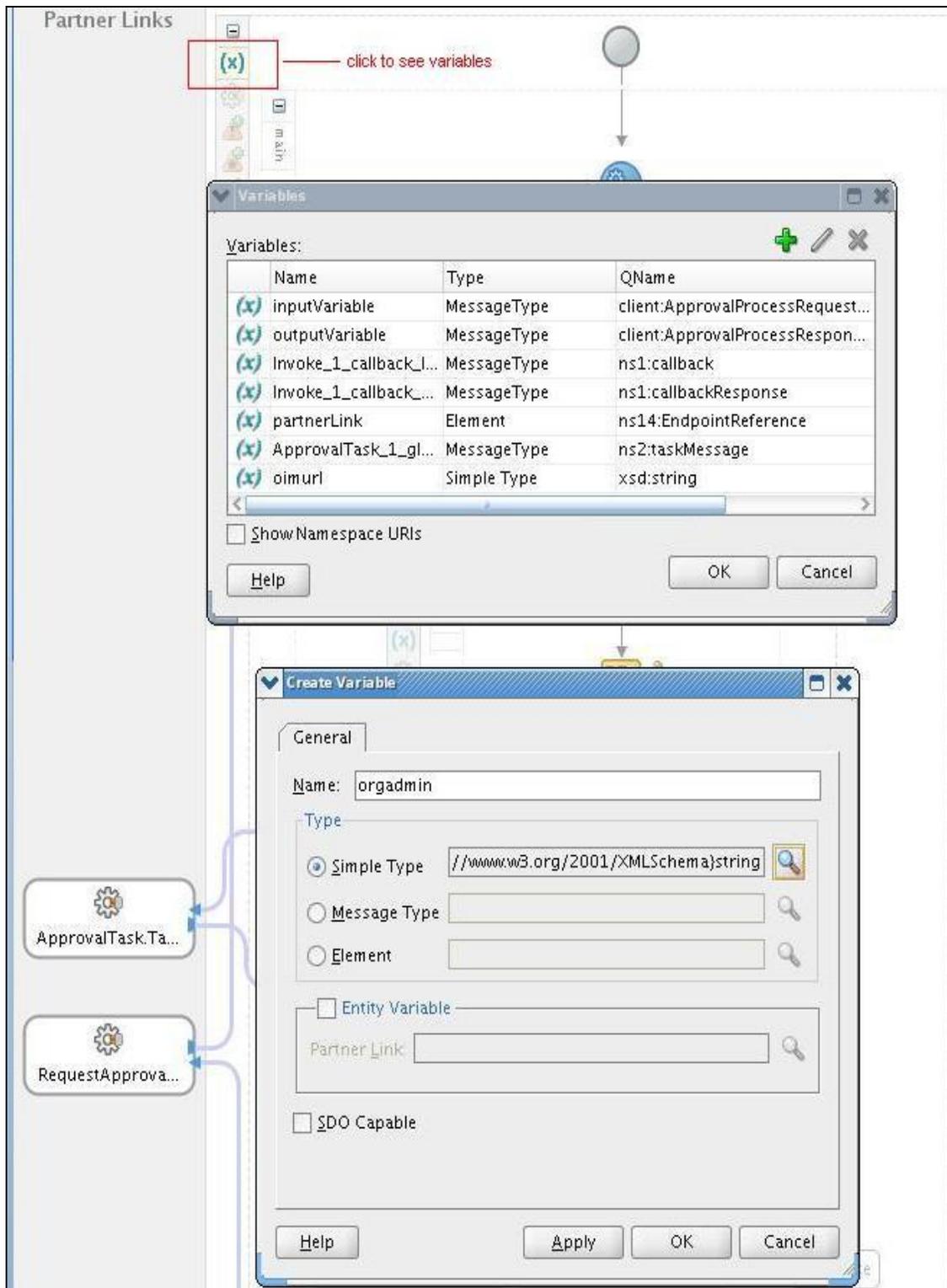


Figure 2: Add variables to approval process

- From the components palette →BPEL Activities, drag and drop Assign activity after receive input activity (see Figure 3). Rename this activity to Assign_oimurl.

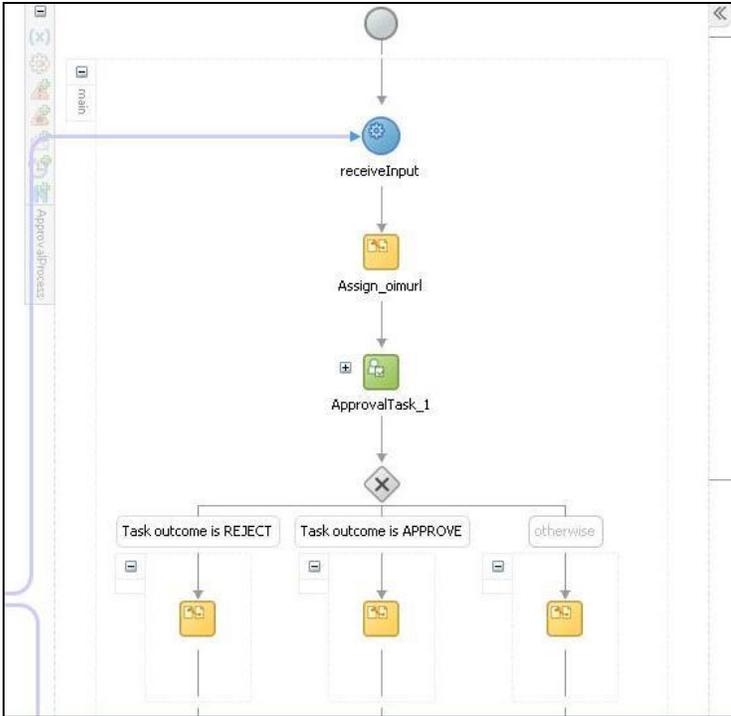


Figure 3: Assign_oimurl activity in approval process

7. Double click on Assign_oimurl activity. Click on “+” sign and select Copy Operation. Create Copy Operation dialog will open (See Figure 4).

