Oracle® Banking Platform

Administrator's Guide

Release 2.3.0.0.0

E56276-01

July 2014

ORACLE®

Oracle Banking Platform Administrator's Guide, Release 2.3.0.0.0

E56276-01

Contents

8 Configuration Export-Import Operations

List of Figures

List of Tables

Preface

This guide describes how to administer the Oracle Banking Platform Applications environment, including user administration, batch execution, DPA approvals, application monitoring, and bank, branch and channels set up.

Oracle recommends that you review its contents before installing, or working with the product.

This preface contains the following topics:

Audience

This guide is intended for the administrators of Oracle Banking Platform.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Organization of the Guide

This document contains:

Chapter 1, "Users Administration"

This chapter describes all user management related activities to be performed by an administrator for Oracle Banking Platform.

Chapter 2, "Approvals Management"

This chapter describes Discretionary Pricing Assessment (DPA) approvals, manual credit decision approvals and worklist authorization related activities to be performed as an administrator.

Chapter 3, "Data Management"

This chapter describes data related activities to be performed as an administrator.

Chapter 4, "Setting Up The Bank And Branch"

This chapter provides the process of setting up the bank and the branch commonly referred to as the Day 0 setups. It gives the details of the Administration User Interface (UI) offered by Oracle Banking Channels Bank User Experience, using which, the administrator can perform a bank and a branch setup.

Chapter 5, "Setting Up Channels"

This chapter describes the process of setting up the channels.

Chapter 6, "Application Monitoring Using OBP Application"

This chapter provides an overview on the various monitoring operations performed as an administrator using Oracle Banking Platform screens.

Chapter 7, "Application Monitoring Using OBP EM Plugin"

This chapter provides an overview on the various monitoring operations performed as an administrator, using Oracle Banking Platform - Enterprise Manger (EM) Plugin.

Chapter 8, "Configuration Export-Import Operations"

This document gives an insight to the Configuration Export-Import operations.

Related Documents

For more information, see the following documentation:

- For installation and configuration information, see the Oracle Banking Installation Guide - Silent Installation
- For a comprehensive overview of security for Oracle Banking, see the Oracle Banking Security Guide
- For the complete list of Oracle Banking licensed products and the Third Party licenses included with the license, see the Oracle Banking Licensing Guide
- For information related to customization and extension of Oracle Banking, see the Oracle Banking Extensibility Guide
- For information on the functionality and features of the Oracle Banking product licenses, see the respective Oracle Banking Functional Overview documents

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Users Administration

This chapter describes all user management related activities to be performed by an administrator for Oracle Banking Platform.

This chapter includes the following topics:

- Creating Users in Oracle Identity Manager (OIM)
- Assigning Roles to Users in OIM
- Locking Users in OIM
- Unlocking Users in OIM
- User Management Using the OBP Admin Application
- Unlocking Users in Oracle WebLogic Server (OWS) Administration Console

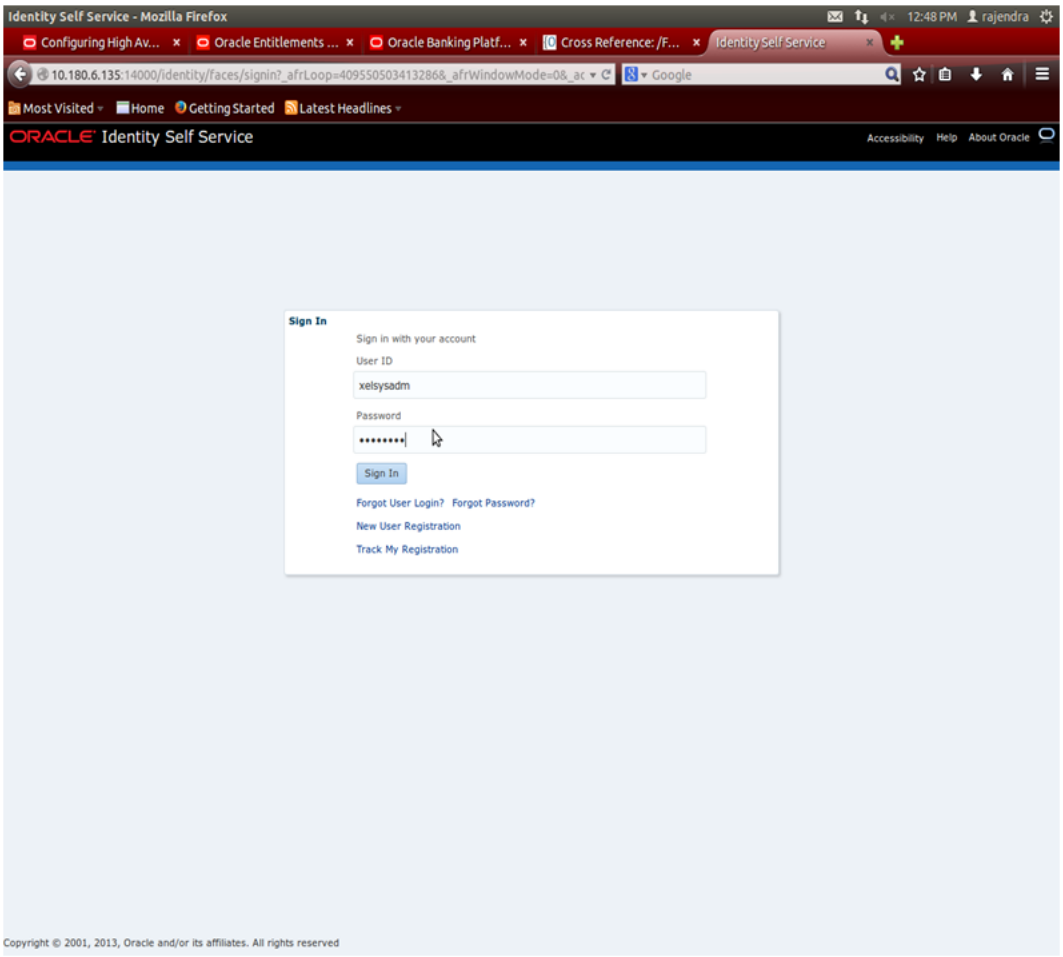
1.1 Creating Users in Oracle Identity Manager (OIM)

This section explains the procedure to create users in Oracle Identity Manager (OIM).

To create users in OIM:

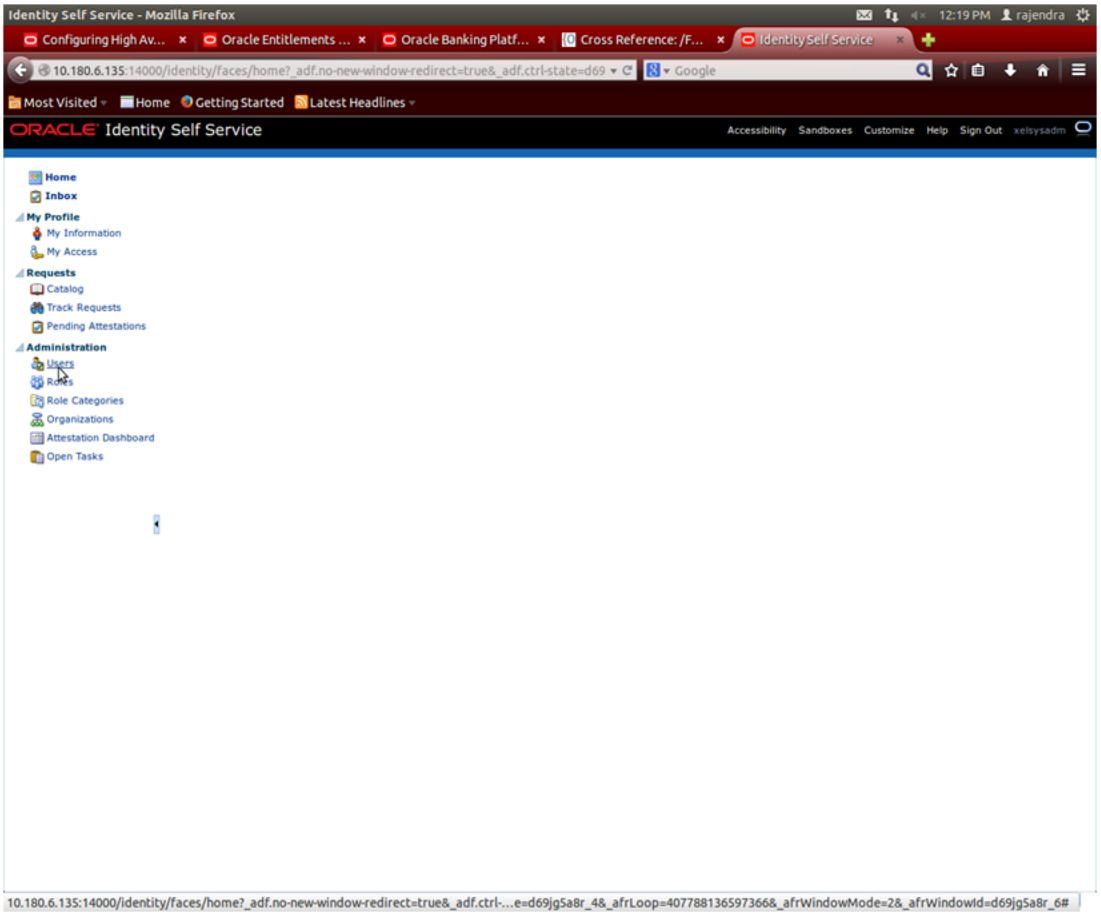
1. Log in to OIM with the User ID as **xelsysadm** and the relevant <Password>.

Figure 1–1 OIM Log in



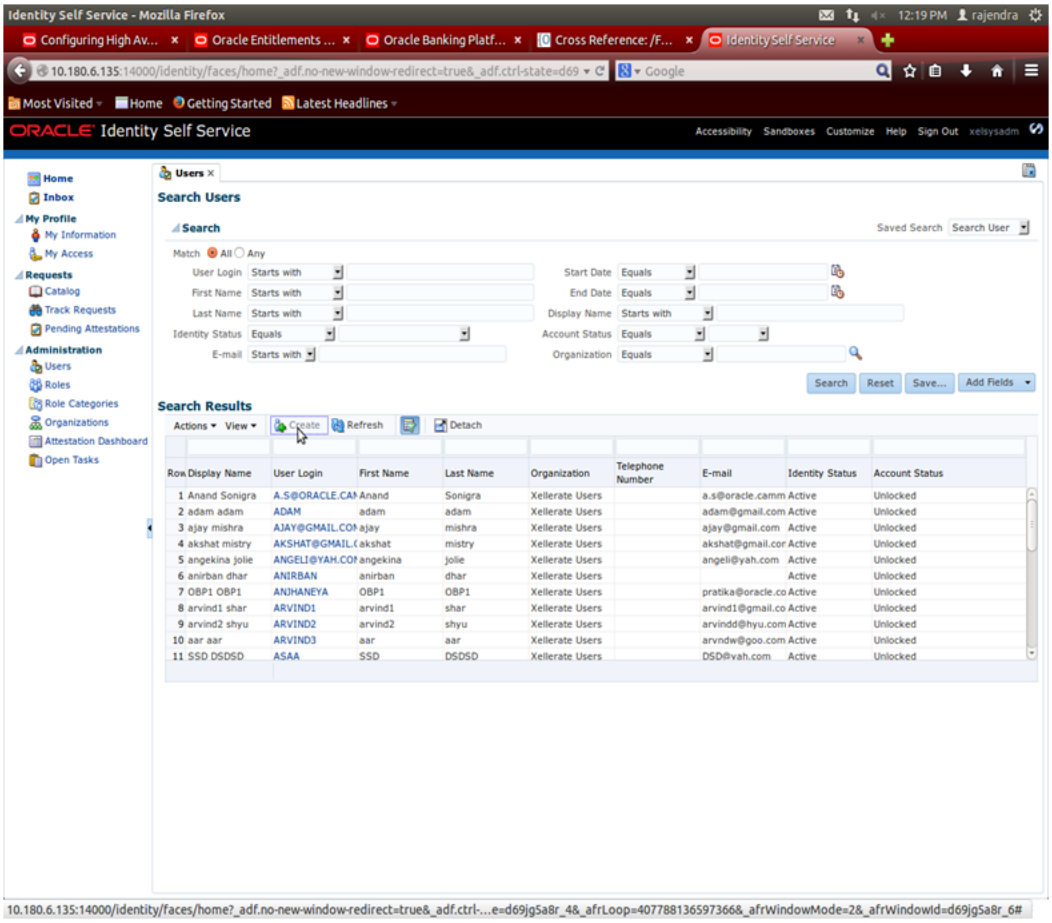
2. Click **Users** under the Administration section.

Figure 1–2 Administration - Click Users



3. In the **Search Users** page, search for existing users. The Search Results appear.
4. Click **Create** in the Search Results section to create a new user.

Figure 1–3 Search Users - Click Create



5. In the **Create User** page, enter the required user details.

Figure 1–4 Enter User Details

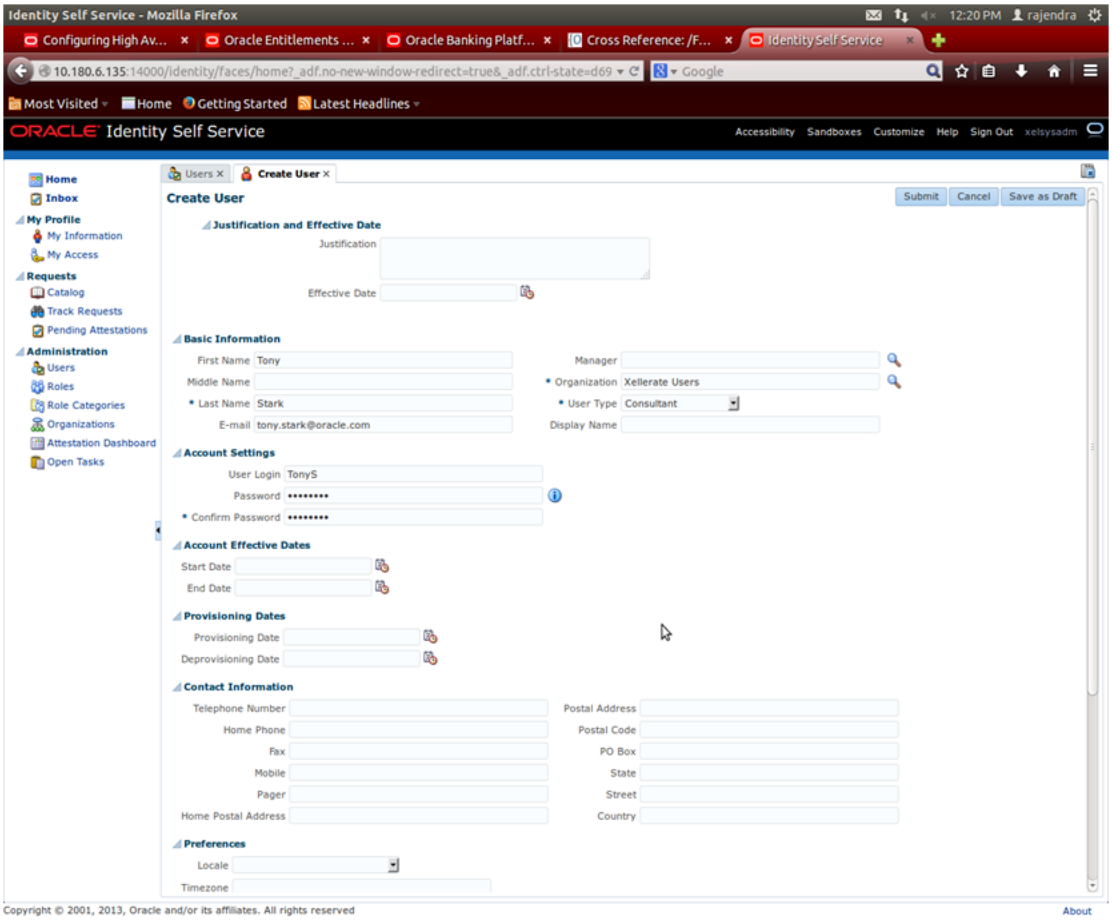
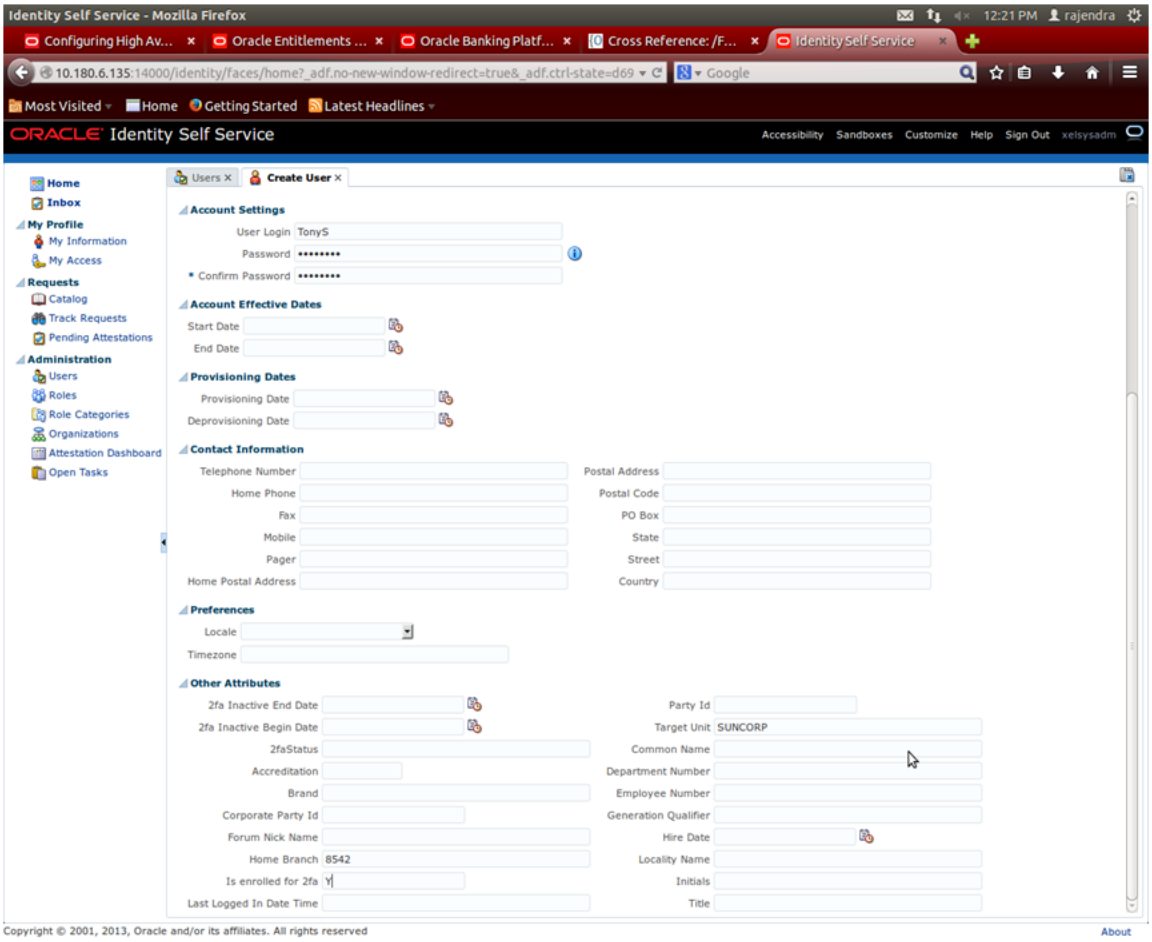


Figure 1–5 Enter User Details (Continued)



6. Click **Submit.**

On completion of this procedure the user gets created in OIM.

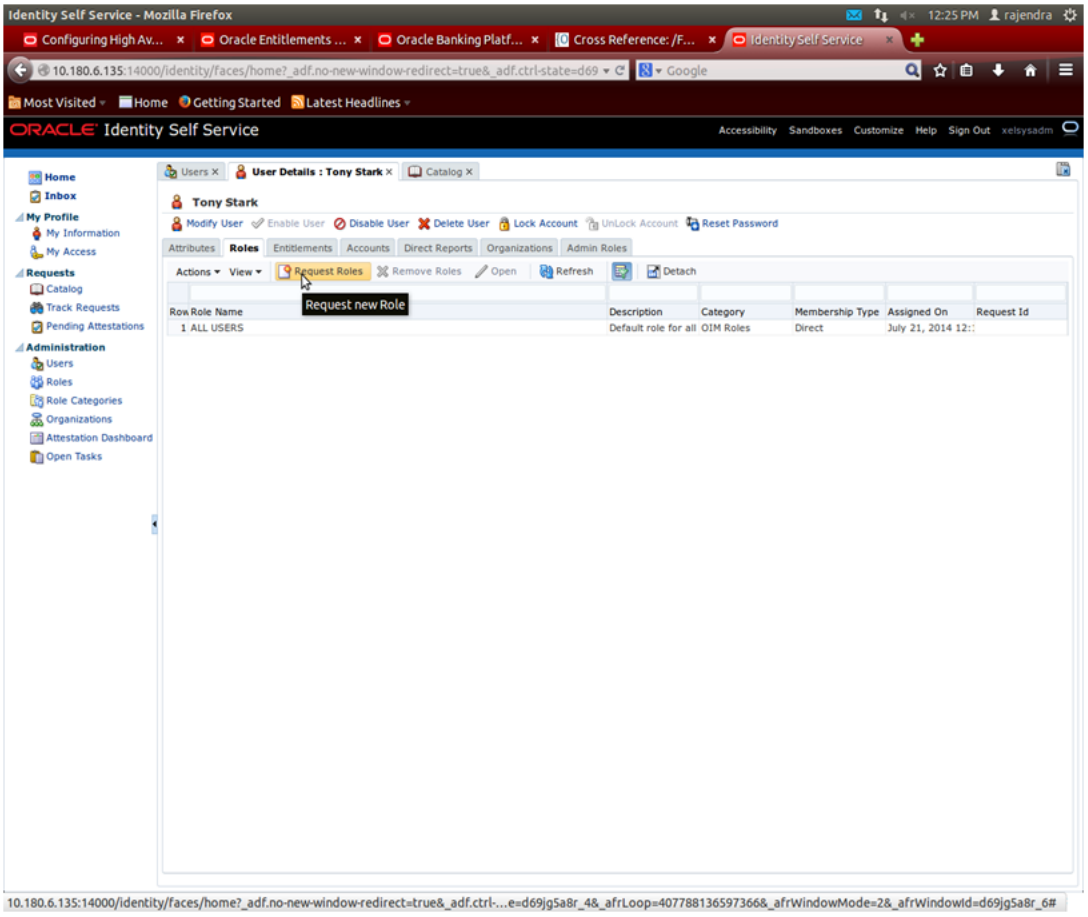
1.2 Assigning Roles to Users in OIM

This section explains how to assign roles to the user in OIM.

To assign a role to a user:

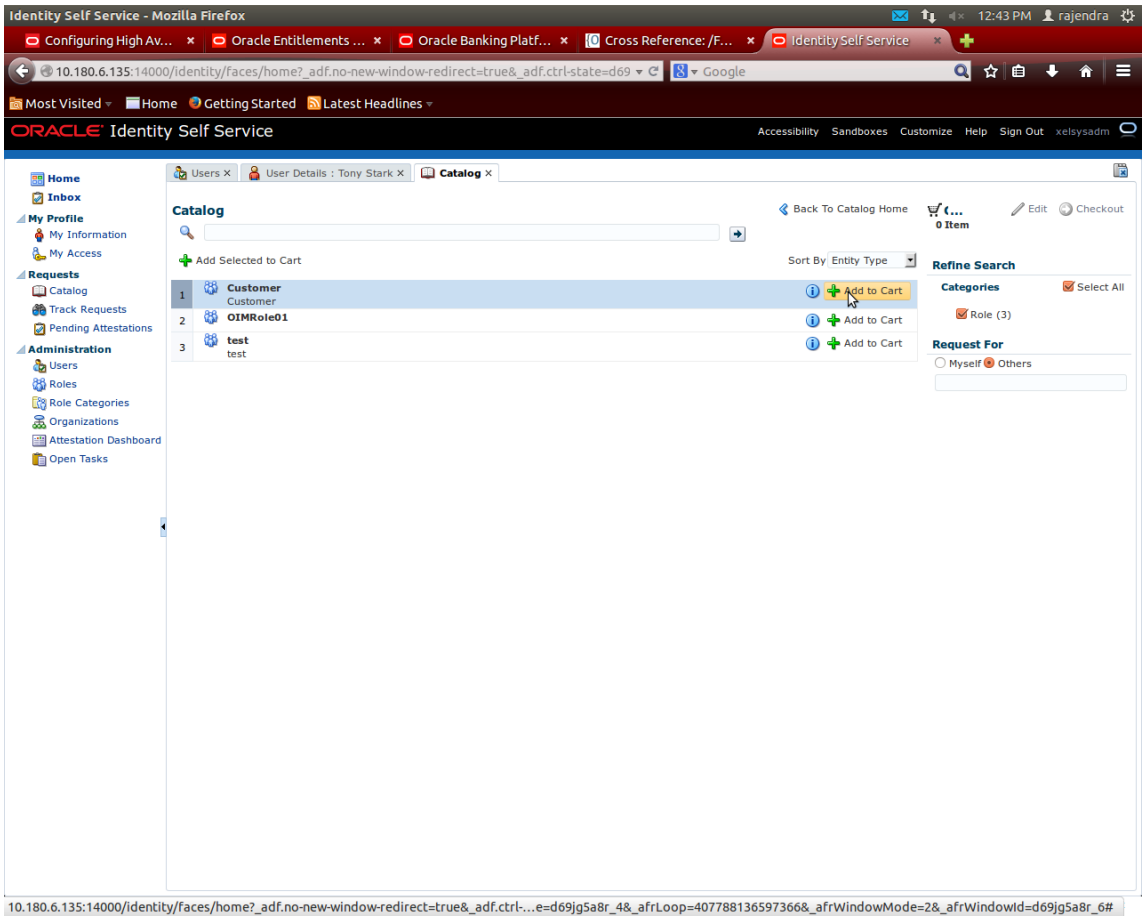
1. Log in to OIM.
2. Navigate to the **Roles Tab** under the User.
3. Click **Request Roles.**

Figure 1–6 Requesting Roles



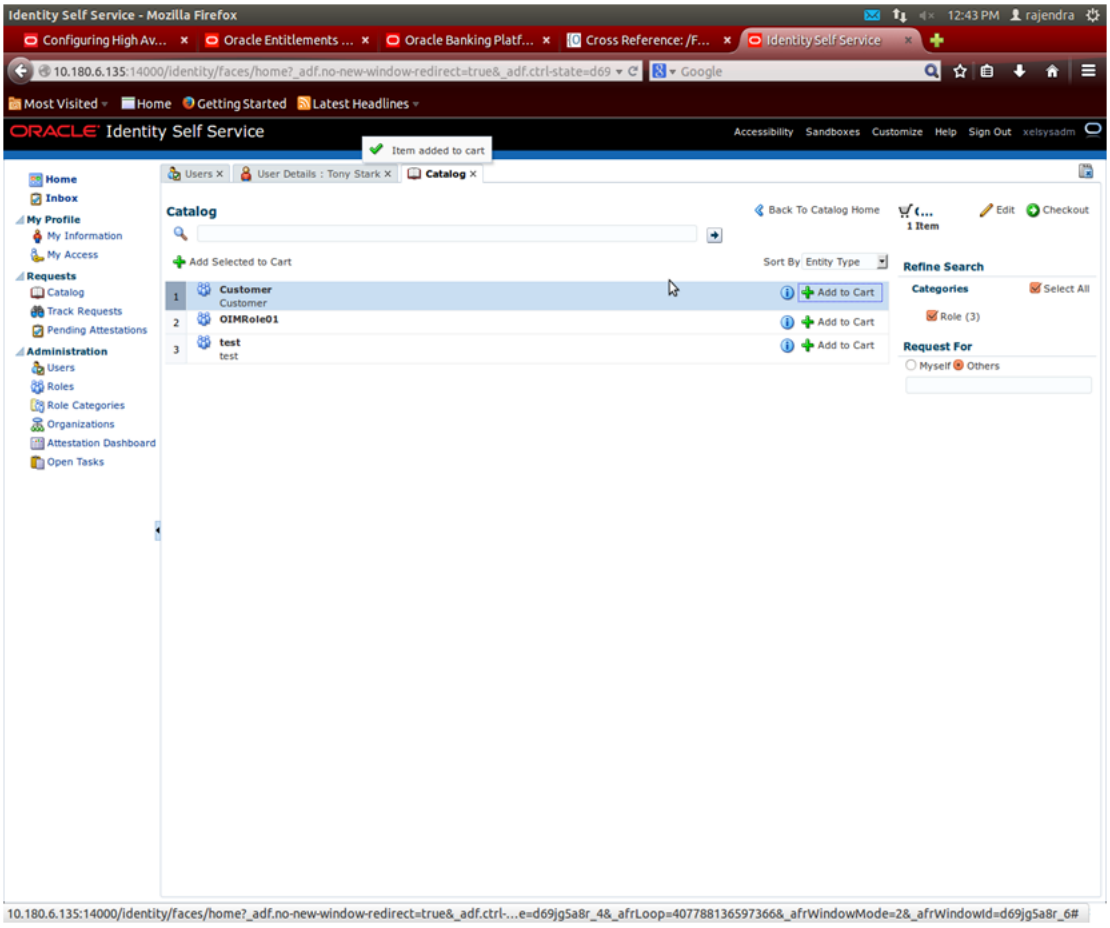
4. In the **Catalog** page, select the required role and click **Add to Cart.** The item gets added to the cart.

Figure 1–7 Adding to Cart



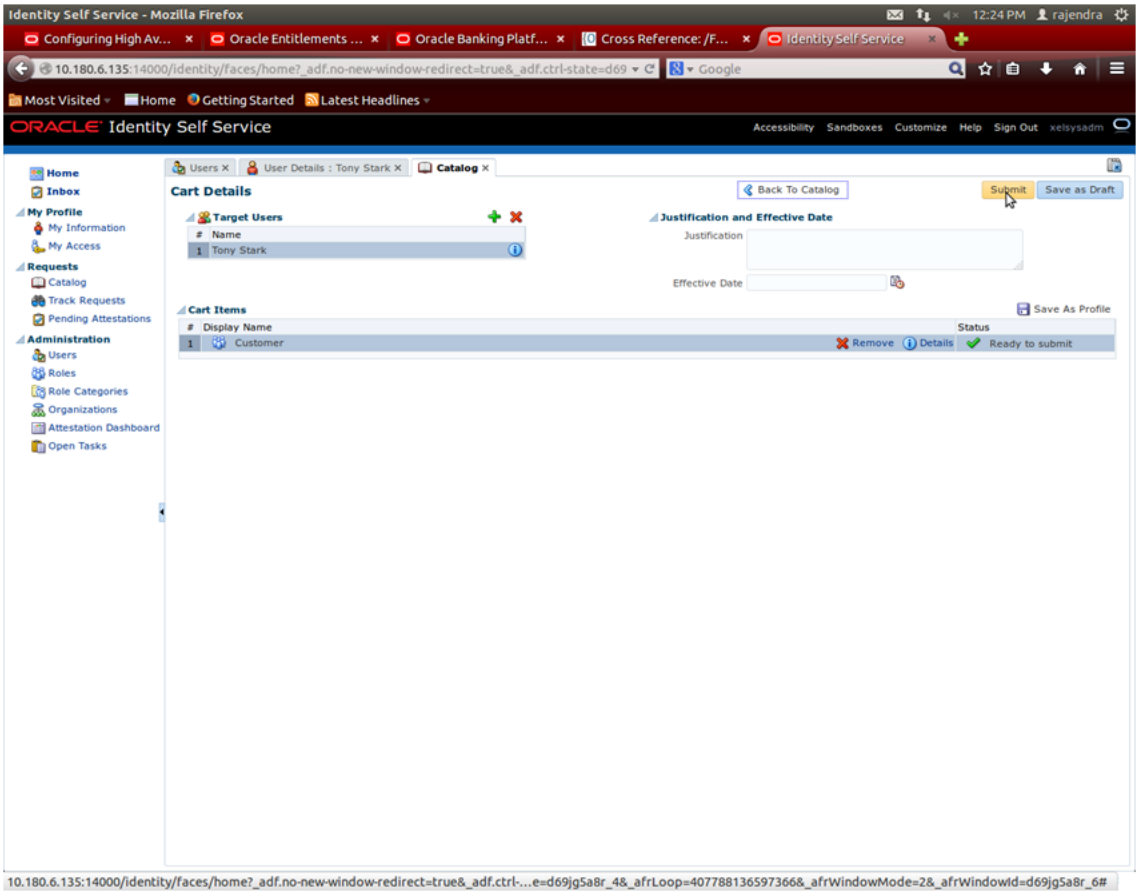
5. Click **Checkout.**

Figure 1–8 Checkout Cart



6. In the **Cart Details** page, click **Submit.**

Figure 1–9 Submit Cart



On completion of this procedure the role gets assigned to the user in OIM.

1.3 Locking Users in OIM

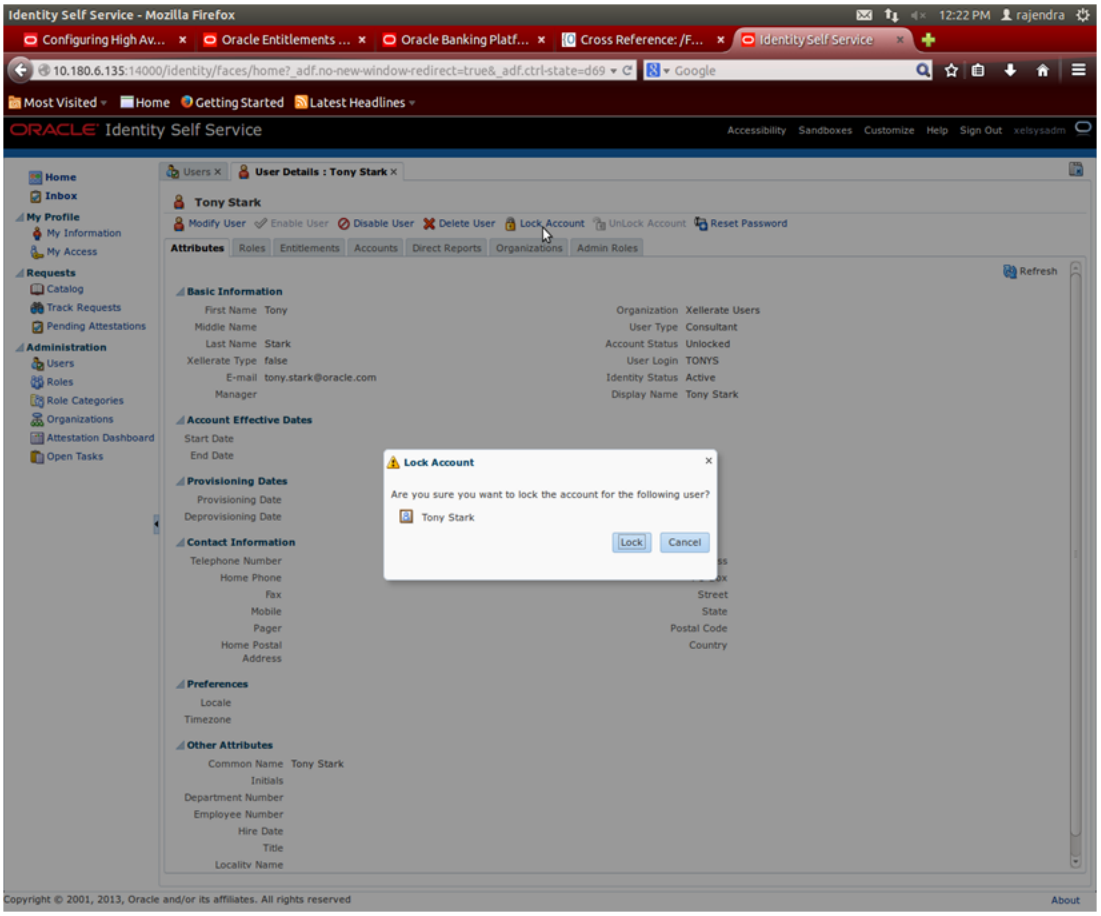
This section explains how to lock the user in OIM.

To lock a user:

1. Log in to OIM.
2. Click **Lock Account** to lock a user.

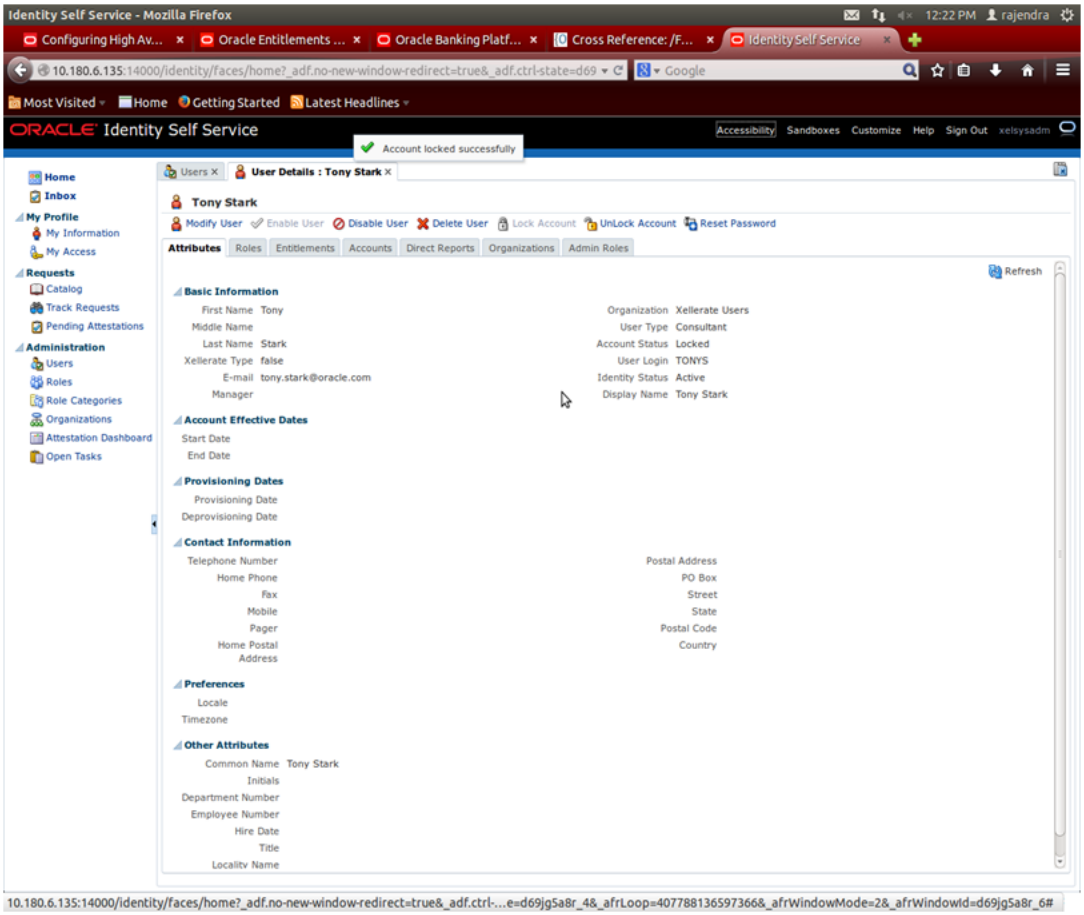
   A message appears, "Are you sure you want to lock the account for the following user?"
3. Click **Lock.**

Figure 1–10 Locking User



The user is locked successfully.

Figure 1–11 User Locked Successfully



1.4 Unlocking Users in OIM

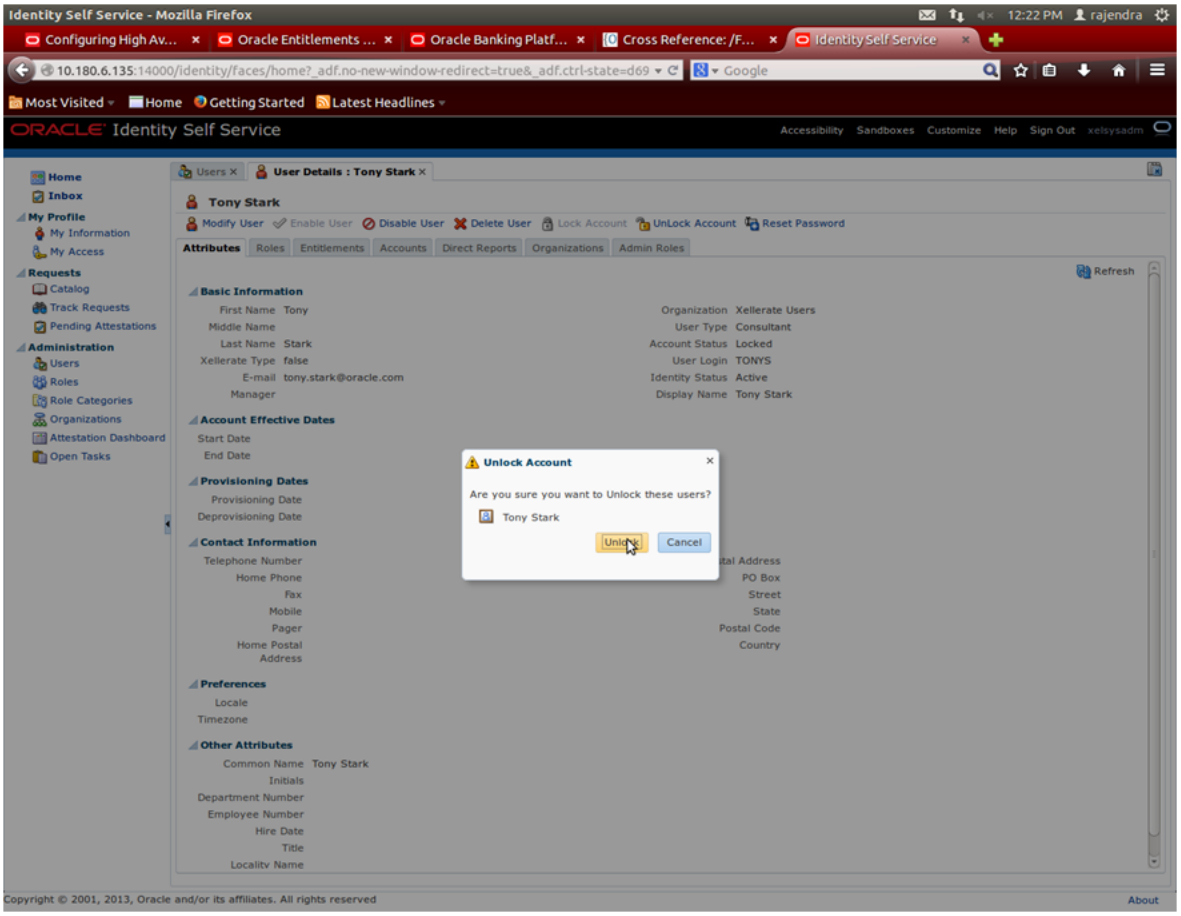
This section explains how to unlock the user in OIM.

To unlock a user:

1. Log in to OIM.
2. Click **Unlock Account** to unlock a user.

   A message appears, "Are you sure you want to Unlock these users?"
3. Click **Unlock.**

Figure 1–12 Unlocking User



The user is unlocked successfully.