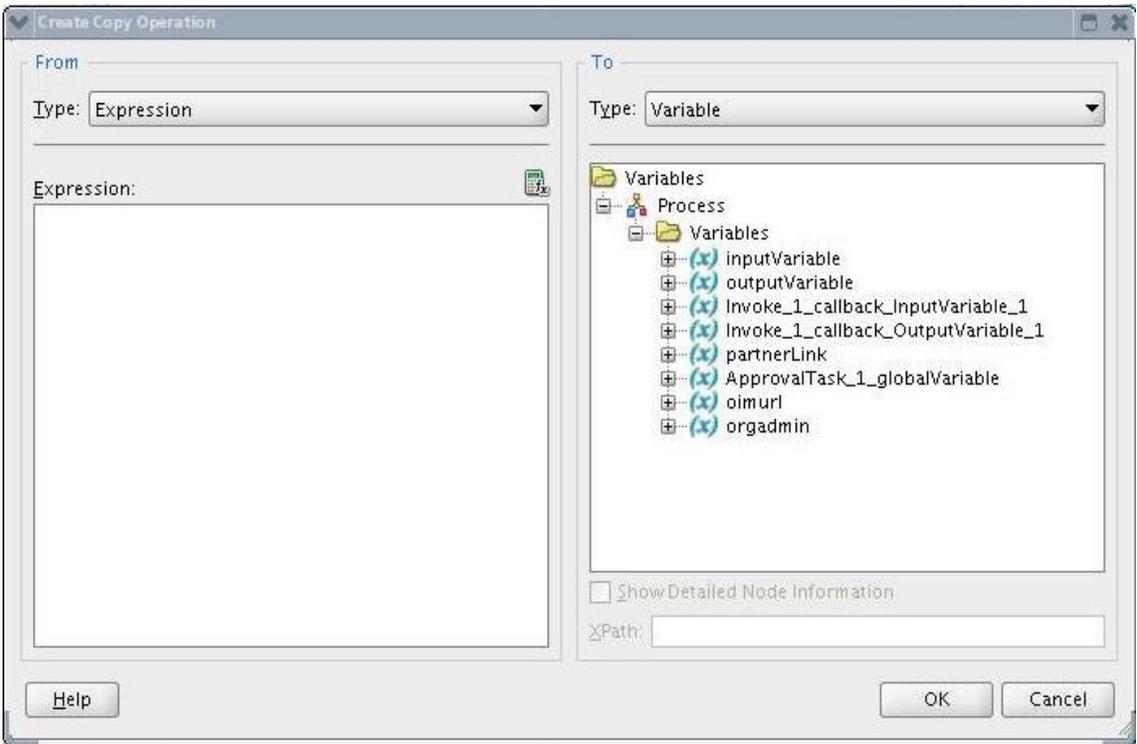


Figure 4: Create Copy Operation dialog

- Under From header choose type as Expression. Click on Expression Builder. Expression builder dialog will open (see Figure 5). Under Functions header, select “BPEL XPath Extension Functions” from the drop down. Select `getPreference` and click on Insert Into Expression. Pass ‘oimurl’ as argument to this function and click OK. This will read the value of ‘bpel.preference.oimurl’ property that is defined in composite.xml.

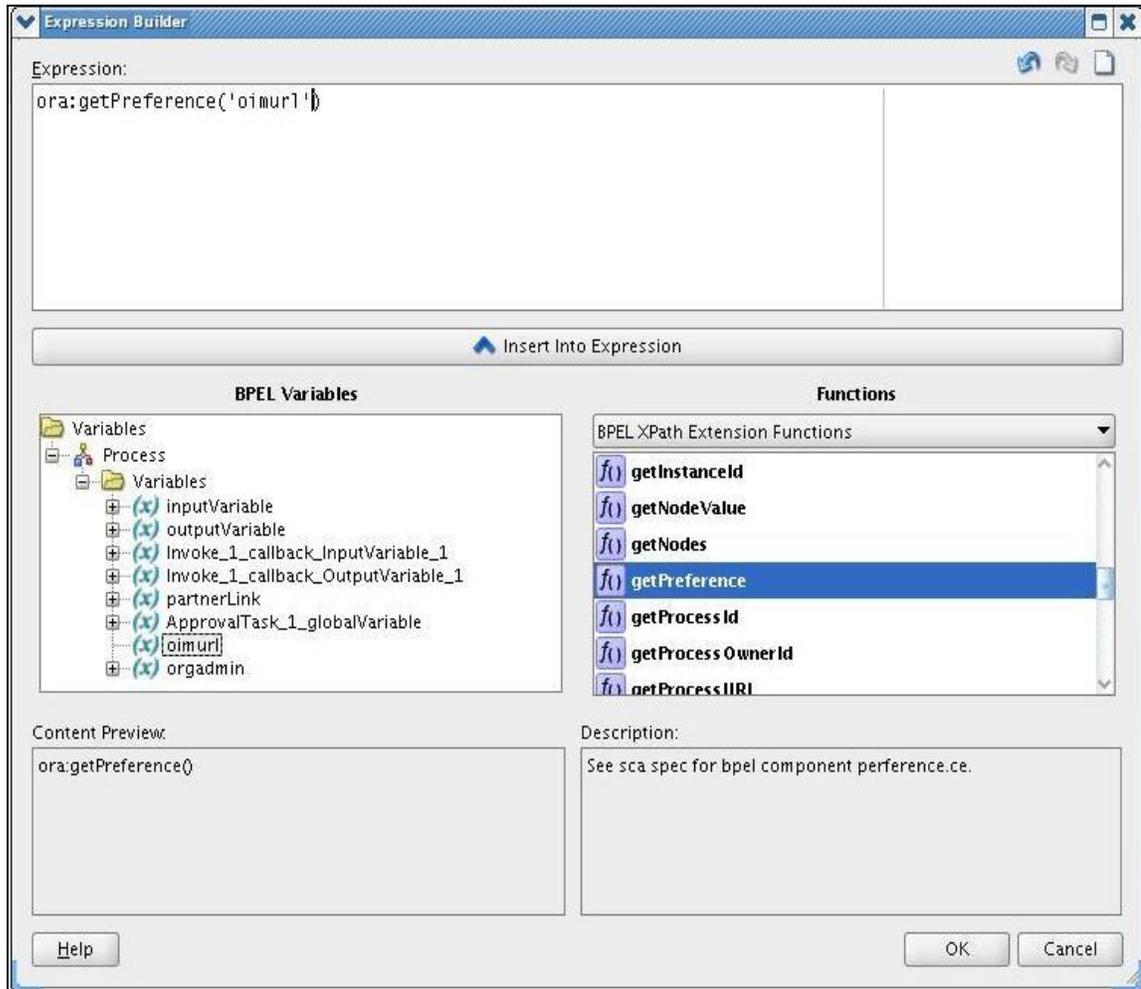


Figure 5: Expression Builder dialog

- In Create Copy Operation dialog select Process → variables → oimurl. Click OK.
- From the components palette → BPEL Activities, drag and drop Java Embedding activity after Assign_oimurl activity (see Figure 6). Rename this activity to GetOrgAdmin.

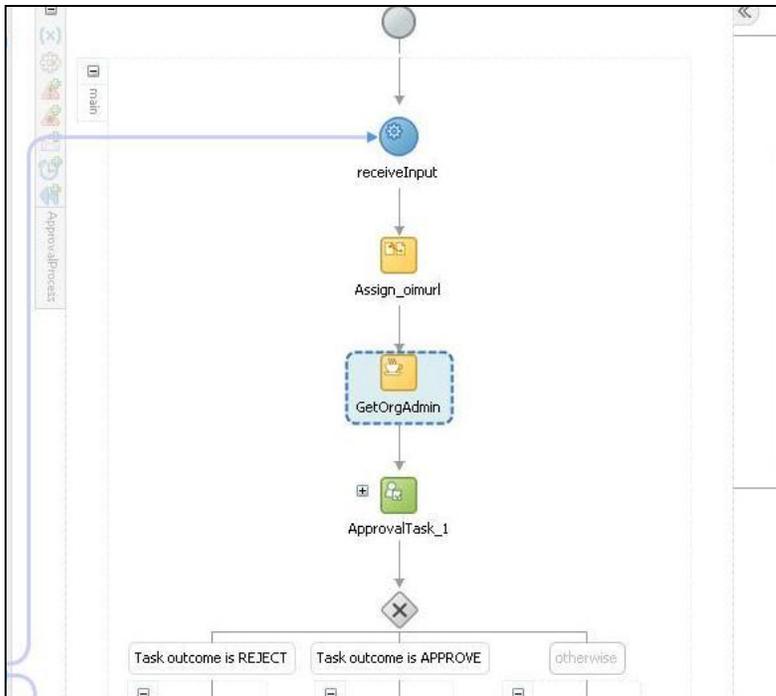


Figure 6: GetOrgAdmin activity

11. Double click on GetorgAdmin activity. Write java code here to get Organization Admin using OIM APIs (Refer Appendix A for the java code and description). Click OK.
12. To use OIM APIs we will have to add oimclient.jar to this composite. Copy oimclient.jar from location OIMHOME/server/client to SelfRegistrationApprovalApp/SelfRegistrationApproval /SCA-INF/lib.
13. We will also be using oracle cwallet to get credentials in java code. For that we will have to add jps-manifest.jar to project library. Right click on SelfRegistrationApproval project and click on project properties. Select “Libraries and Classpath” from the left pane. Click on “Add Jar/Directory” and add jps-manifest.jar from the location BEAHOME/oracle_common/modules/oracle.jps_11.1.1/jps-manifest.jar.
14. Open ApprovalTask.task file under SelfRegistrationApproval→SOA Content
15. Select Data from left pane. Click “+” sign and select “Add string payload”. Give parameter name as OrganizationAdmin. The organization administrators obtained using OIM APIs in step 11 will be passed as string payload to the human task.
16. Go back to ApprovalProcess.bpel. Expand the human task node. Double click on ApprovalTask_1_AssignTaskAttributes.
17. Select the row where payload XML is copied to variable /ns2:initiateTask/task:task/task:payload and click on edit. Add OrganizationAdmin element to the payload as shown below.

```
<payload xmlns="http://xmlns.oracle.com/bpel/workflow/task">
  <RequestID xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <RequestModel xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <RequestTarget xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <url xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <RequesterDetails xmlns="http://xmlns.oracle.com/request/RequestDetails"/>
  <BeneficiaryDetails xmlns="http://xmlns.oracle.com/request/RequestDetails"/>
  <ObjectDetails xmlns="http://xmlns.oracle.com/request/RequestDetails"/>
  <OtherDetails xmlns="http://xmlns.oracle.com/request/RequestDetails"/>
  <RequesterDisplayName xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <BeneficiaryDisplayName xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <Requester xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
  <OrganizationAdmin xmlns="http://xmlns.oracle.com/bpel/workflow/task"/>
</payload>
```

This entry in payload corresponds to the OrganizationAdmin defined in step 15.

18. Click on “+” and select Copy Operation. Create Copy Operation dialog will appear (see Figure 7). Under From header, select orgadmin variable. Under To header, select initiateTask→task:task→task:payload. The xpath would appear as /ns2:initiateTask/task:task/task:payload. Append /task:OrganizationAdmin to this xpath. The xpath for To variable would look as follows /ns2:initiateTask/task:task/task:payload/task:OrganizationAdmin

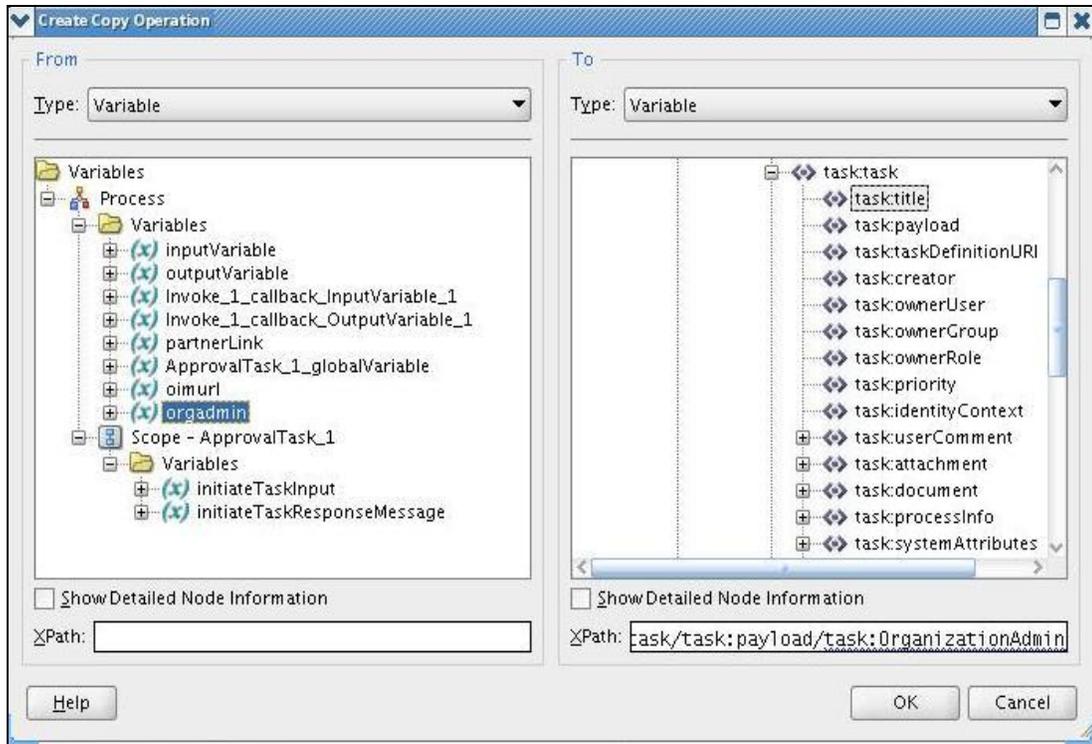


Figure 7: Create Copy Operation

19. Go to ApprovalTask.task. Click on Assignment from left pane.
20. Double click on Stage1.Participant1. Edit Participant Type dialog will appear (see Figure8). Set Identification Type as Group, Data Type as By Expression. Click on Expression builder (“...”) button in value column. Select task:task → task:payload→task:OrganizationAdmin.