1.5 User Management Using the OBP Admin Application

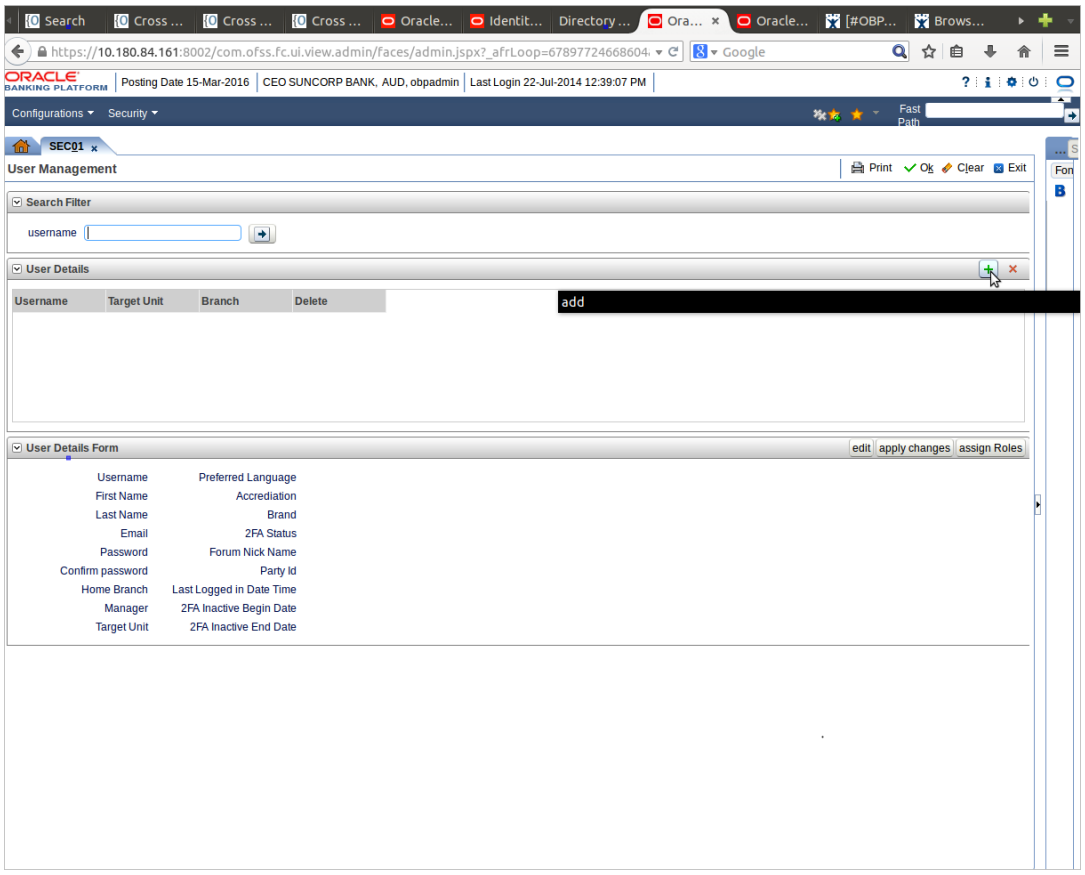
The User Management screen is a quick start UI, provided to create initial users and verify the OBP installation.

https://<ui-server-name>:<ui-server-port>/com.ofss.fc.ui.view.admin/faces/admin.jspx

To create initial users and verify the OBP installation, perform the below mentioned steps:

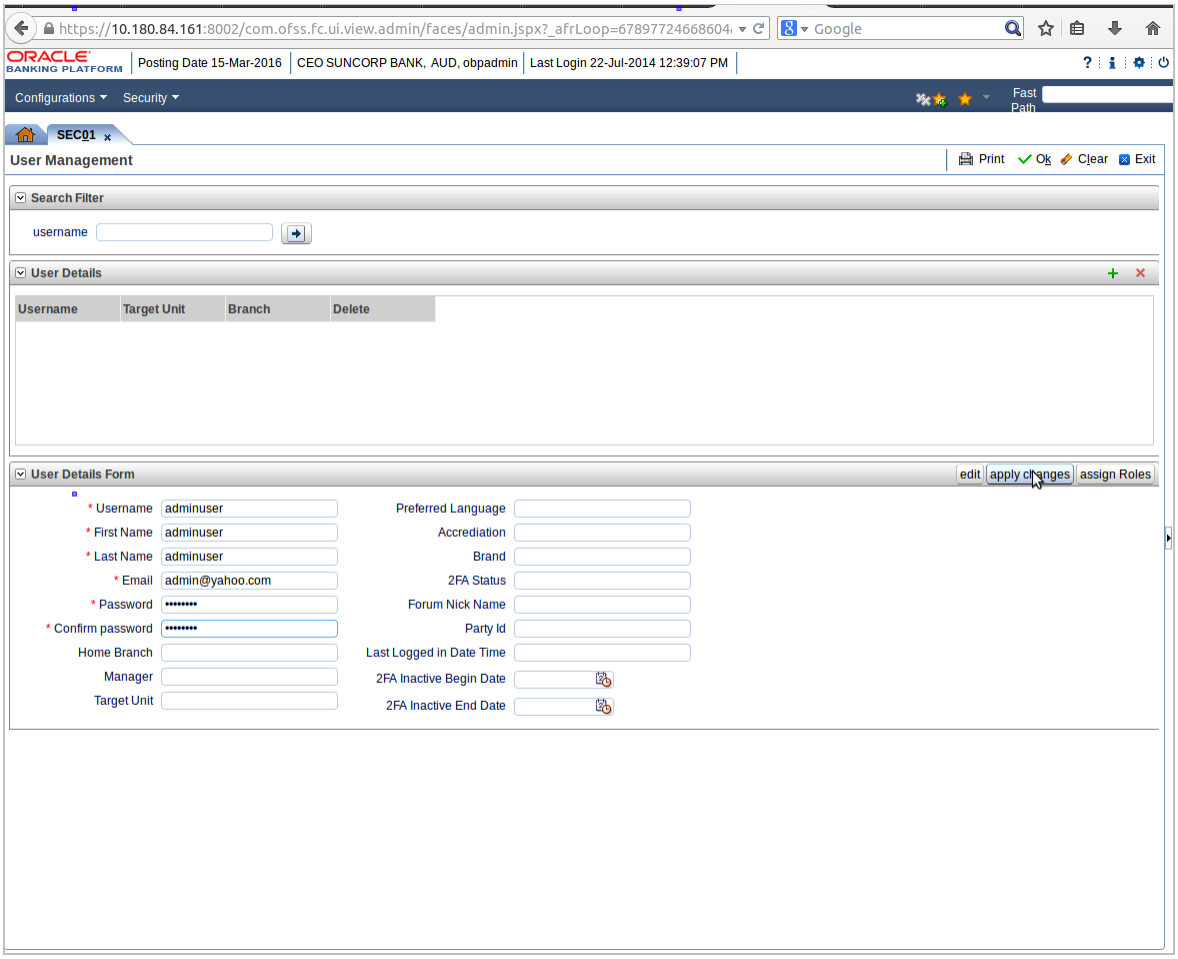
1. Click **Security** tab in **View Admin**.
2. Select **User Management**.
3. Click **+** icon to add a user.

Figure 1–13 Adding a User



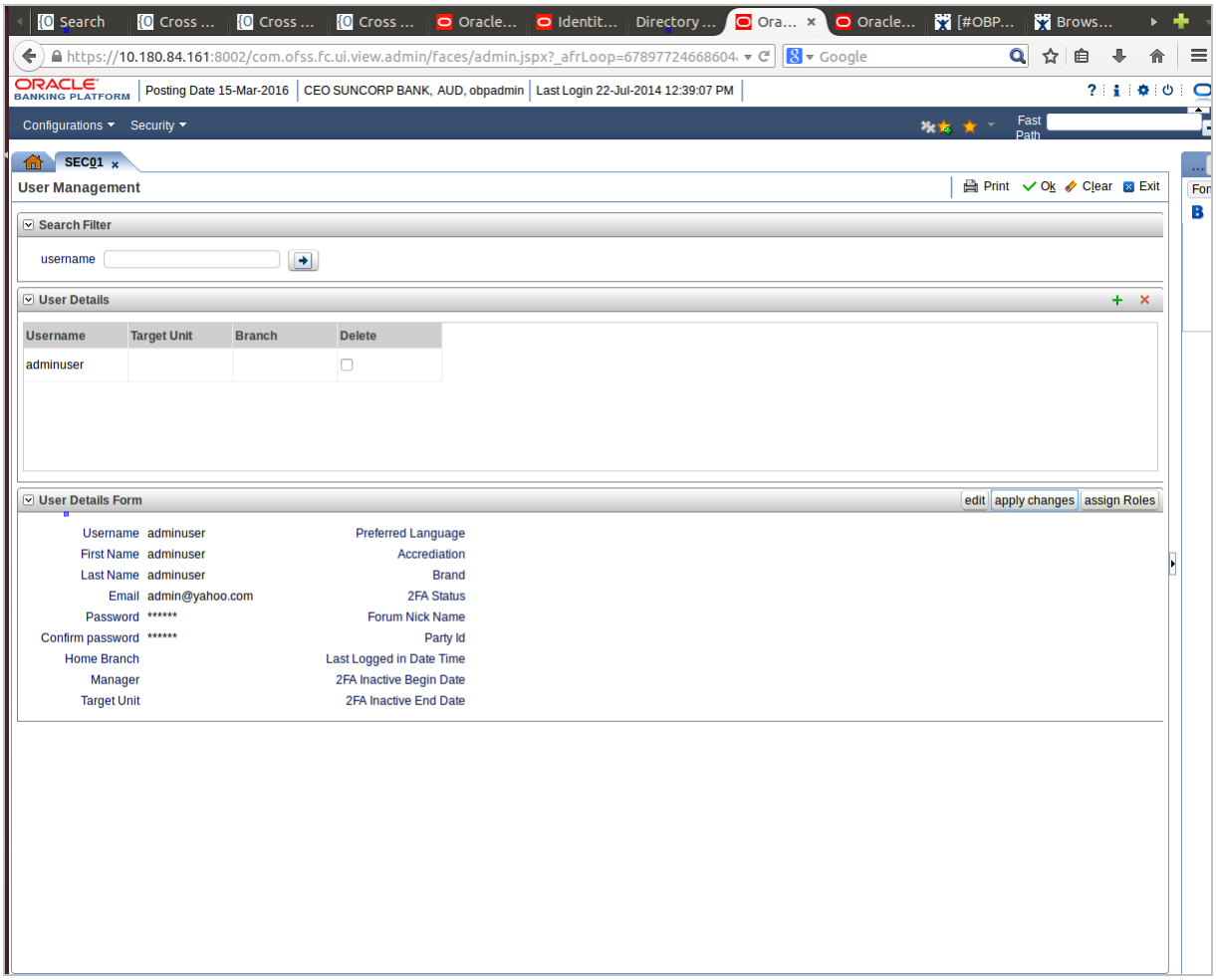
4. Fill all the mandatory fields required for creating a user.

Figure 1–14 Enter Mandatory Details



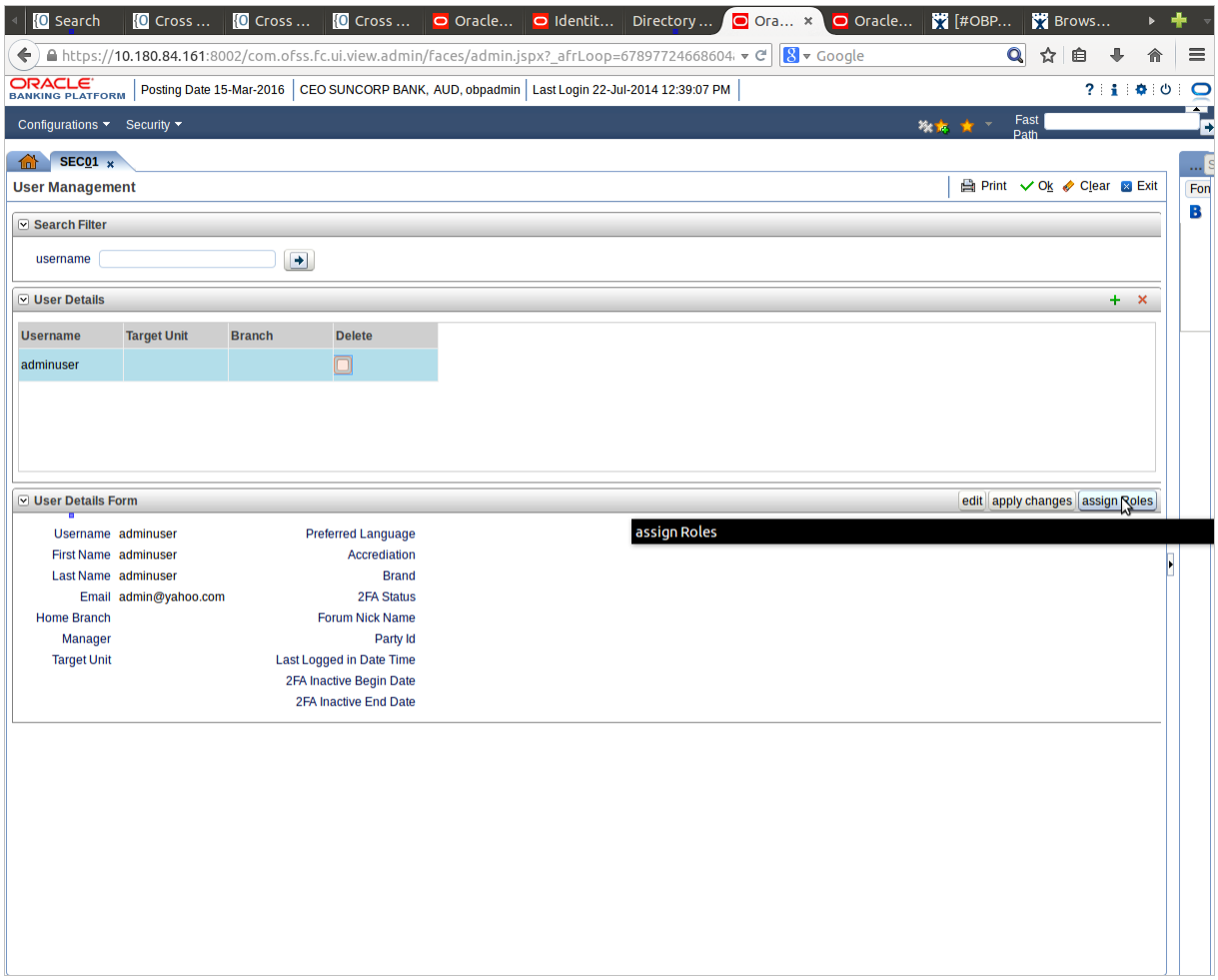
5. Click **Apply Changes** to save the user details locally.

Figure 1–15 Applying Changes



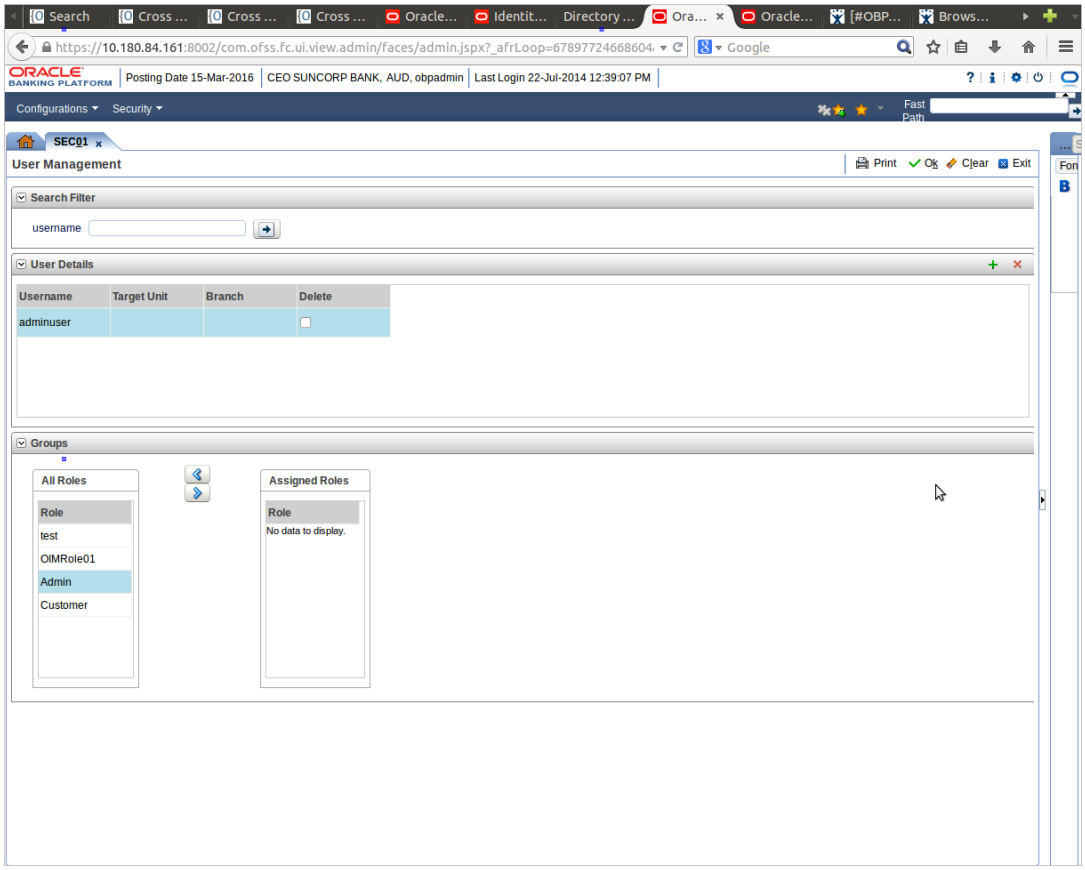
6. To add a user to a group, select the row containing the user and click **Assign Roles**.

Figure 1–16 Adding User to a Group



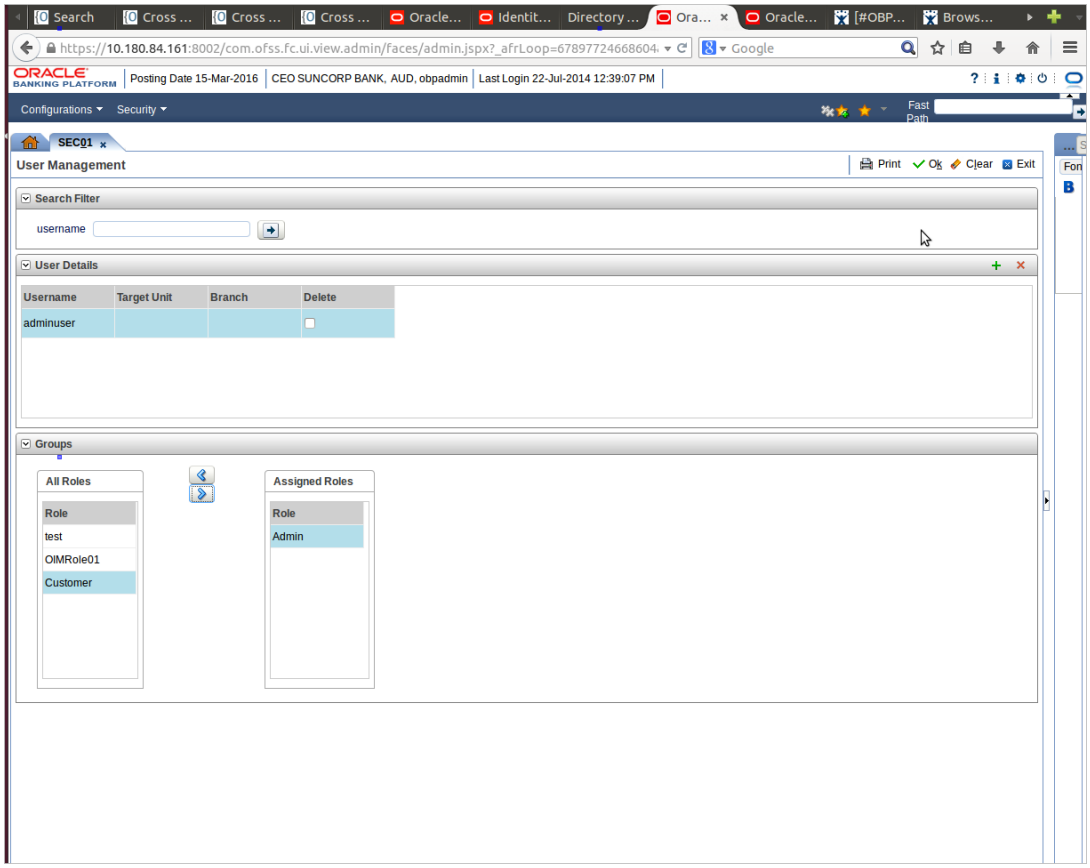
The available and assigned roles appears.

Figure 1–17 Available and Assigned Roles



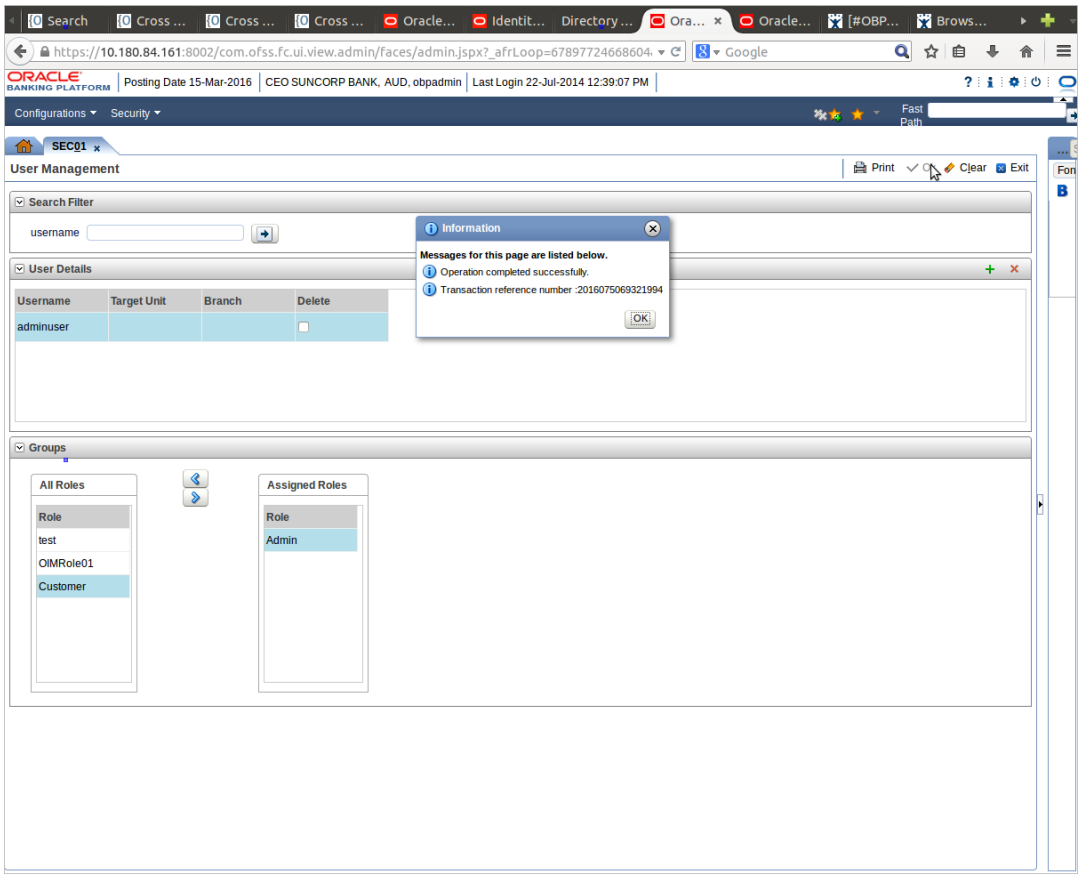
7. Select the group to add user and move it to the **Assigned Roles** table.

Figure 1–18 Adding User to Assigned Roles Table



8. Click **Ok** to save the changes.

Figure 1–19 Save Changes



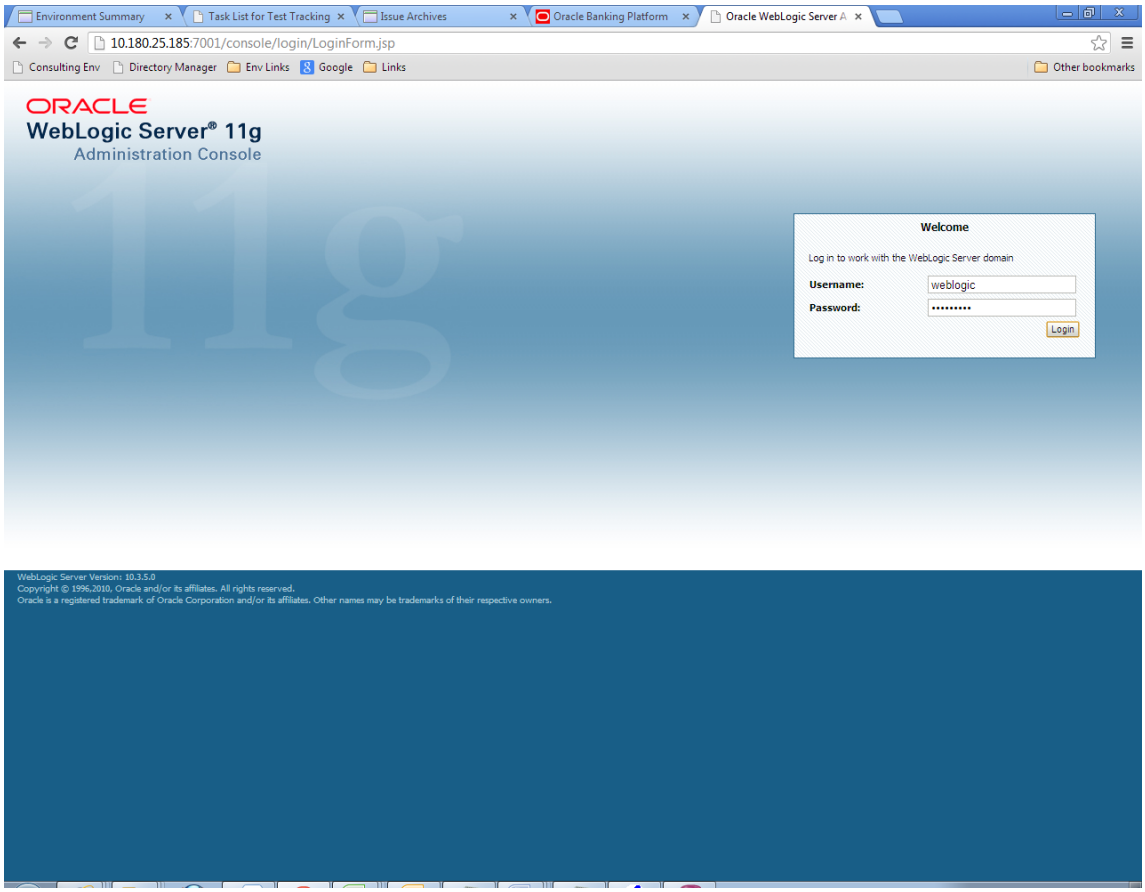
1.6 Unlocking Users in Oracle WebLogic Server (OWS) Administration Console

This section explains the procedure to unlock users in Oracle WebLogic Server (OWS) using Administration Console. If users unsuccessfully attempt to log in to a WebLogic Server instance for more than the configured number of retry attempts, they are locked out of further access. This procedure allows you to unlock locked users so that they can log in again.

To unlock a user in OWS:

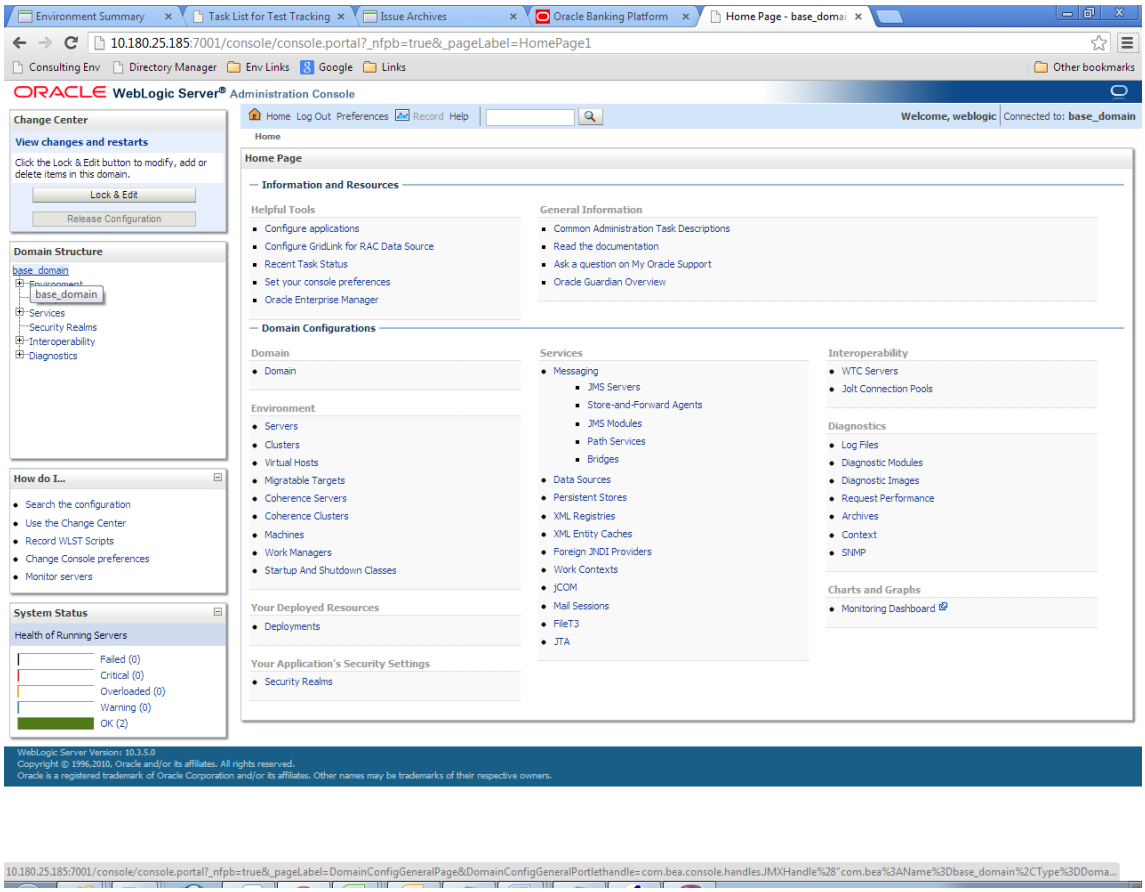
1. Log in to OWS. The **Home Page** of OWS Administration Console appears.

Figure 1–20 OWS Log in



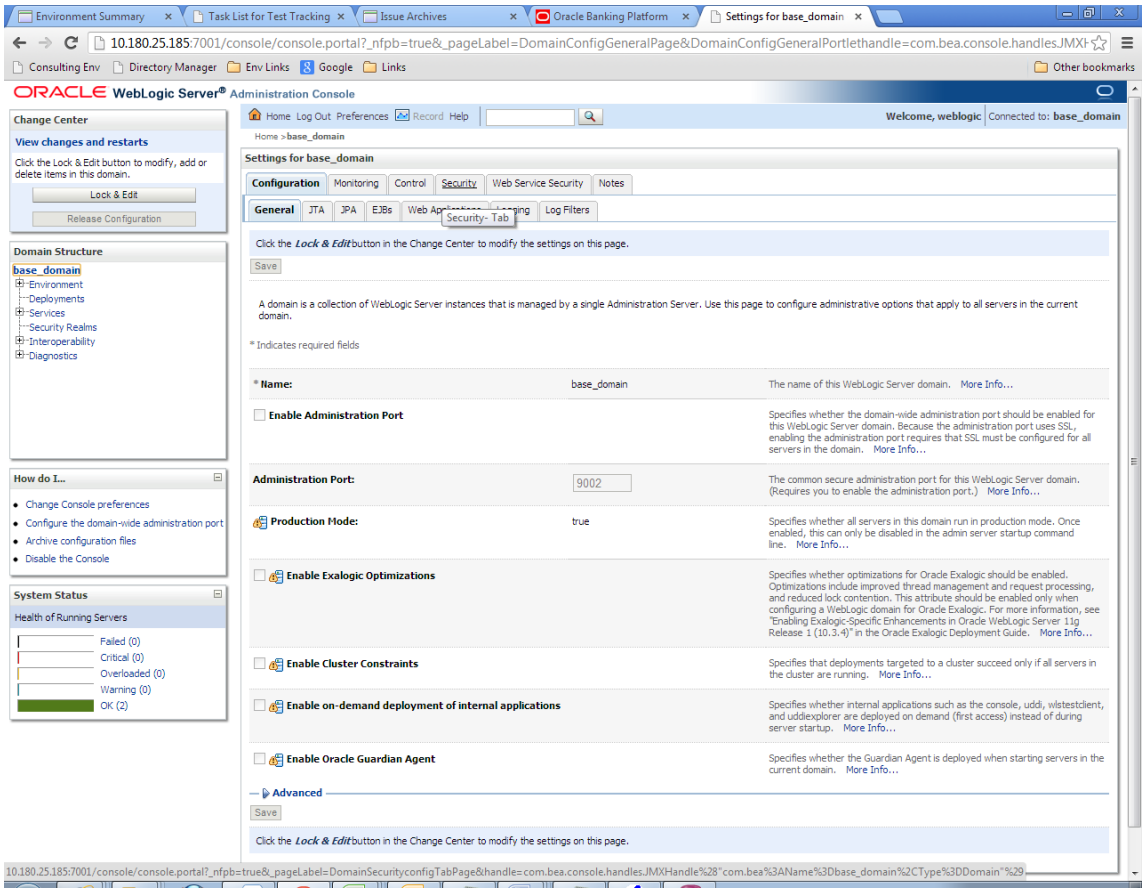
2. In the **Domain Structure** section, click the **base_domain** link.

Figure 1–21 base_domain



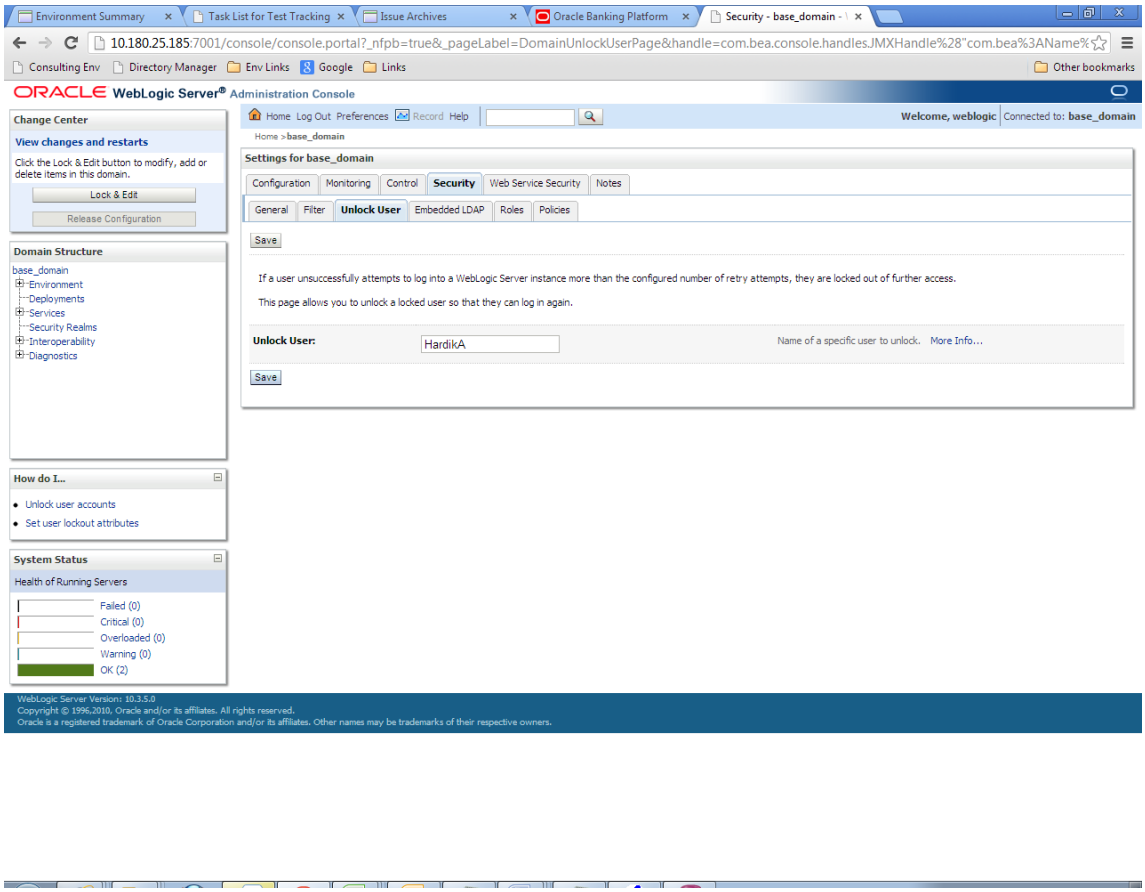
3. In the **Settings for base_domain** page that appears, click the **Security** tab.

Figure 1–22 Security tab



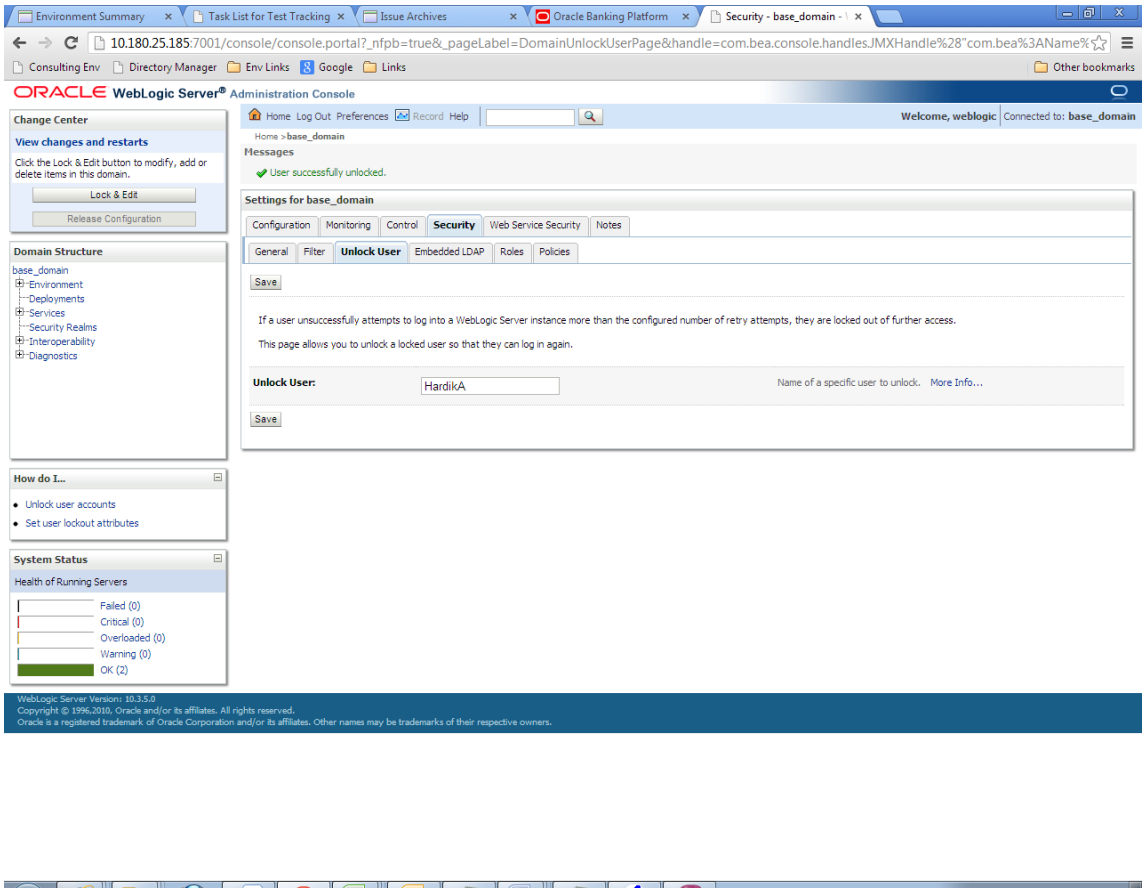
4. Click the **Unlock User** tab.
5. In the **Unlock User** field, enter the User ID to unlock the user.

Figure 1–23 Unlock User



6. Click **Save.** The message *User successfully unlocked* appears.

Figure 1–24 User Successfully Unlocked



On completion of this procedure the user gets unlocked in OWS.

2

Approvals Management

This chapter describes Discretionary Pricing Assessment (DPA) approvals, manual credit decision approvals and worklist authorization related activities to be performed as an administrator.

This chapter includes the following topics:

2.1 Discretionary Pricing Assessment (DPA)

This section explains the procedure for Discretionary Pricing Assessment (DPA) approvals.

Overview

DPA can be configured for fee negotiations happening in an account during online transactions as well as for UDM. DPA service is to be called from the respective screens for DPA rule resolution and authorization functionality.

Fee can be configured in following list of modules and functions. DPA service will be called from the following screens:

Table 2–1 List of Modules and Functions for Fee Configuration

Sl No	**Loans**
1	Offer Swap
2	Re-draw
3	Close Account
4	Offset
5	Disbursement
6	OD to Loan
7	Consolidation
8	Adhoc Fee
9	Partial Payoff
10	Partial payoff from RPA

Table 2–1 (Cont.) List of Modules and Functions for Fee Configuration

	CASA
11	Cash Deposit
12	Adhoc Fee
13	Cheque Issuance
14	Cash Withdrawal
15	Funds Transfer
16	Standing Instructions
17	Sweep in Instruction
18	Sweep out Instruction
19	Multi Fund Deposit
20	Offer Swap
21	Close Out Transfer
22	Stop Payment
	TD
23	Redeem Term Deposit (TD015)
	Origination
24	Loans
25	CASA
26	TD
27	OD
	Payments
28	Payments and Collections (PC216)
29	Bank Draft Issue (PC241)
30	Bank Draft Status Maintenance - Lost
31	Bank Draft Status Maintenance - Damaged
32	Bank Draft Liquidation - Refund
33	Bank Draft Liquidation - Cancel
34	Bank Draft Status - Stop
	ELPBC
35	ALPBC
36	FLPBC

Fee amount applicable for the event is displayed in the fee panel in the respective screens. If there is any negotiation (upward or downward) that happens in the fee panel, the relevant issues have been raised/postponed for the Patch set release.

UDM can be configured in following list of modules and functions. DPA service will be called from the following screens:

Table 2–2 List of Modules and Functions for UDM Configuration

Sl No	**Loans**
1	Offer Swap
2	OD to Loan

Table 2–2 (Cont.) List of Modules and Functions for UDM Configuration

3	Consolidation
4	Split Loan
5	New Account Opening

2.1.1 Setup Details

This topic discusses the setup details required to configure the DPA services.