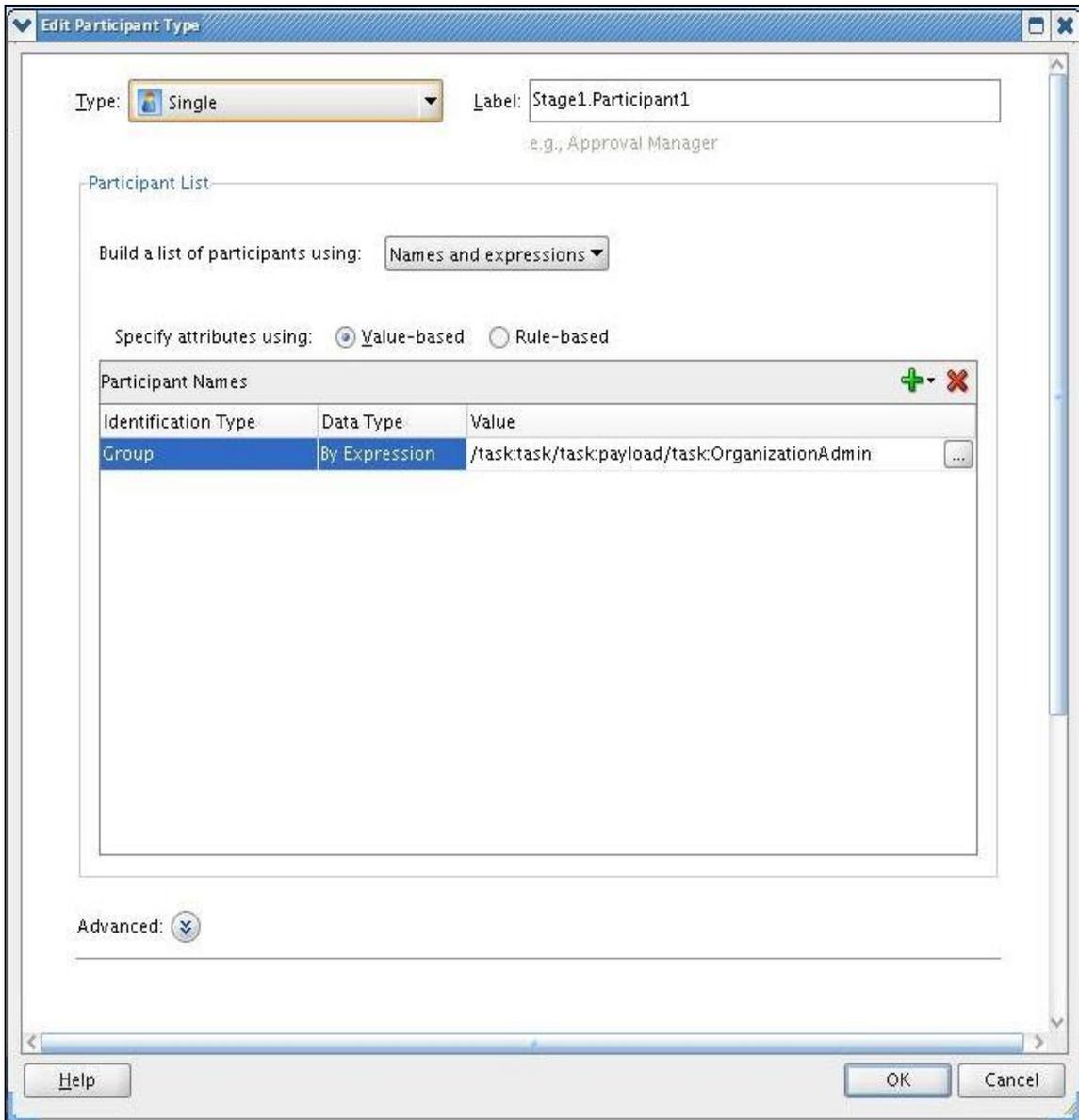


Figure 8: Edit Participant Type dialog

21. Save your work.

Next section illustrates how to deploy this composite to Oracle SOA Managed server.

6.3. Deploying SOA composite to Oracle SOA Managed Server

Before deploying the SOA composite, we need to set the BpelcClasspath property in the System MBean Browser of Oracle Enterprise Manager Fusion Middleware Control (EM) Console

1. Go to EM console. Login as weblogic user.
2. Expand Weblogic Domain from the left pane. Right click on <WLS_DOMAIN> → System MBeans Browser
3. Go to Application Defined MBeans → oracle.as.soainfra.config → Server:<SOA_SERVER> → BPELConfig → bpel
4. Under Attributes column click on BpelcClasspath. Provide full path for oimclient.jar and jps-manifest.jar. These files are located at following location in a shiphome

<OIMHOME>/server/client/oimclient.jar

<BEAHOME>/oracle_common/modules/oracle.jps_11.1.1.1/jps-manifest.jar

Note: Replace OIMHOME and BEAHOME with corresponding full paths. The paths should be separated by a colon (:) on UNIX and semicolon (;) on Windows

Now follow these steps to deploy SOA composite to Oracle SOA server.

1. Right click on SelfRegistrationApproval project and click Deploy → SelfRegistrationApproval. A popup wizard will appear which will guide you through the deployment of composite to Oracle SOA server.
2. In Deployment Action step, select Deploy to Application Server. Click Next
3. In Deployment Configuration step, check “Overwrite any existing composite with same revision ID” option (see Figure 9). Click Next.

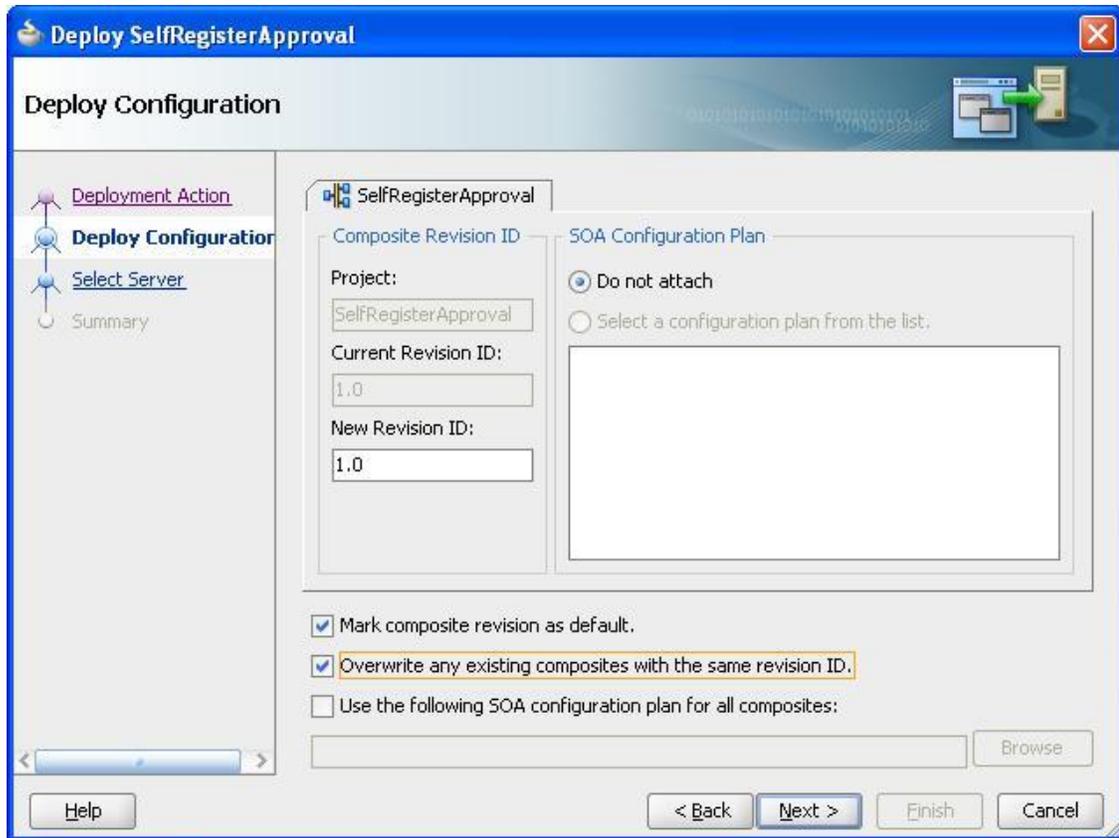


Figure 9: Deploy SelfRegistrationApproval wizard

5. In Select Server step, select the connection to your application server. If connection does not exist, create a new connection by clicking on + sign, "Create Application Server Connection" popup will come up. If you have already created a connection to Application Server, then go to Step 10, else proceed with Step 6.
6. In "Create Application Server Connection" popup, provide a name for connection and click Next.
7. Provide password for weblogic user in Authentication step. Click Next.
8. Give hostname and port for Admin server and also enter weblogic domain in Configuration step (see Figure 10). Click Next.

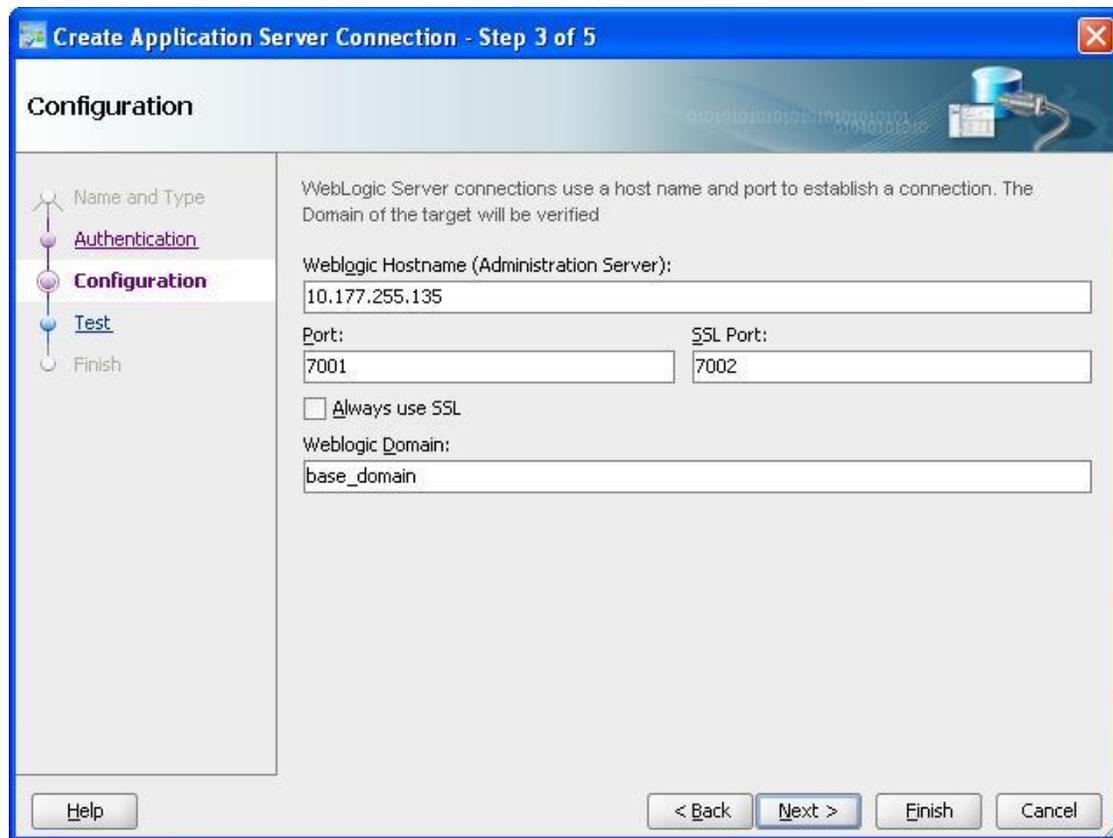


Figure 10: Create Application Server Connection

9. Click on Test connection. If all tests are successful, click Finish.
10. Click Next.
11. In Select SOA server ensure SOA server is selected. Click Finish.
12. Check compiler and deployment logs in JDeveloper for any errors.

Next section illustrates how to register this composite with OIM Managed server.

6.4. Registering SOA composite with OIM Managed Server

The SOA composite must be registered with Oracle Identity Manager before it can be used as an approval process. To register a SOA composite with Oracle Identity Manager:

1. Set the environment if not already set (for Linux machines)
 - `cd <BEAHOME>/wlserver_10.3/server/bin`
 - `bash`
 - `source setWLSEnv.sh`
2. Create a `SelfRegistrationApproval.props` property file at directory location `OIM_HOME/server/workflows/registration/` with the following content:

```
name=SelfRegistrationApproval
category=Approval
providerType=BPEL
serviceName=RequestApprovalService
domainName=default
version=1.0
payloadID=payload
operationID=process
listOfTasks=ApprovalTask
```

Note: Do not give any extra space in the properties file.

3. Run the following command from `OIM_HOME/server/workflows/new-workflow/` directory:
 - `ant -f registerworkflows-mp.xml register`
4. Enter Oracle Identity Manager administrator username when the following prompt is displayed.
Enter the username
5. Enter Oracle Identity Manager administrator password when following prompt is displayed.

Enter the password

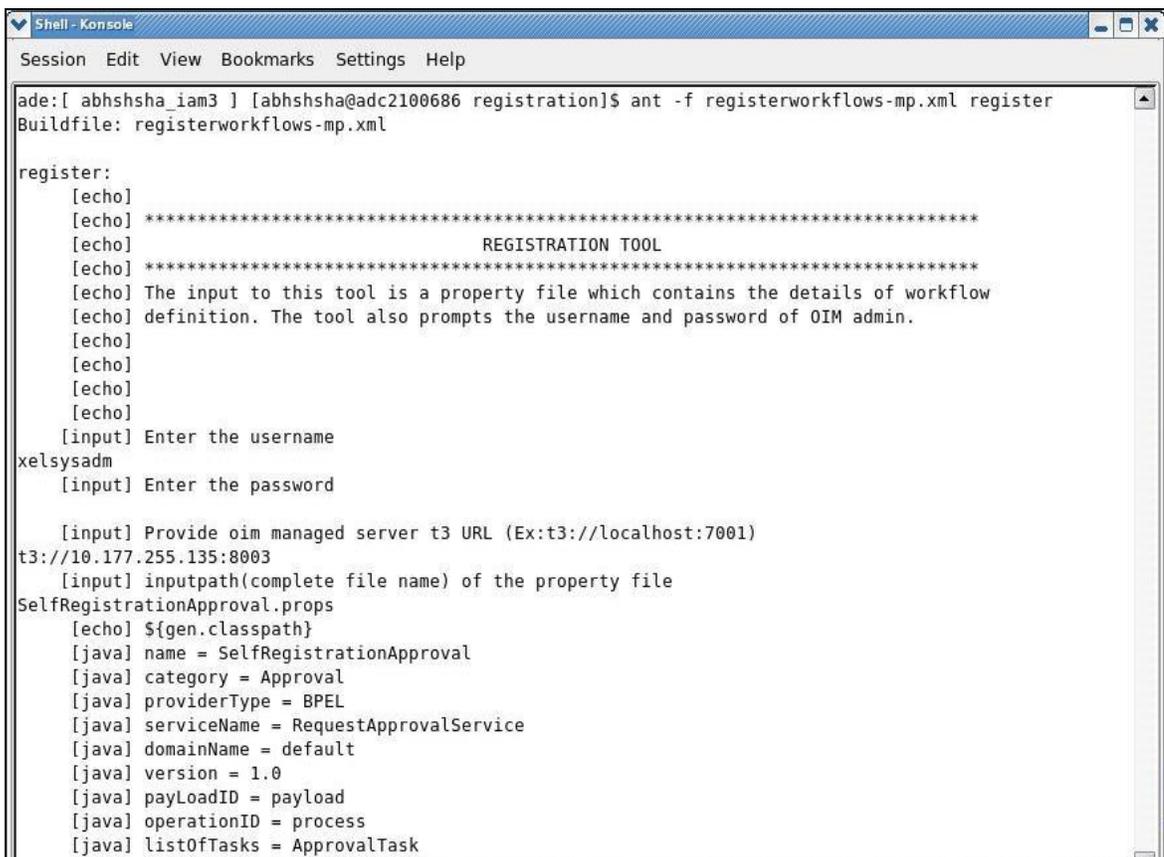
6. Enter OIM Managed server t3 URL (Example t3://10.177.255.135:8003) when following prompt is displayed.

Provide oim server t3 URL

7. Enter the path (absolute or relative) of the property file name that you created in step 1 when following prompt is displayed.