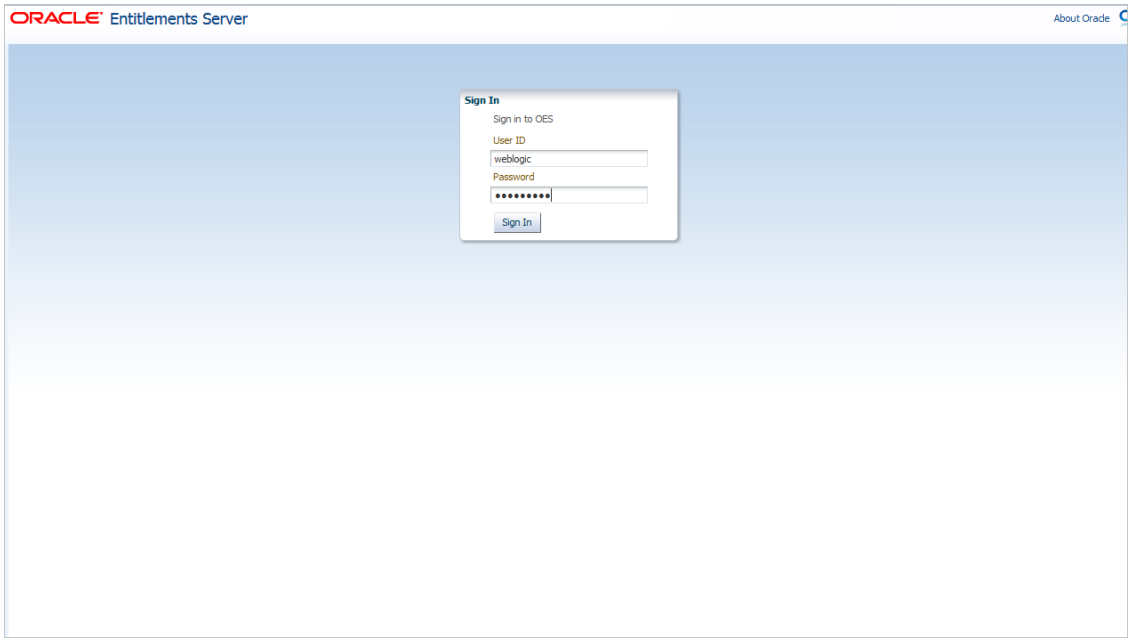
2.1.1.1 APM Policy Setup

To initiate, the user needs to set up policies at APM for auto approval. This is the policy which specifies which transactions with what data will be auto approved. When the data that is sent from the application matches the policy setup, the transaction is auto approved and if the policy is not matched, then the transaction goes for approval.

Following is the procedure to be followed during APM policy setup:

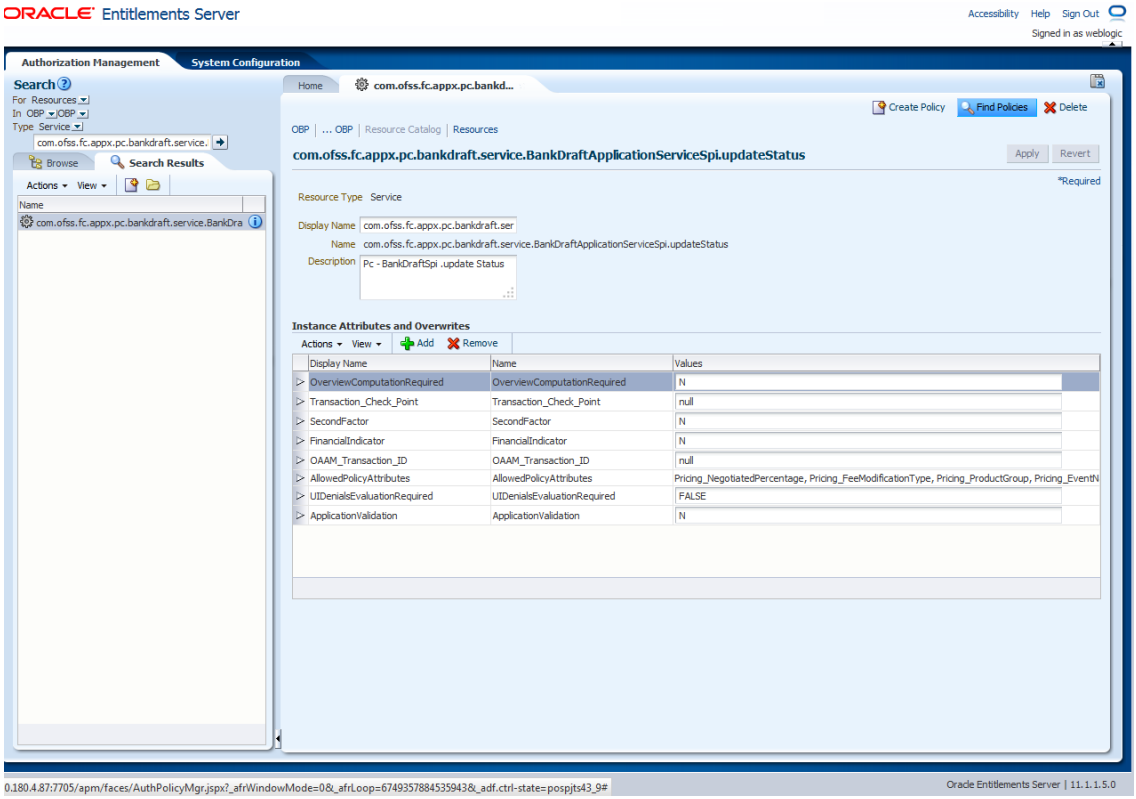
1. Log in to the APM as a user with administrator privileges.

Figure 2–1 Log in to APM



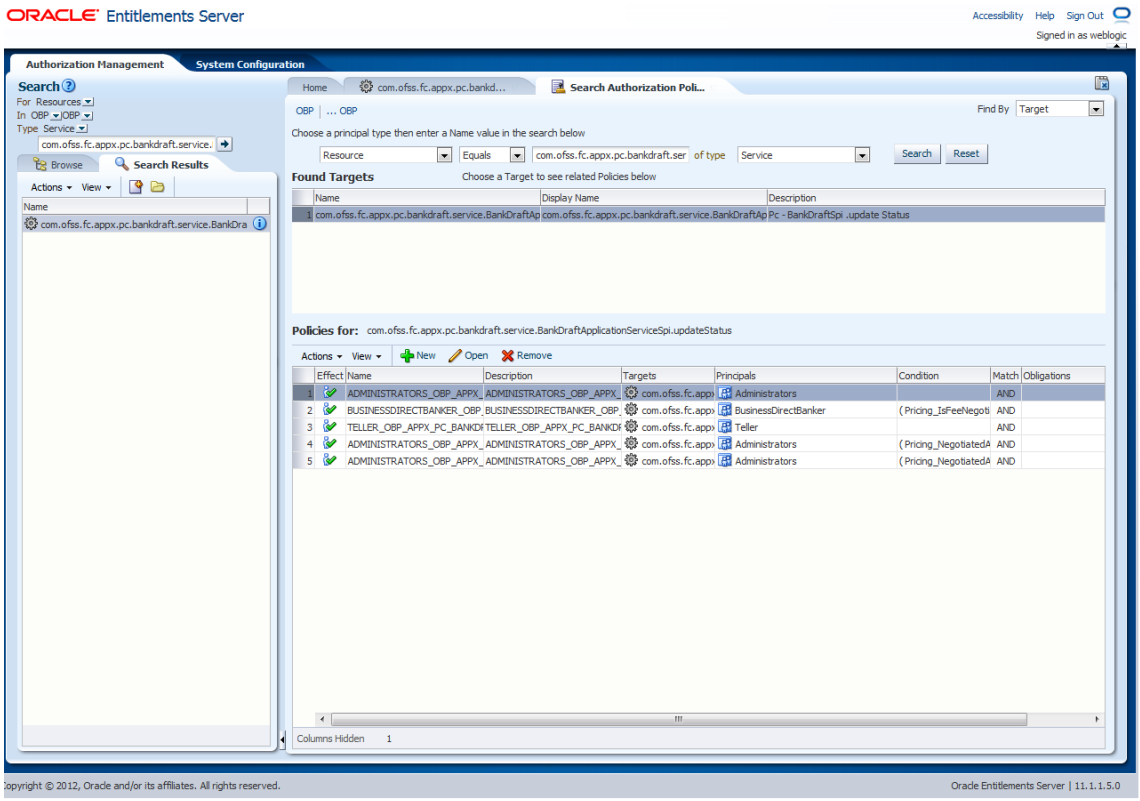
2. Search for the requisite service and click **Find Policies.**

Figure 2–2 Search for Policies



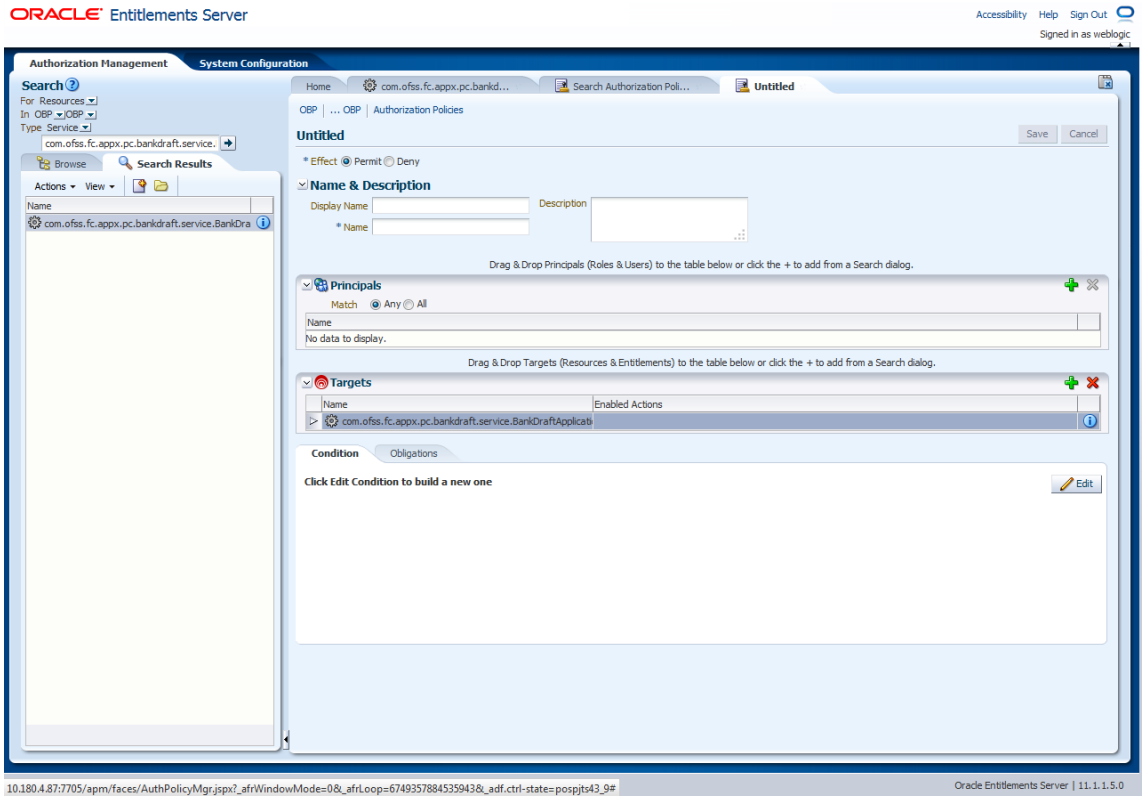
Clicking **Find Policies** opens the below screen.

Figure 2–3 Find Policies



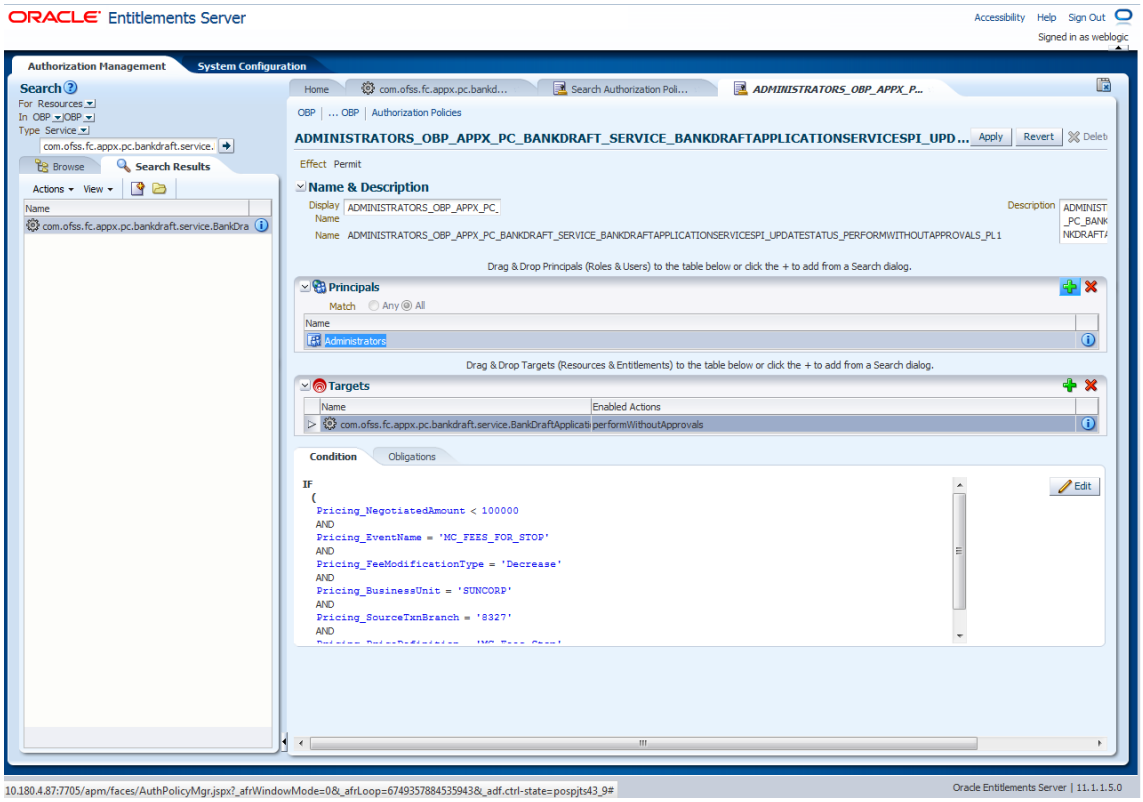
3. Click **New.**
4. Add new policy as shown in Figure 2–4.

Figure 2–4 Adding New Policy



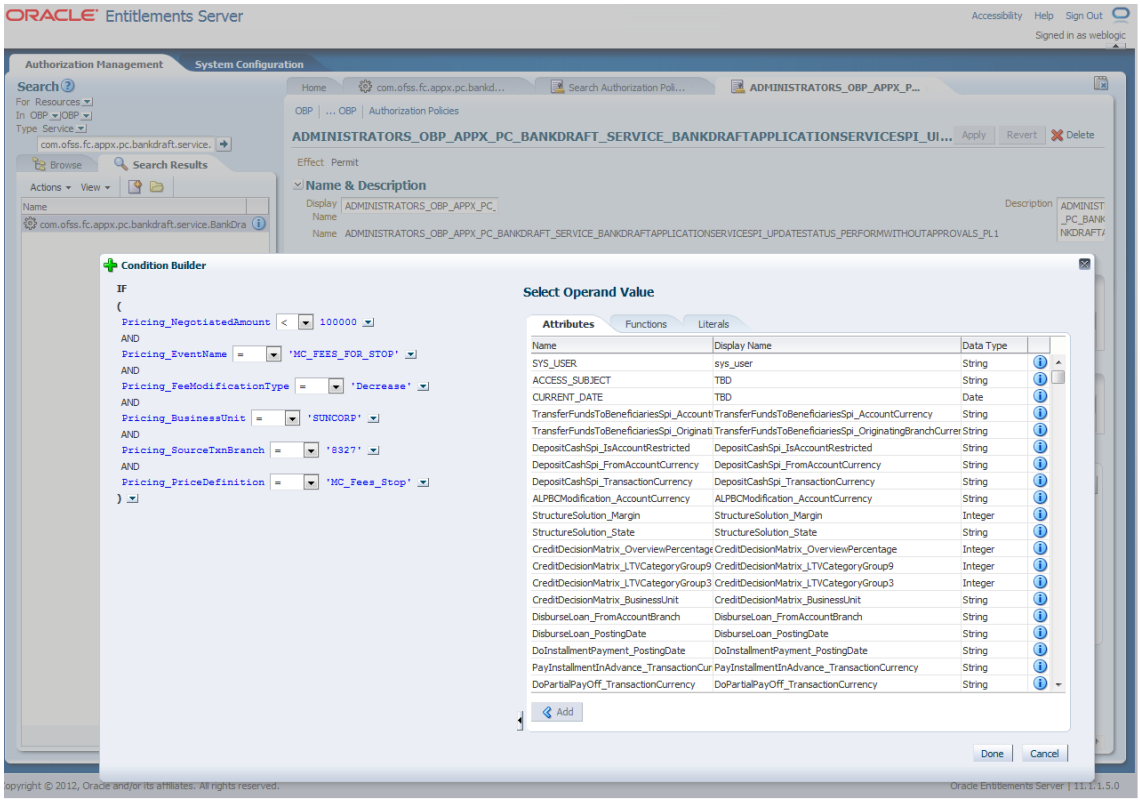
5. Add the following details:
   - **Name and Description**
   - **Principals**
   - **Enabled Actions** as *Perform without approvals*

Figure 2–5 Adding Name, Principal and Enable Action Details



6. Click **Edit.**
7. Add a new policy.
8. Click **Done.**
9. Click **Apply.**

Figure 2–6 Adding a New Policy



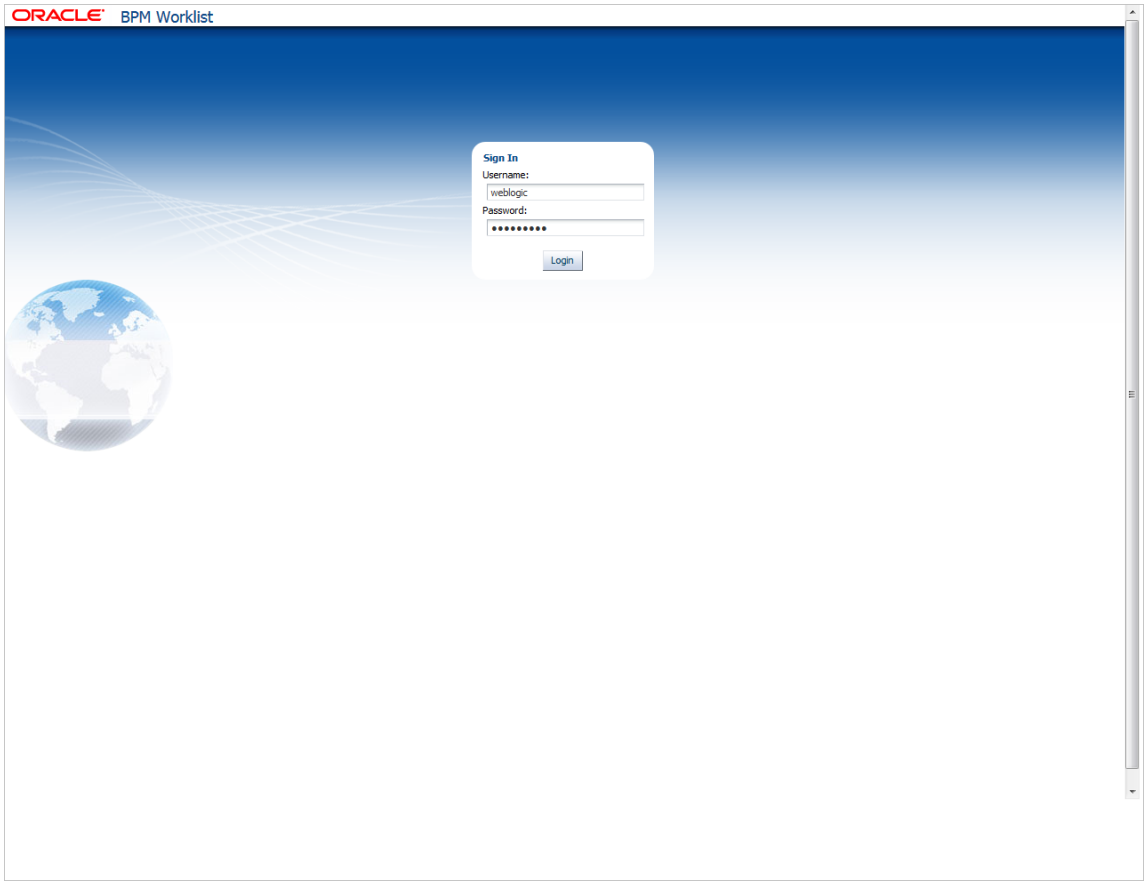
2.1.1.2 BPM Worklist Rules Setup

After the rules are set up in APM, the user needs to set up approval rules in BPM worklist. These rules should cover all scenarios in which the transaction can come out without being auto approved at the APM level. Any transaction which does not trigger the rules at BPM level is auto rejected.

Following is the procedure to be followed during BPM worklist rules setup:

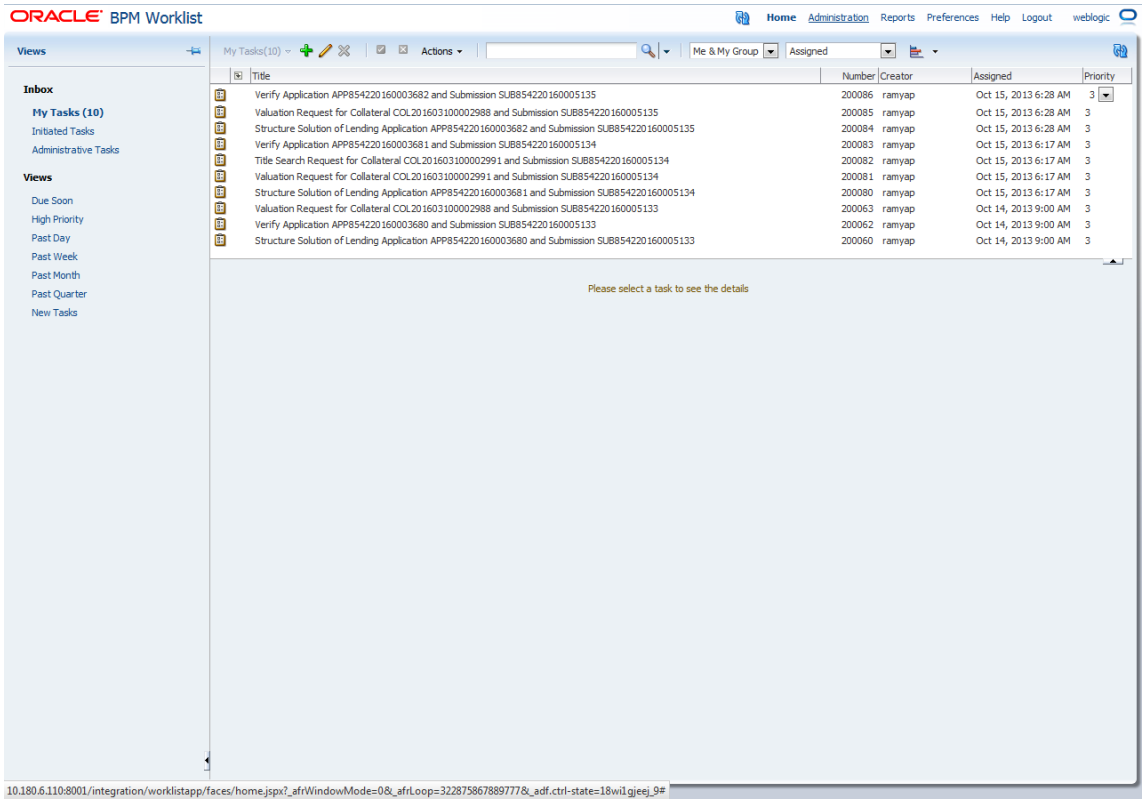
1. Log in to the BPM worklist application as a user with administrative privileges.