Input path (complete file name) of the property file

Following screenshot (Figure 11) shows registration of SelfRegistrationApproval SOA composite.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
ade:[ abhshsha_iam3 ] [abhshsha@adc2100686 registration]$ ant -f registerworkflows-mp.xml register
Buildfile: registerworkflows-mp.xml

register:
[echo]
[echo] *****
[echo]                                REGISTRATION TOOL
[echo] *****
[echo] The input to this tool is a property file which contains the details of workflow
[echo] definition. The tool also prompts the username and password of OIM admin.
[echo]
[echo]
[echo]
[input] Enter the username
xelsysadm
[input] Enter the password

[input] Provide oim managed server t3 URL (Ex:t3://localhost:7001)
t3://10.177.255.135:8003
[input] inputpath(complete file name) of the property file
SelfRegistrationApproval.props
[echo] ${gen.classpath}
[java] name = SelfRegistrationApproval
[java] category = Approval
[java] providerType = BPEL
[java] serviceName = RequestApprovalService
[java] domainName = default
[java] version = 1.0
[java] payloadID = payload
[java] operationID = process
[java] listOfTasks = ApprovalTask
```

Figure 11: Register Custom SOA Composite

Once the registration is successful, proceed to the next section which illustrates creating approval policy to invoke this composite.

6.5. Creating Approval Policies in OIM

We will need two approval policies. One approval policy to assign Self-Register User tasks at request level to SYSTEM ADMINISTRATORS role and another approval policy to assign tasks at operation level to Organization Administrators.

Following are the steps to create approval policy for request level approval.

1. Login to OIM as Shirley Schmidt.
2. Go to Advanced Admin console
3. Click on Policies tab
4. Click on Create button in left pane.
5. Provide policy name (say SelfRegisterPolicyRL)
6. Select Request type as Self-Register User.
7. Choose Request Level of approval from the drop down
8. Select “default/DefaultRoleApproval!1.0” as Approval Process. Click Next
9. Provide a Rule name (say SelfRegistrationRuleRL)
10. Click on Add Simple Rule. A popup will appear (see Figure 12)
11. Provide following values for the rule (see Figure 12)
 - Entity → Request
 - Attribute → Request Type
 - Condition → Equals
 - Value → Self-Register User
 - Parent Rule Container → Approval Rule

Add Simple Rule

* Indicates required fields.

* Entity: Request

* Attribute: Request Type

* Condition: Equals

* Value: Self-Register User

* Parent Rule Container: Approval Rule

Save Cancel

Figure 12: Add Simple Rule popup

12. Click on Save
13. Click Next on main page
14. Click Finish. Success message of creation of approval policy should be displayed.

Following are the steps to create approval policy for request level approval.

1. Click on Create button in left pane.
2. Provide policy name (say SelfRegisterPolicyOL)
3. Select Request type as Self-Register User.
4. Choose Operation Level of approval from the drop down
5. Check All Scope.
6. Select the newly created composite (default/SelfRegistrationApproval!1.0) as Approval Process. Click Next
7. Provide a Rule name (say SelfRegistrationRuleOL)
8. Click on Add Simple Rule. A popup will appear (see Figure 12)
9. Provide following values for the rule (see Figure 12)
 - Entity → Request
 - Attribute → Request Type

- Condition → Equals
- Value → Self-Register User
- Parent Rule Container → Approval Rule

10. Click on Save

11. Click Next on main page

12. Click Finish. Success message of creation of approval policy should be displayed.

13. Logout of OIM.

The next section will discuss how to test custom approval process.

6.6. Testing of custom approval process

Before invoking the SOA composite, we need to provide a valid OIM url for the property set in composite.xml of SOA composite. We can set this value from EM console. Follow the steps below to achieve this.

1. Go to EM console. Login as weblogic user.
2. Expand Weblogic Domain from the left pane. Right click on <WLS_DOMAIN> → System MBeans Browser
3. Go to Application Defined MBeans → oracle.soa.config → Server:<SOA_SERVER> → SCAComposite → SelfRegistrationApproval[1.0] → SCAComposite.SCAComponent → ApprovalProcess.
4. Under Attributes column click on Properties → Element_0. Provide valid OIM url for the “value” key (e.g. t3://10.177.255.135).

Now follow these steps for testing the approval process:

1. Go to the login page of OIM.
2. Click on “Register” link
3. Provide user details in Basic Information step. Click Next
4. Fill challenge questions. Click Register
5. Make a note of registration tracking request number. Click on “Back to login.”
6. Login as Shirley Schmidt.
7. Click on Search Approval Tasks. Make sure you are able to see Self-Registration task with the same request ID.

8. Select the task row with the request number matching registration tracking request number and click Open task details.
9. Provide Organization as “FINANCE”
10. Click Approve. The task will move to Operation level of approval
11. Logout of OIM.
12. Login as Danny Crane (This user is a member of FINANCE_APPROVERS group. So he should be able to see the task).
13. Click on Search Approval Tasks. Make sure you are able to see Self-Registration task with the same request ID.
14. Click on Approve. The task should have been approved successfully and the user created in OIM.
15. Logout of OIM.
16. Click on Track Registration. Provide tracking id and click Submit. Verify that registration request status is completed.

Appendix A

Java Code

```
try {

    System.out.println("Prototype for invoking an OIM API from a SOA Composite");
    System.out.println("RTM Usecase: Self Registration Approval by Organization
                        Administrator");

    String oimUserName = "";
    String oimPassword = "";
    String oimURL = "";

    //get system administrator's credentials
    oracle.security.jps.JpsContext ctx =
        oracle.security.jps.JpsContextFactory.getContextFactory().getContext();

    final oracle.security.jps.service.credstore.CredentialStore cs =
        (oracle.security.jps.service.credstore.CredentialStore)ctx.getServiceInstance(
            oracle.security.jps.service.credstore.CredentialStore.class);

    oracle.security.jps.service.credstore.CredentialMap cmap =
        cs.getCredentialMap("oracle.oim.sysadminMap");

    oracle.security.jps.service.credstore.Credential cred = cmap.getCredential("sysadmin");

    if (cred instanceof oracle.security.jps.service.credstore.PasswordCredential) {

        oracle.security.jps.service.credstore.PasswordCredential pcred =
            (oracle.security.jps.service.credstore.PasswordCredential)cred;

        char[] p = pcred.getPassword();
        oimUserName = pcred.getName();
        oimPassword = new String(p);
    }

    //get oimurl
    Object obj = getVariableData("oimurl");
    oimURL = obj.toString();
    System.out.println("oimurl=" + oimURL);

    // set the initial context factory
```

```

String oimInitialContextFactory = "weblogic.jndi.WLInitialContextFactory";

// set up the environment for making the OIM API invocation
java.util.Hashtable env = new java.util.Hashtable();
env.put(oracle.iam.platform.OIMClient.JAVA_NAMING_FACTORY_INITIAL,
        oimInitialContextFactory);
env.put(oracle.iam.platform.OIMClient.JAVA_NAMING_PROVIDER_URL, oimURL);

oracle.iam.platform.OIMClient client = new oracle.iam.platform.OIMClient(env);
client.login(oimUserName, oimPassword.toCharArray());
System.out.println("Login Successful");

// get the RequestService to get details of the request
oracle.iam.request.api.RequestService reqSvc =
        (oracle.iam.request.api.RequestService)client.getService(
                oracle.iam.request.api.RequestService.class);

Object reqIdXMLElement = getVariableData("inputVariable",
        "payload",
        "/ns3:process/ns4:RequestID");

String reqId = ((oracle.xml.parser.v2.XMLElement)reqIdXMLElement).getText();
System.out.println("The request ID is "+reqId);

// invoke the getBasicRequestData() method on the RequestService API
oracle.iam.request.vo.Request req = reqSvc.getBasicRequestData(reqId);
String act_key = "";
java.util.List< oracle.iam.request.vo.RequestEntity> targetEntities =
        req.getTargetEntities();

for( oracle.iam.request.vo.RequestEntity reqEntity: targetEntities){
    java.util.List< oracle.iam.request.vo.RequestEntityAttribute> attributes =
        reqEntity.getEntityData();
    for( oracle.iam.request.vo.RequestEntityAttribute attribute : attributes){
        if(attribute.getName().equalsIgnoreCase("Organization")){
            act_key = attribute.getValue().toString();
        }
    }
}

System.out.println("Organization Key is "+act_key);

```

```

if(act_key != "" && act_key != " ") {
    Thor.API.Operations.tcOrganizationOperationsIntf orgAPI =
        (Thor.API.Operations.tcOrganizationOperationsIntf)client.getService(
            Thor.API.Operations.tcOrganizationOperationsIntf.class);

    Thor.API.tcResultSet rset= orgAPI.getAdministrators(Long.parseLong(act_key));

    StringBuffer sb = new StringBuffer();

    for (int i = 0; i < rset.getRowCount();i++){
        rset.goToRow(i);
        sb.append(rset.getStringValue("Groups.Group Name"));
        if(i >= 0 && i < (rset.getRowCount()-1)){
            sb.append(",");
        }
    }
    String grpNames = sb.toString();
    System.out.println("Groups="+grpNames);
    setVariableData("orgadmin",grpNames);
}
else{
    setVariableData("orgadmin","SYSTEM ADMINISTRATORS");
}
Object obj1 = getVariableData("orgadmin");
System.out.println("OrganizationAdmins = " + obj1.toString());
} catch (Exception e){
    System.out.println("-----");
    e.printStackTrace();
    System.out.println("-----");
}
}

```

Block level description

In order to get the organization administrator sing OIM API's we perform the following steps in Java code:

1. Get credentials for system administrator
2. Login as system administrator
3. Get Organization Administrator using OIM APIs

Each of these steps have been explained below

1. Get credentials for system administrator

Credentials of system administrator are stored in a credential store (cwallet). First we get the Credential Store, then the credential map and then the credential using the key. Following code snippet explains this

```
//get Credential store
oracle.security.jps.JpsContext ctx =
    oracle.security.jps.JpsContextFactory.getContextFactory().getContext();

final oracle.security.jps.service.credstore.CredentialStore cs =
    (oracle.security.jps.service.credstore.CredentialStore)ctx.getServiceInstance(
oracle.security.jps.service.credstore.CredentialStore.class);

//get Credential
oracle.security.jps.service.credstore.CredentialMap cmap =
    cs.getCredentialMap("oracle.oim.sysadminMap");
oracle.security.jps.service.credstore.Credential cred = cmap.getCredential("sysadmin");
```

2. Login as system administrator

We first setup the environment and then login to OIM as System Administrator. Following code snippet explains this

```
//setup the environment
String oimInitialContextFactory = "weblogic.jndi.WLInitialContextFactory";
java.util.Hashtable env = new java.util.Hashtable();
env.put(oracle.iam.platform.OIMClient.JAVA_NAMING_FACTORY_INITIAL,
    oimInitialContextFactory);
env.put(oracle.iam.platform.OIMClient.JAVA_NAMING_PROVIDER_URL, oimURL);

//login to OIM
oracle.iam.platform.OIMClient client = new oracle.iam.platform.OIMClient(env);
client.login(oimUserName, oimPassword.toCharArray());
    System.out.println("Login Successful");
```

3. Get Organization Administrator using OIM APIs

oracle.iam.request.api.RequestService api has been used to get the request object. Thor.API.Operations.tcOrganizationOperationsIntf api has been used to get the Organization administrators

Appendix B

Store Credentials in CSF

Follow the steps mentioned below to store credential in CSF:

1. Go to EM console. Login as weblogic.
2. Expand Weblogic Domain in left pane.
3. Right click on <WLS_DOMAIN>. Select Security → Credentials.
4. Click on “Create Map” button. Provide name for map as “oracle.oim.sysadminMap”. Click OK.
5. Click on “Create Key” button (See Figure 13). Provide following details:
 - Select Map: oracle.oim.sysadminMap
 - Key: sysadmin
 - Type: Password
 - Username: <Shirley Schmidt’s login ID>
 - Password: < Shirley Schmidt’s password>

Click OK.

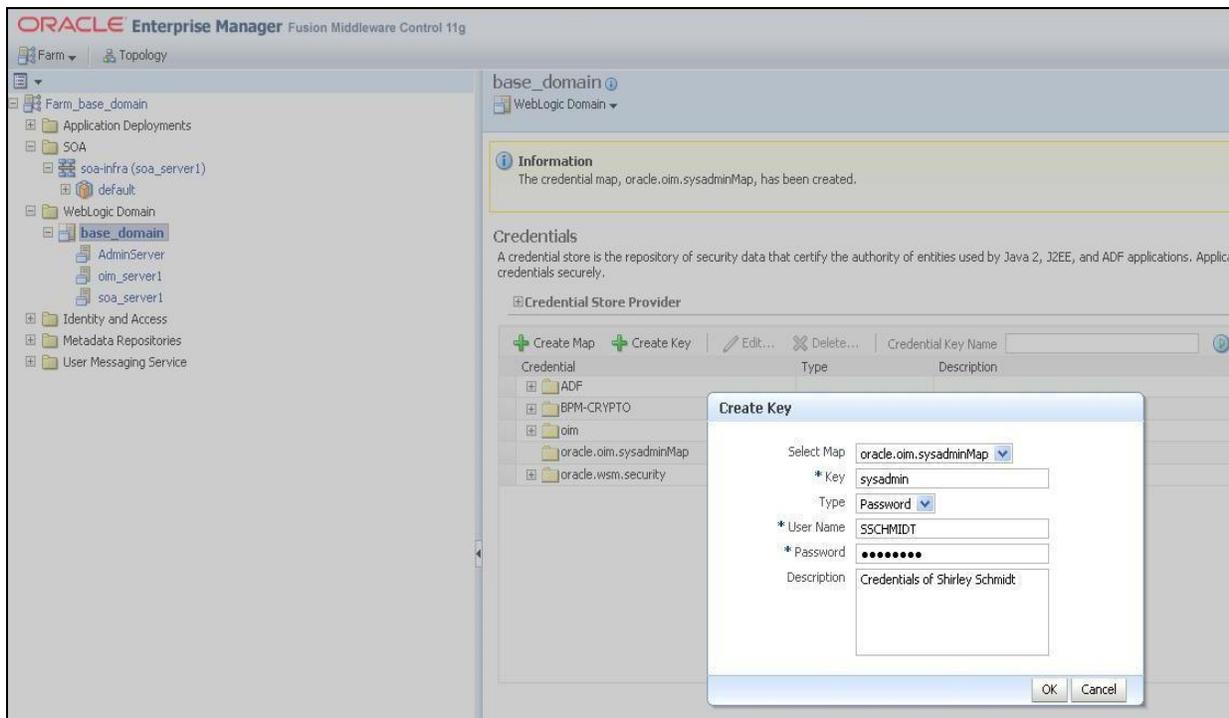


Figure 13: Credentials for OIM System Administrator

Appendix C

Cleanup

Delete the following approval policies (if present) from Advanced Admin console

1. SelfRegisterPolicyRL
2. SelfRegisterPolicyOL

If you had registered the approval process (Section 6.4), you have to disable it. To disable approval processes in OIM, follow these steps:

1. Set the environment if not already set (for Linux machines)
 - `cd <BEAHOME>/wlserver_10.3/server/bin`
 - `bash`
 - `source setWLSEnv.sh`
2. Run the following command from `<OIMHOME>/server/workflows/registration/` directory:
 - `ant -f registerworkflows-mp.xml disable`
3. Enter Oracle Identity Manager administrator username when the following prompt is displayed.
Enter the username
4. Enter Oracle Identity Manager administrator password when following prompt is displayed.
Enter the password
5. Enter OIM Managed server t3 URL (Example `t3://10.177.255.135:8003`) when following prompt is displayed.
Provide oim managed server t3 URL
6. Enter the domain of the workflow when following
Input the domain of the workflow
7. Enter the name of the workflow (SelfRegistrationApproval) when following
Input the name of the workflow
8. Enter the version of the workflow (1.0) when following
Input the version of the workflow

Following screenshot (Figure 17) shows disabling of SelfRegistrationApproval SOA composite.