Figure 2–7 Log in to BPM



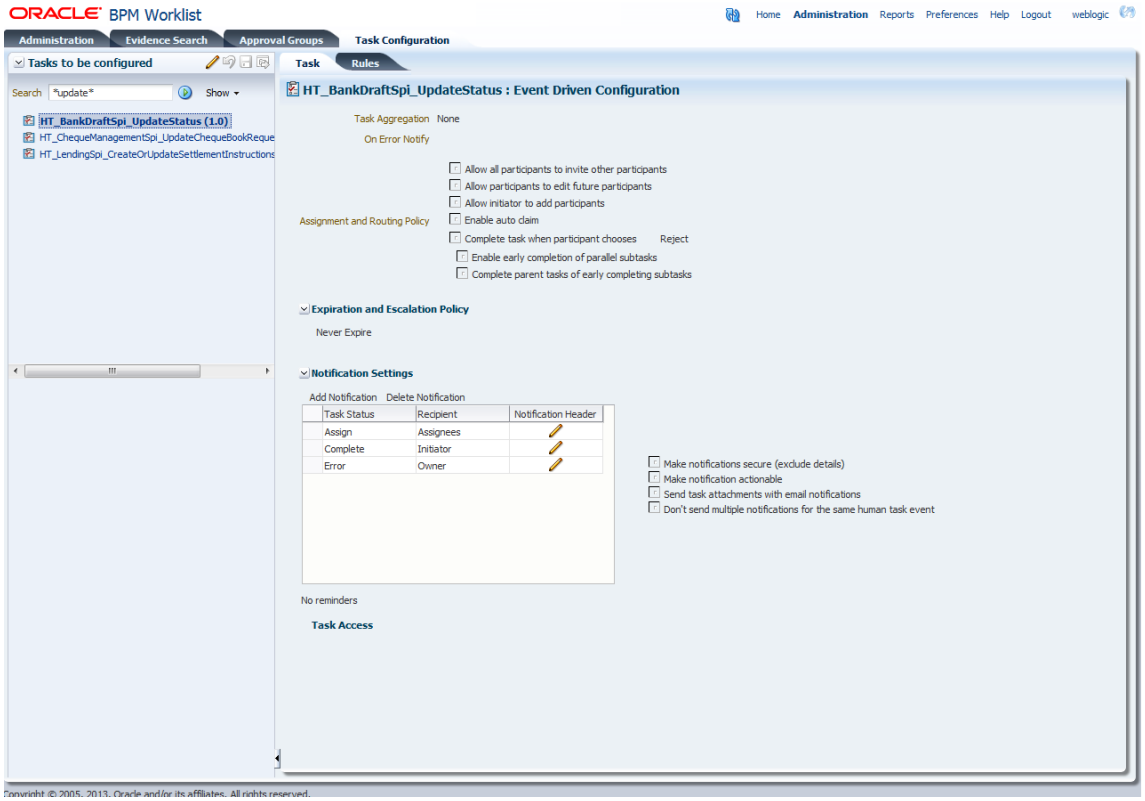
2. Click the **Administration** link in the top left corner.

Figure 2–8 Click Administration Link



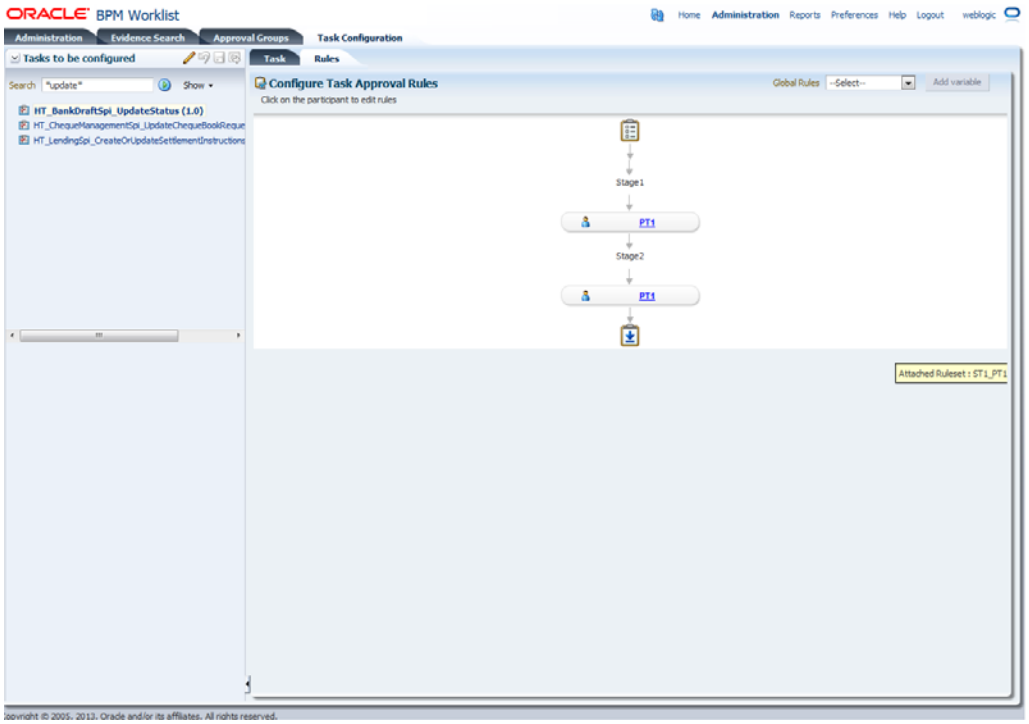
3. Open the **Task Configuration** tab and search for the specific process.

Figure 2–9 Searching Specific Process



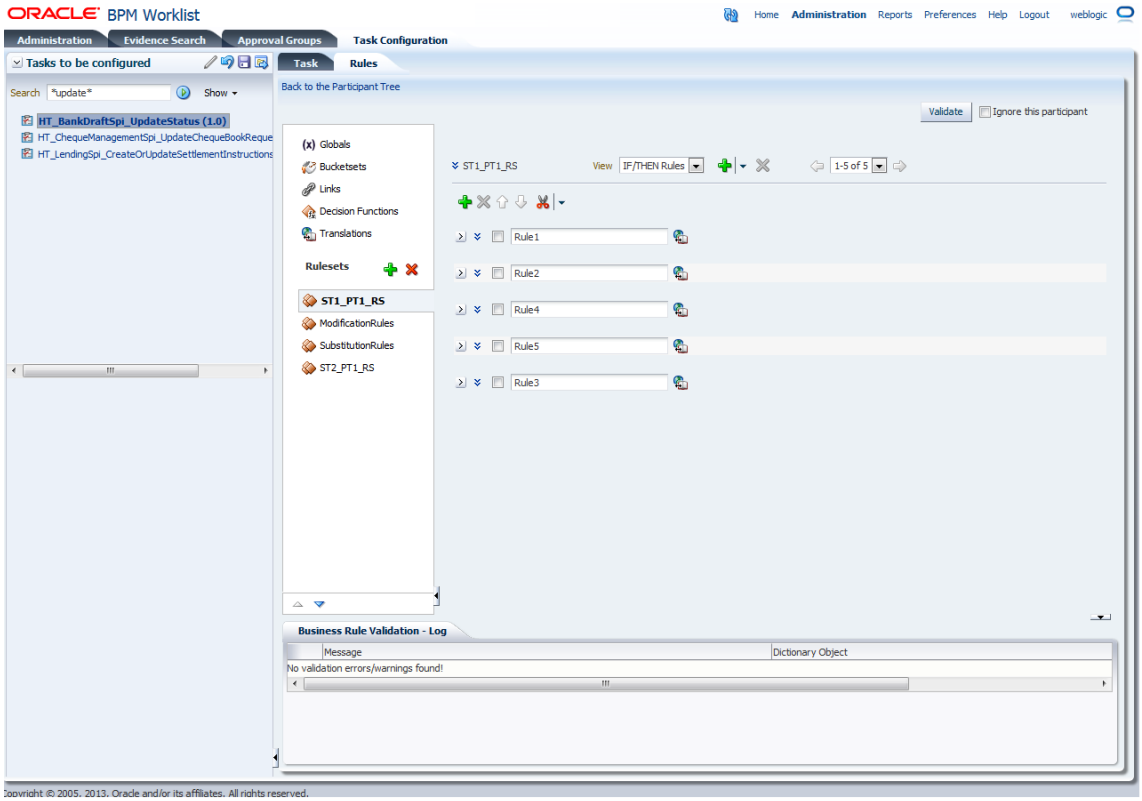
4. Open the Rules tab and click the links (PT1, PT2 and so on) provided for any stage of your choice.

Figure 2–10 Clicking the Rules Link



5. Click the **Edit** (pencil) icon for creating the rule and the following will be visible.

Figure 2–11 Creating the Rule

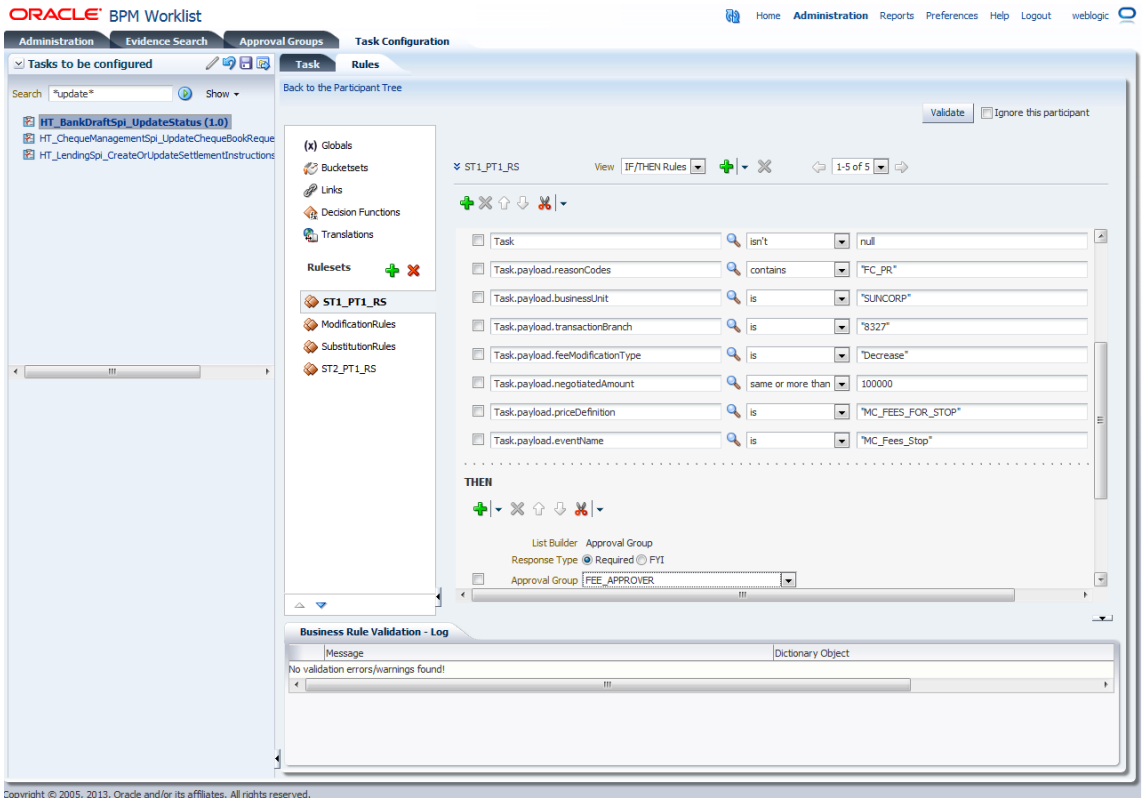


6. Select the rule and add attributes to the rule.

> **Note:** The rule being created must be the Active rule. Only one rule can be active at a time.

7. Save and commit the rule.

Figure 2–12 Selecting Rule



2.1.1.3 SMS Setup

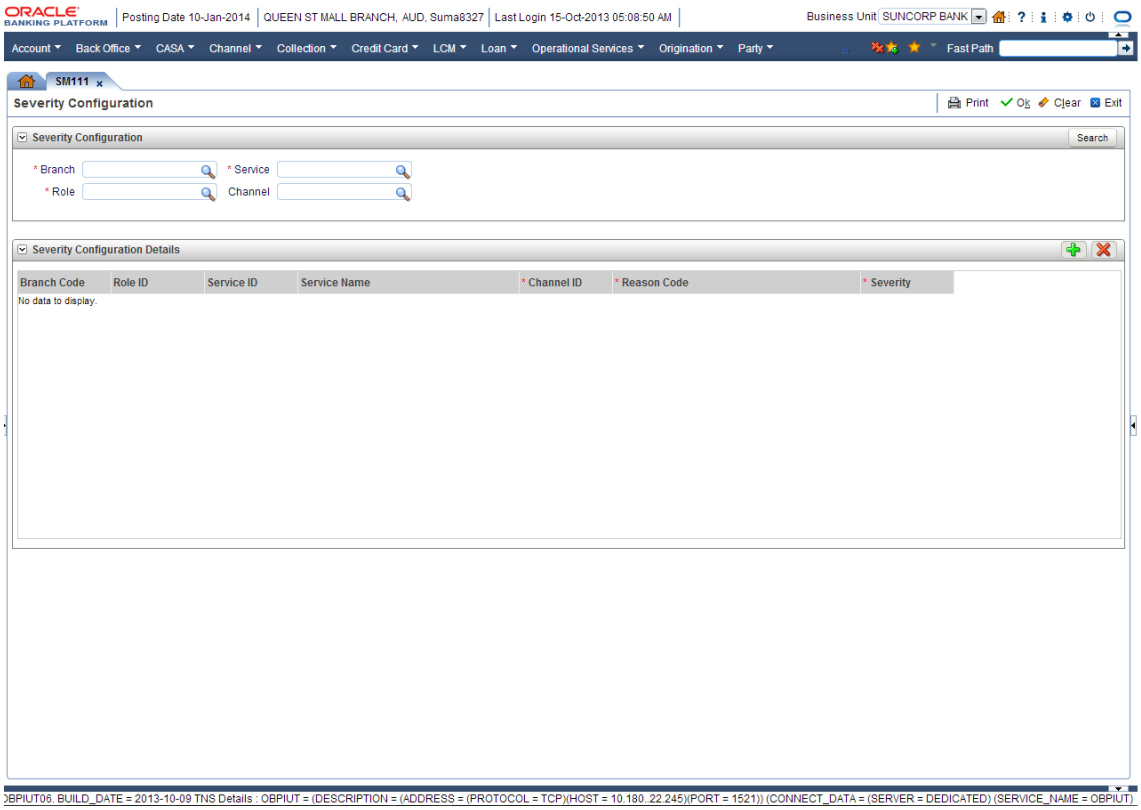
The user also needs to set up overridable exception for the application to send the transaction to worklist application. This can be achieved through the **Severity Configuration (Fast path: SM111)** screen.

> **Note:** The given procedure provides details on SMS setup for Fees. The same procedure can be followed for SMS Setup for UDM with UDM related details.

Following is the procedure to be followed during SMS setup:

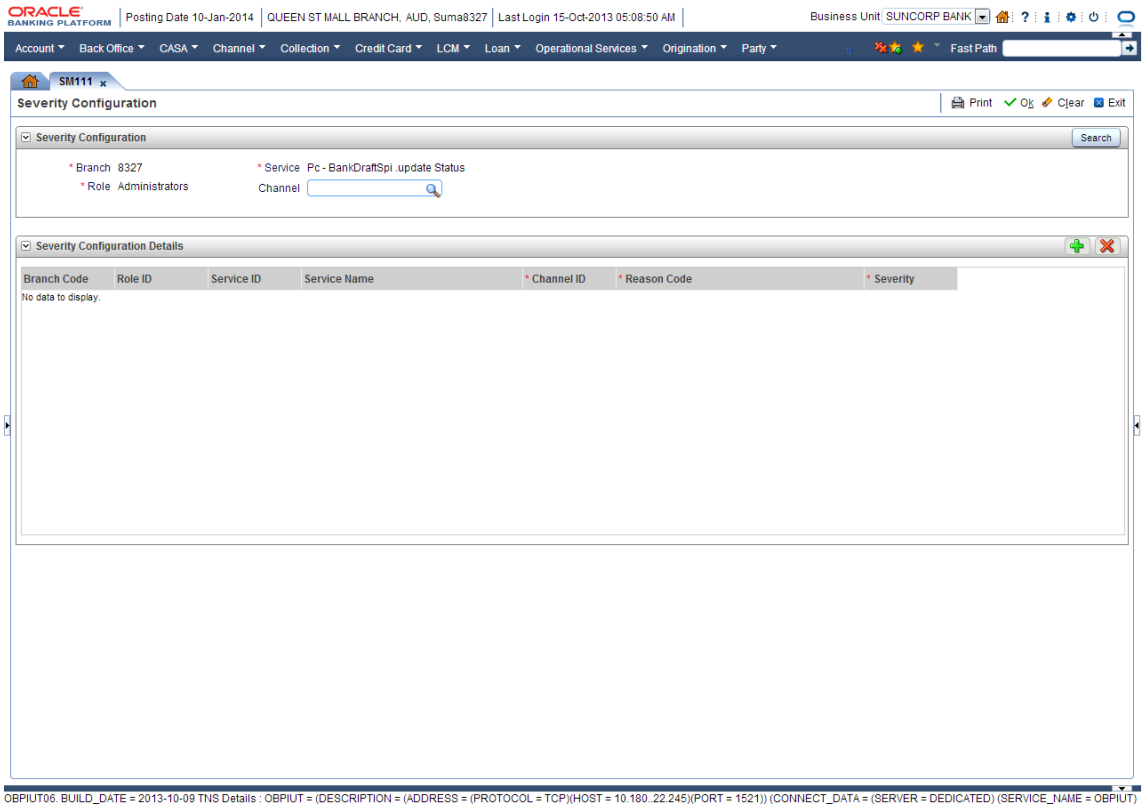
1. Log in to the OBP application as admin user and open the **Severity Configuration (Fast path: SM111)** screen.

Figure 2–13 Log in to Severity Configuration



2. Enter the following details:
   - **Branch**
   - **Role**
   - **Service**

Figure 2–14 Entering Branch, Role and Service

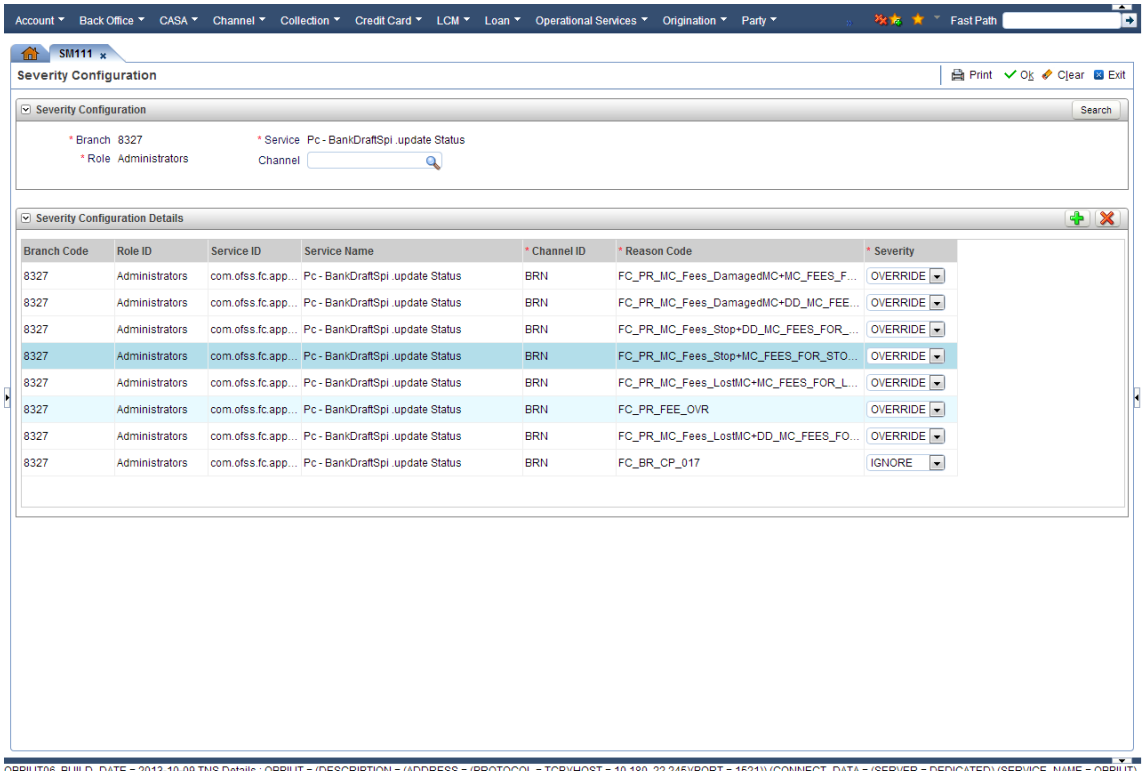


3. Click **Search** and add exception for FC_PR_FEE_OVR and the specific price definition.

> **Note:** The exception for UDM is FC_PR_UDM_OVR.