```

Shell - Konsole
Session Edit View Bookmarks Settings Help
bash-3.2$ ant -f registerworkflows-mp.xml disable
Buildfile: registerworkflows-mp.xml

disable:
  [echo]
  [echo] *****
  [echo]                               DISABLE-COMPOSITES TOOL
  [echo] *****
  [echo] The input to this tool is a name of workflow definition.The tool also prompts
  [echo] the username and password of OIM admin.
  [echo]
  [input] Enter the username
sschmidt
  [input] Enter the password

  [input] Provide oim managed server t3 URL (Ex:t3://localhost:7001)
t3://10.177.255.135:8003
  [input] Input the domain of the workflow
default
  [input] Input the name of the workflow
SelfRegistrationApproval
  [input] Input the version of the workflow
1.0

```

Figure 14: Disable SelfRegistrationApproval

To undeploy the composite from Oracle SOA server, follow these steps:

1. Set the environment if not already set (for Linux machines)
 - `cd <BEAHOME>/wlserver_10.3/server/bin`
 - `bash`
 - `source setWLSEnv.sh`
2. Run the following command from <SOAHOME>/bin/ directory:
 - `ant -f ant-sca-deploy.xml undeploy -DserverURL=server.url -DcompositeName=compoite.name -Drevision=revision.id -Duser=user -Dpassword=password -Dpartition=partition.name`

where

- serverURL - the url of the server that hosts soa-infra app, (e.g. `http://10.177.255.135:8001`)
- compositeName - the name of the composite (SelfRegistrationApproval).
- revision - the revision ID of the composite (1.0).
- user/password – weblogic user and password
- partition - optional, the name of the partition the composite is located. default value is "default"

Re-run of sample

Make sure that you have performed the cleanup as described above. After that follow the steps as described in chapter 6.

Note: If you had already registered the approval process in previous run of this sample, you may encounter following exception while trying to execute the steps mentioned in section 6.4.

oracle.iam.platform.workflowservice.exception.IAMWorkflowException: The workflow definition with name default/SelfRegistrationApproval!1.0 already exists in OIM.

In such case, perform the following alternative steps to enable the approval process:

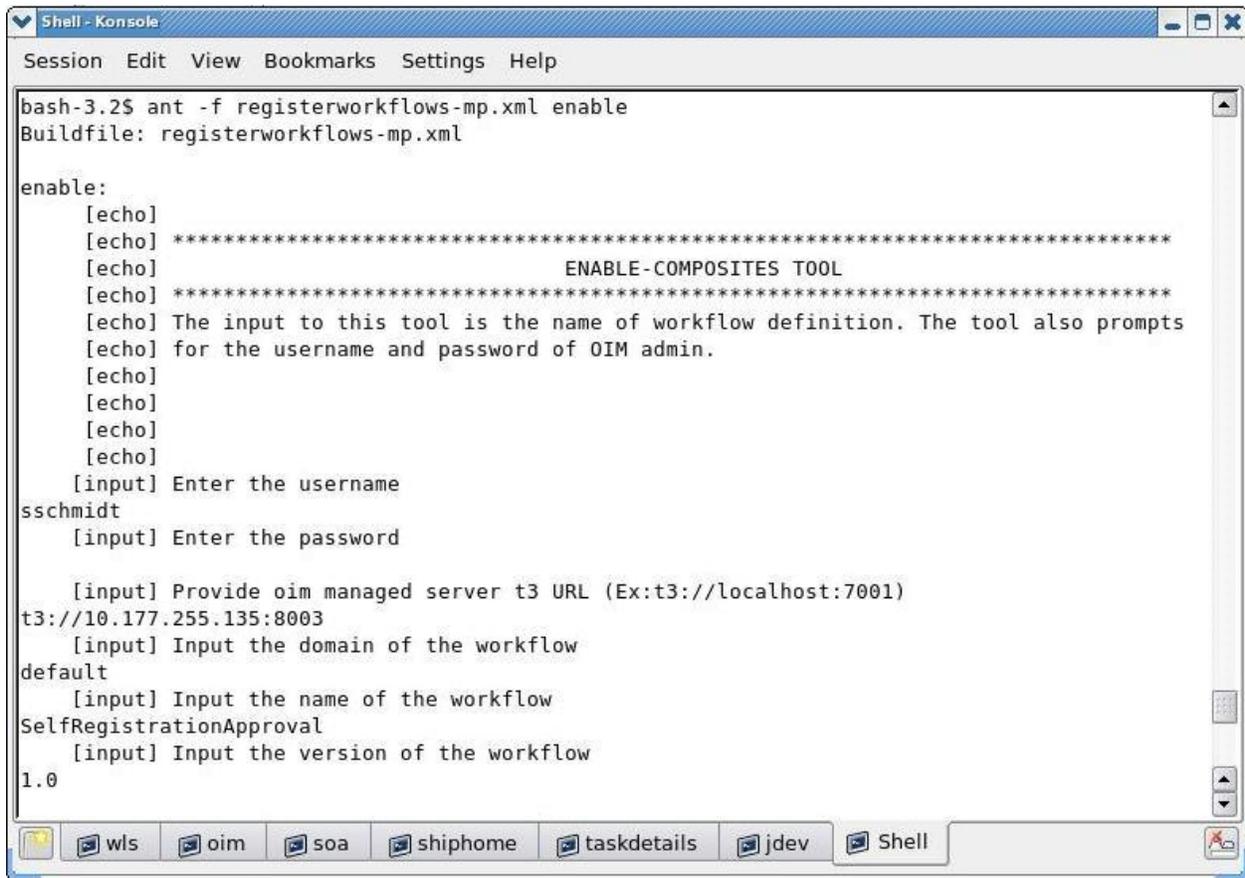
1. Set the environment if not already set (for Linux machines)
 - `cd <BEAHOME>/wlserver_10.3/server/bin`
 - `bash`
 - `source setWLSEnv.sh`
2. Run the following command from `<OIMHOME>/server/workflows/registration/` directory:
 - `ant -f registerworkflows-mp.xml enable`
3. Enter Oracle Identity Manager administrator username when the following prompt is displayed.

Enter the username
4. Enter Oracle Identity Manager administrator password when following prompt is displayed.

Enter the password
5. Enter OIM Managed server t3 URL (Example `t3://10.177.255.135:8003`) when following prompt is displayed.

Provide oim managed server t3 URL
6. Enter the domain of the workflow when following
Input the domain of the workflow
7. Enter the name of the workflow (SelfRegistrationApproval) when following
Input the name of the workflow
8. Enter the version of the workflow (1.0) when following
Input the version of the workflow

Following screenshot (Figure 18) shows enabling of SelfRegistrationApproval SOA composite.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
bash-3.2$ ant -f registerworkflows-mp.xml enable
Buildfile: registerworkflows-mp.xml
enable:
  [echo]
  [echo] *****
  [echo]                               ENABLE-COMPOSITES TOOL
  [echo] *****
  [echo] The input to this tool is the name of workflow definition. The tool also prompts
  [echo] for the username and password of OIM admin.
  [echo]
  [echo]
  [echo]
  [input] Enter the username
sschmidt
  [input] Enter the password

  [input] Provide oim managed server t3 URL (Ex:t3://localhost:7001)
t3://10.177.255.135:8003
  [input] Input the domain of the workflow
default
  [input] Input the name of the workflow
SelfRegistrationApproval
  [input] Input the version of the workflow
1.0
```

Figure 18: Enable SelfRegistrationApproval