4. Save the setup.

Figure 2–15 Adding Exception

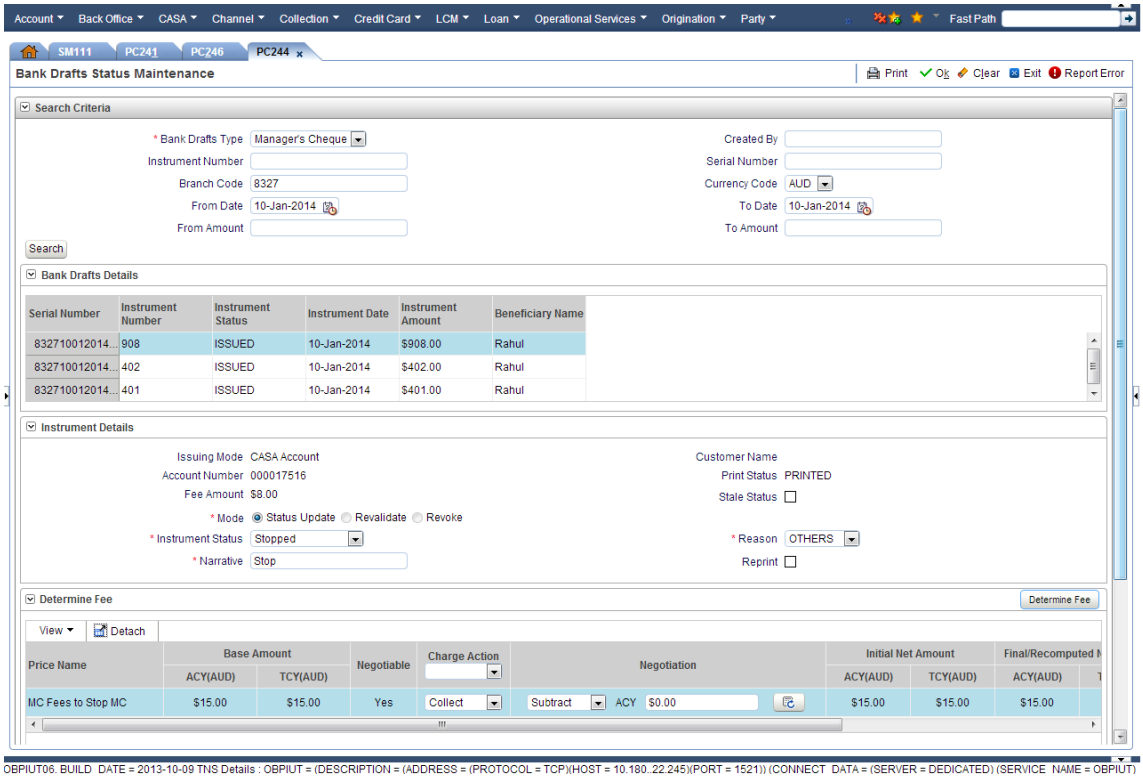


2.1.2 Performing the Transaction

Following procedure explains how to perform a transaction:

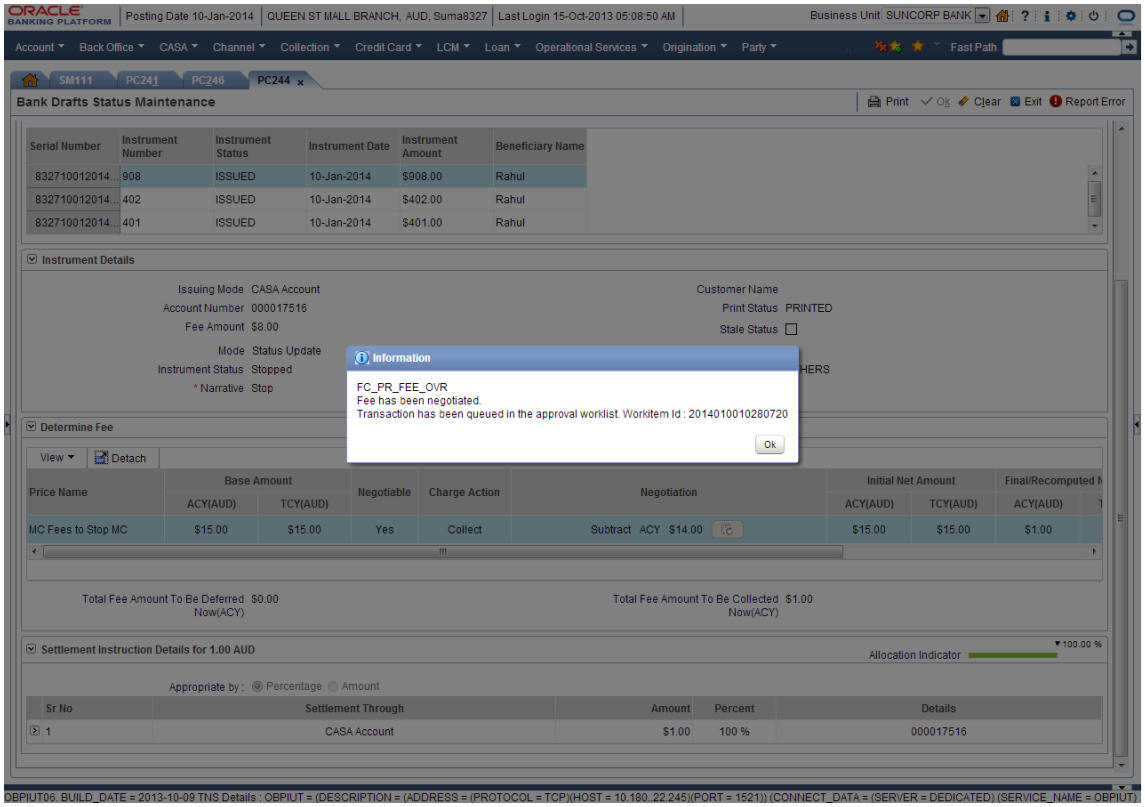
1. Log in to the OBP application as teller user and initiate the transaction.

Figure 2–16 Initiating Transaction



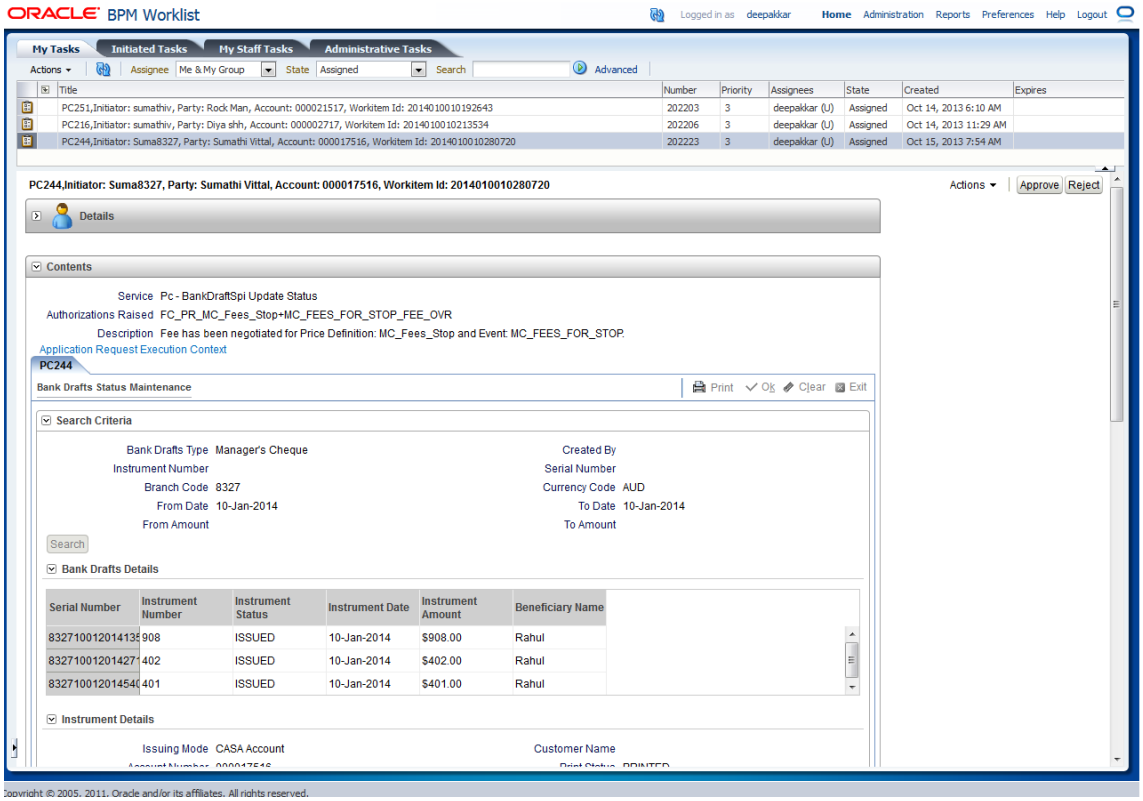
2. Negotiate fee beyond auto approval level so that the transaction goes for approval.

Figure 2–17 Fee Negotiation



3. Log in to the approval worklist as approver user and open the work item.

Figure 2–18 Log in to Approval Worklist



4. User can either approve or reject the work item (approved here) after adding comments.

Figure 2–19 Adding Comment to work item

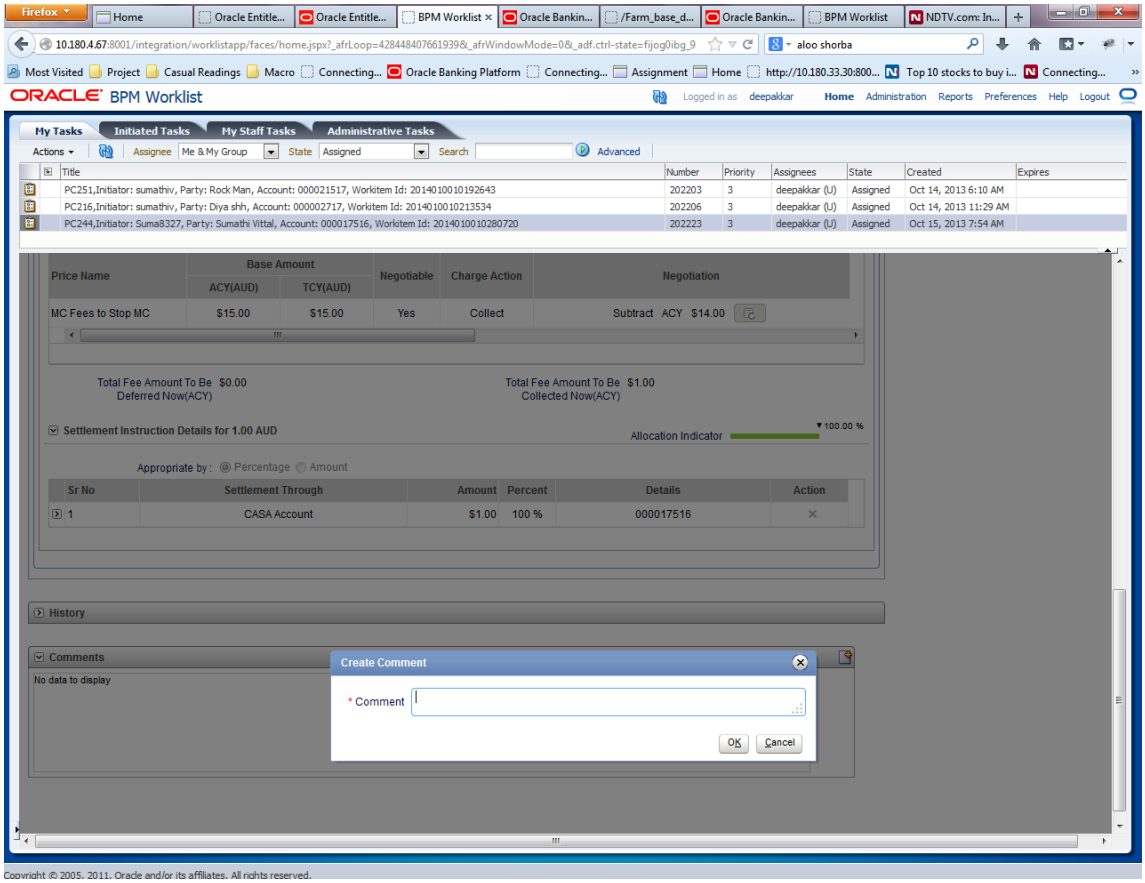
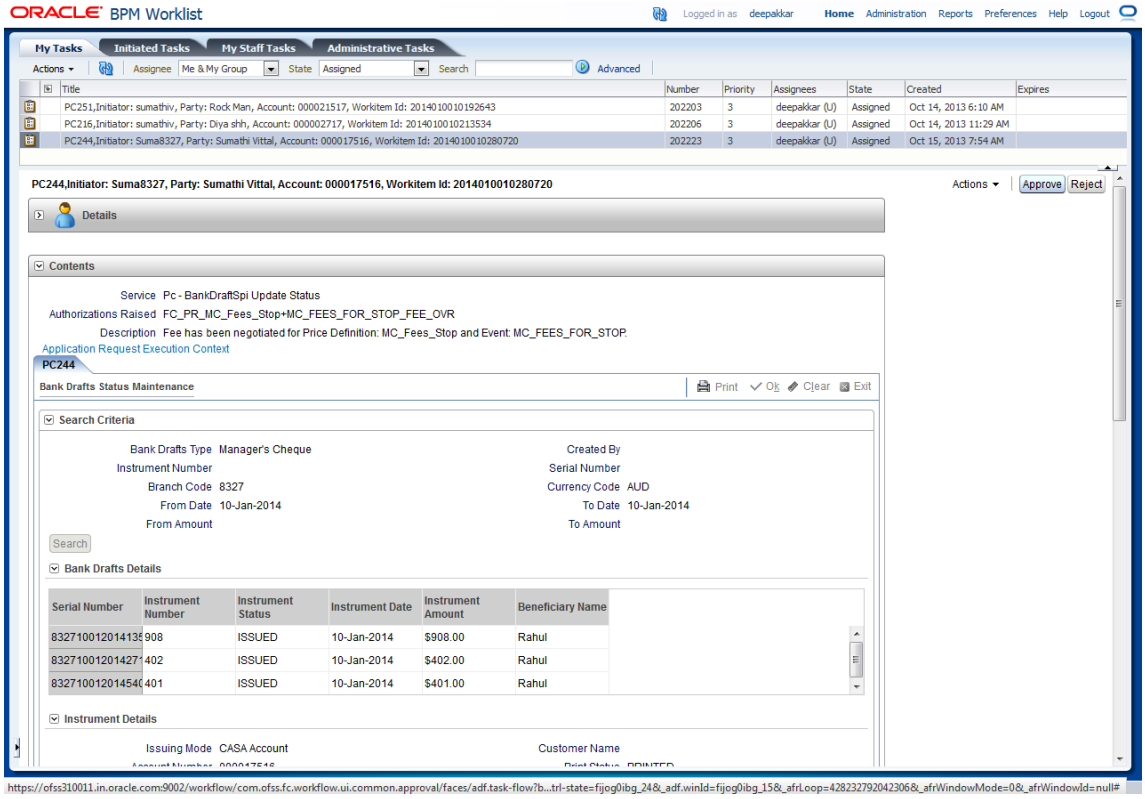
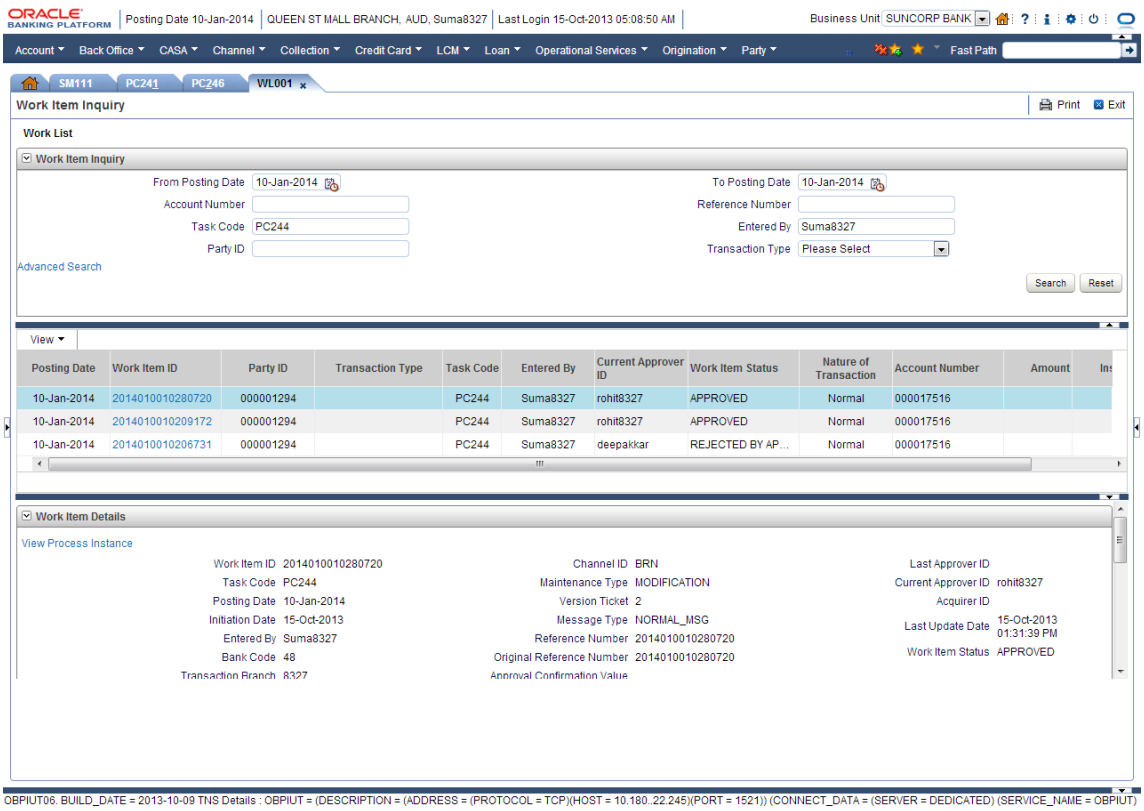


Figure 2–20 Approve or Reject Work item



5. The initiator user can see the status of the work item from the **Work Item Inquiry (Fast path: WL001)** screen.

Figure 2–21 Viewing Status of Work item



2.2 Discretionary Credit Assessment (DCA)

This section explains steps in Discretionary Credit Decision approvals.

Overview

In Origination workflow at Credit Assessment stage, user submits the submission to credit decision engine. If the credit decision status of the submission is Declined or Referred, then submission is moved to manual decision step. The user attached to Enterprise role having rights on Manual Decision Task can assess the credit decision engine recommendation, make the necessary changes and recommend for approval. The system checks whether the role to which user is attached has required authority to approve the task based on values of parameters in the task. If yes, the task is auto authorized. If the values are not within the limits attached to the role, the system checks the approval matrix to find the roles having the required authority to approve the transaction and route task to such roles for approval.

Facts Required for MCD Approvals

Following are the facts required for MCD approvals.

Table 2–3 Facts Required for the MCD Approvals

State
Business Unit
Currency
Personal Aggregate Application Amount
Overall Aggregate Application Amount

Table 2–3 (Cont.) Facts Required for the MCD Approvals

Applicant Credit Card Limit
Aggregate Credit Card Limit
LTV Category Group 1
LTV Category Group 2
LTV Category Group 3
LTV Category Group 4
LTV Category Group 5
Overview Amount
Overview Percentage

These are the facts required to maintain the rules for MCD in OES and SOA. The preconditions are:

- OES configurations
- Group creation in SOA
- Rules configuration in SOA

  Scenario: Submission with Referred status and breaching OES configurations, hence transaction sent for approvals in SOA.

2.2.1 Setup Details

This section discusses the setup details required to configure the DCA services.

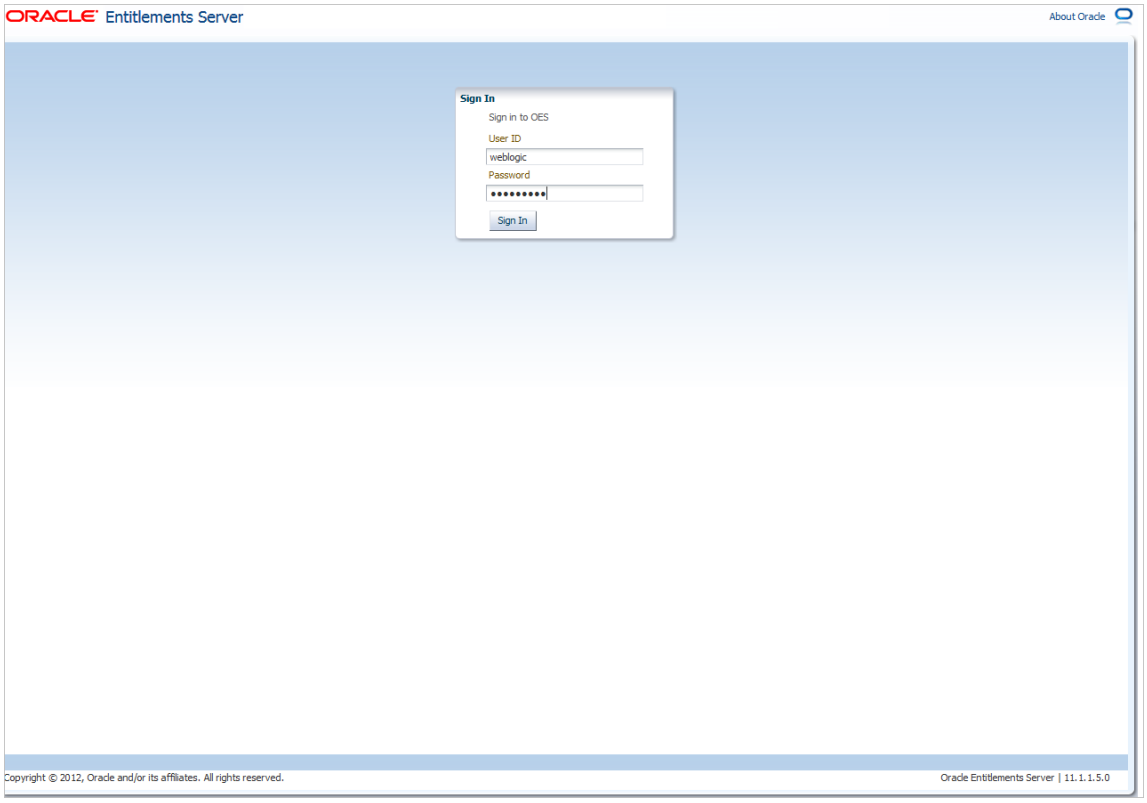
2.2.1.1 APM Policy Setup

To initiate, the user needs to setup policies at APM for auto approval. This policy specifies which transactions with what data will be auto approved. When the data that is sent from the application matches the policy setup, the transaction is auto approved and if the policy is not matched, then the transaction goes for approval.

Following is the procedure to be followed during APM policy setup:

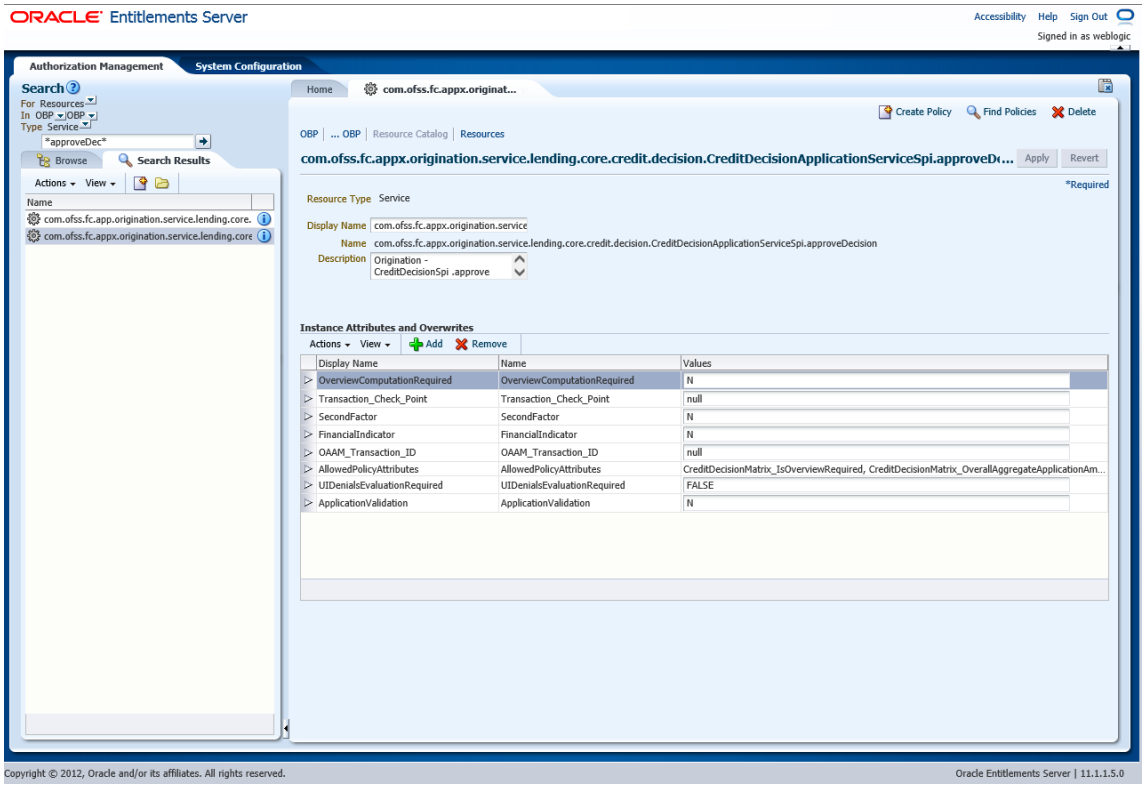
1. Log in to the APM as a user with administrator privileges:

Figure 2–22 Log in to APM



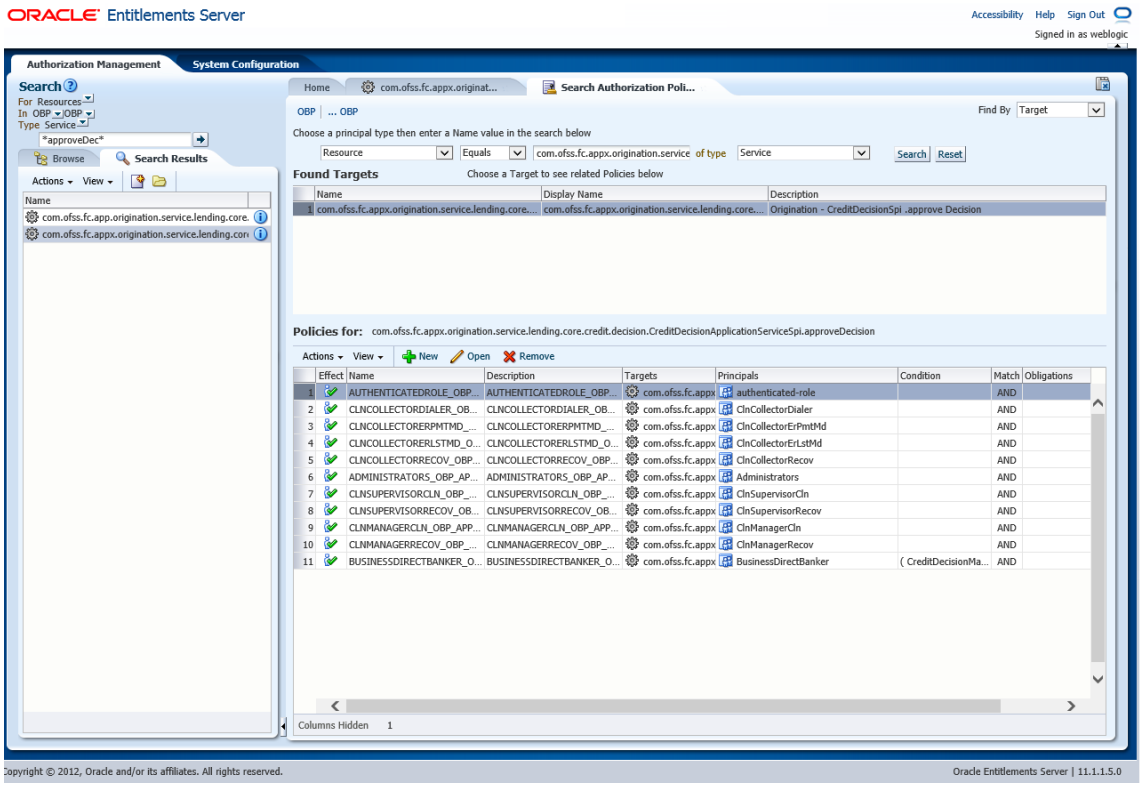
2. Search for the service and click **Find Policies.**

Figure 2–23 Find Policies



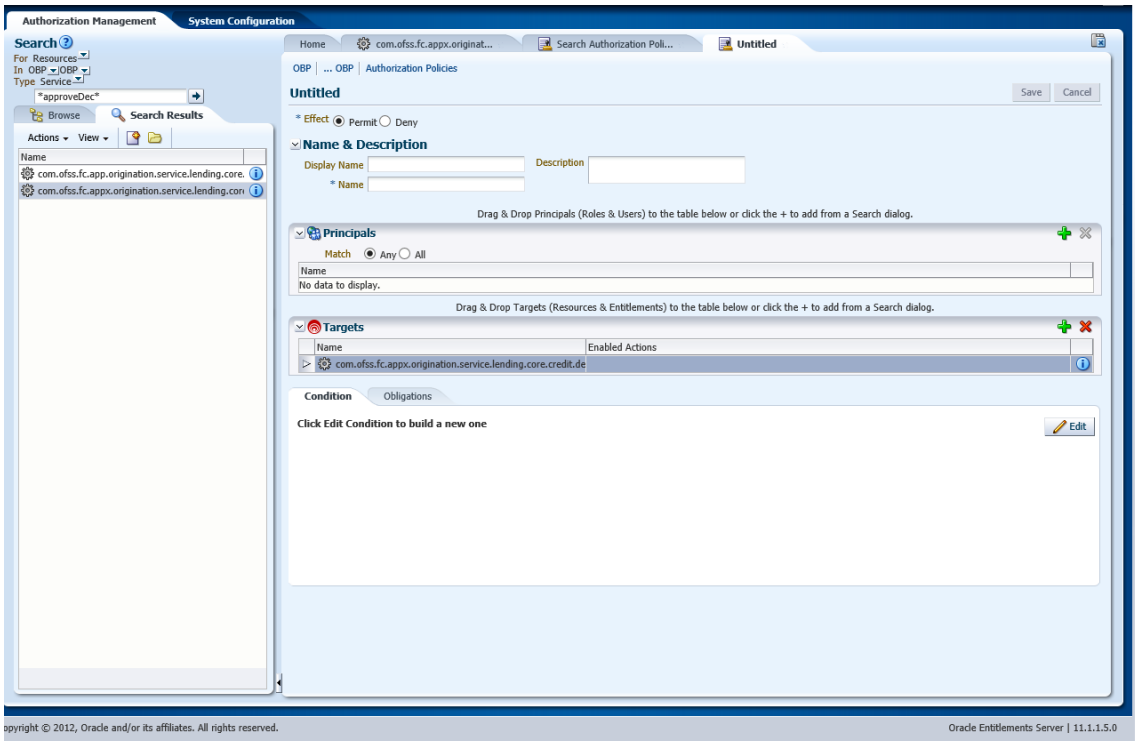
Find Policies opens the below screen.

Figure 2–24 Search Authorization Policies



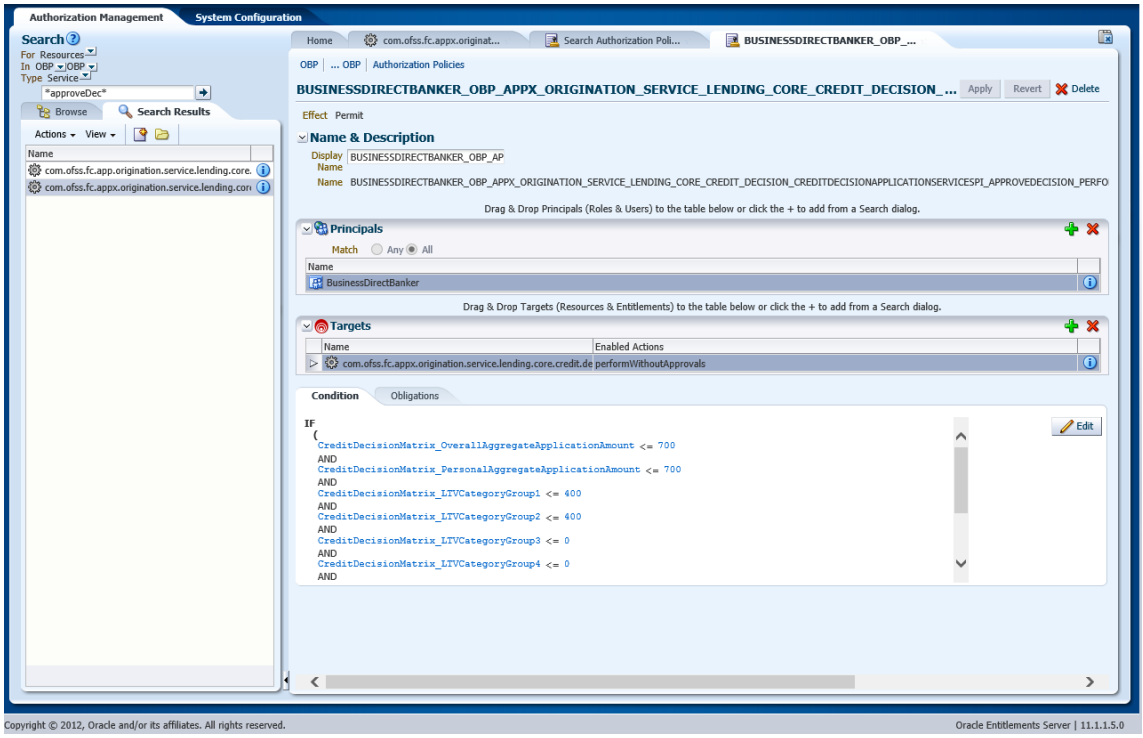
3. Click **New** and add new policy as shown in Figure 2–25.

Figure 2–25 Adding New Policy



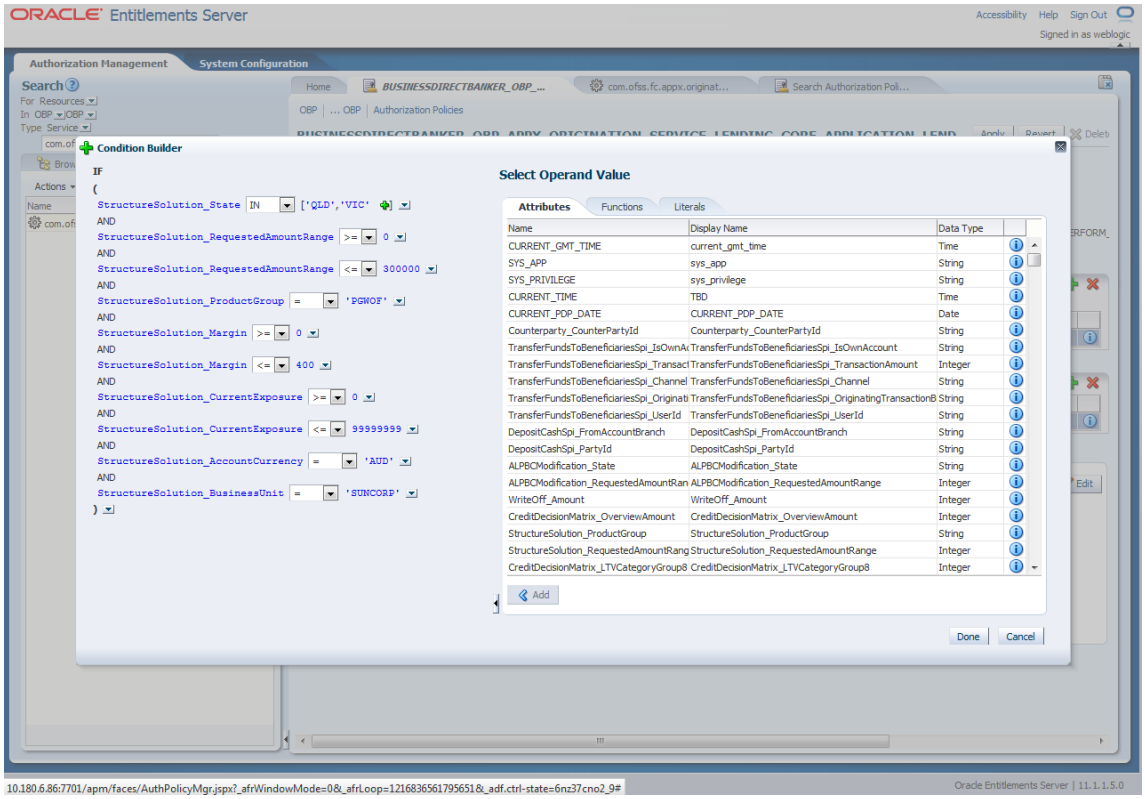
4. Add the following details:
   - **Name and Description**
   - **Principals**
   - **Enabled Action** as *Perform Without approval*

Figure 2–26 Updating Name, Principals and Enable Action Details



5. Click **Done.**
6. Click **Save** to save the policy.

Figure 2–27 Saving Details



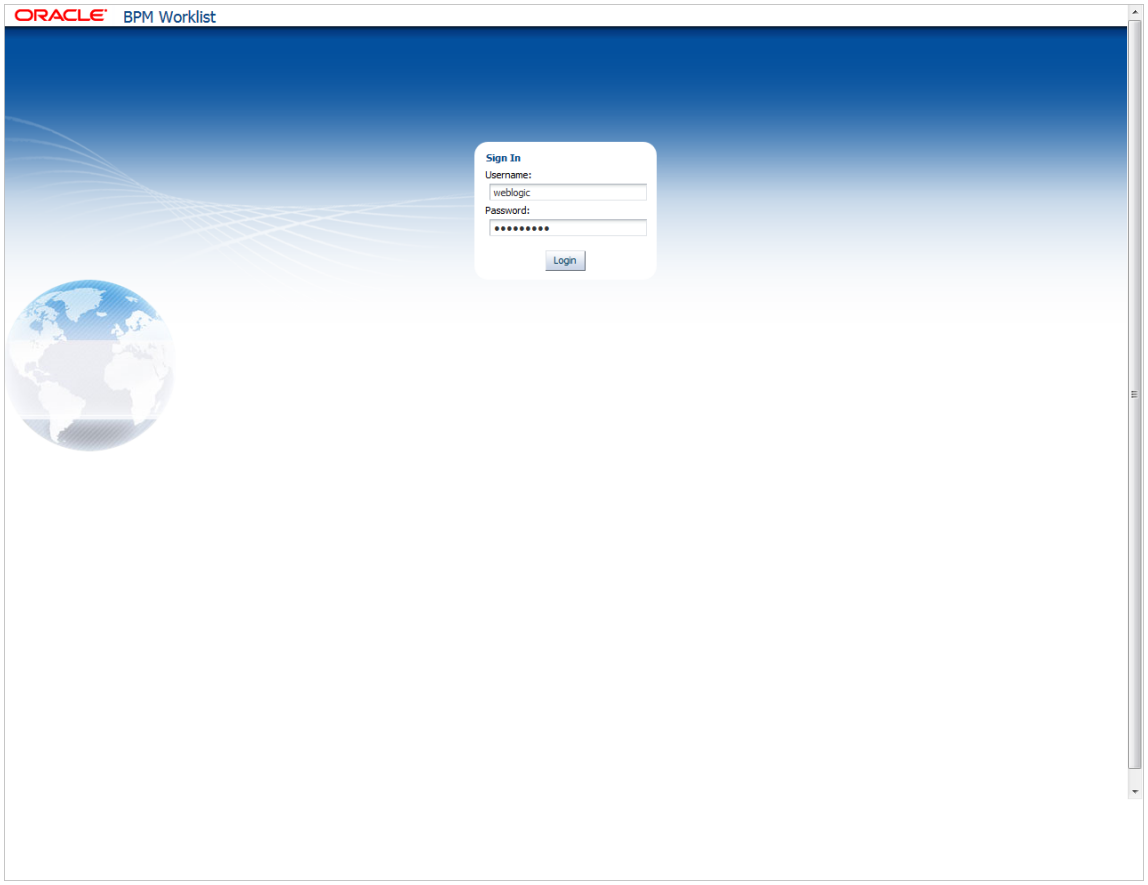
2.2.1.2 BPM Worklist Rules Setup

After the rules are set up in APM, the user needs to set up approval rules in BPM worklist. These rules should cover all scenarios in which the transaction can come out without being auto approved at the APM level. Any transaction which does not trigger the rules at BPM level will be auto rejected.

Following is the procedure to be followed during BPM worklist rules setup:

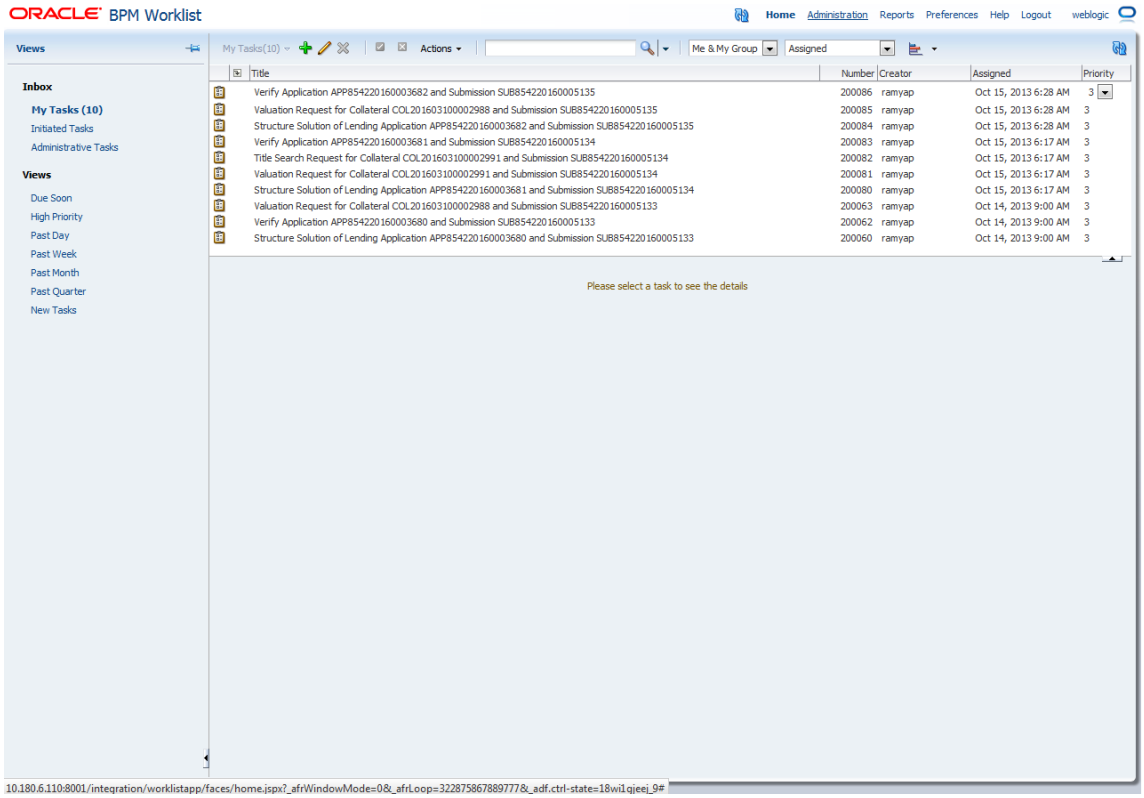
1. Log in to the BPM worklist application as a user with administrative privileges.

Figure 2–28 Log in to BPM Worklist



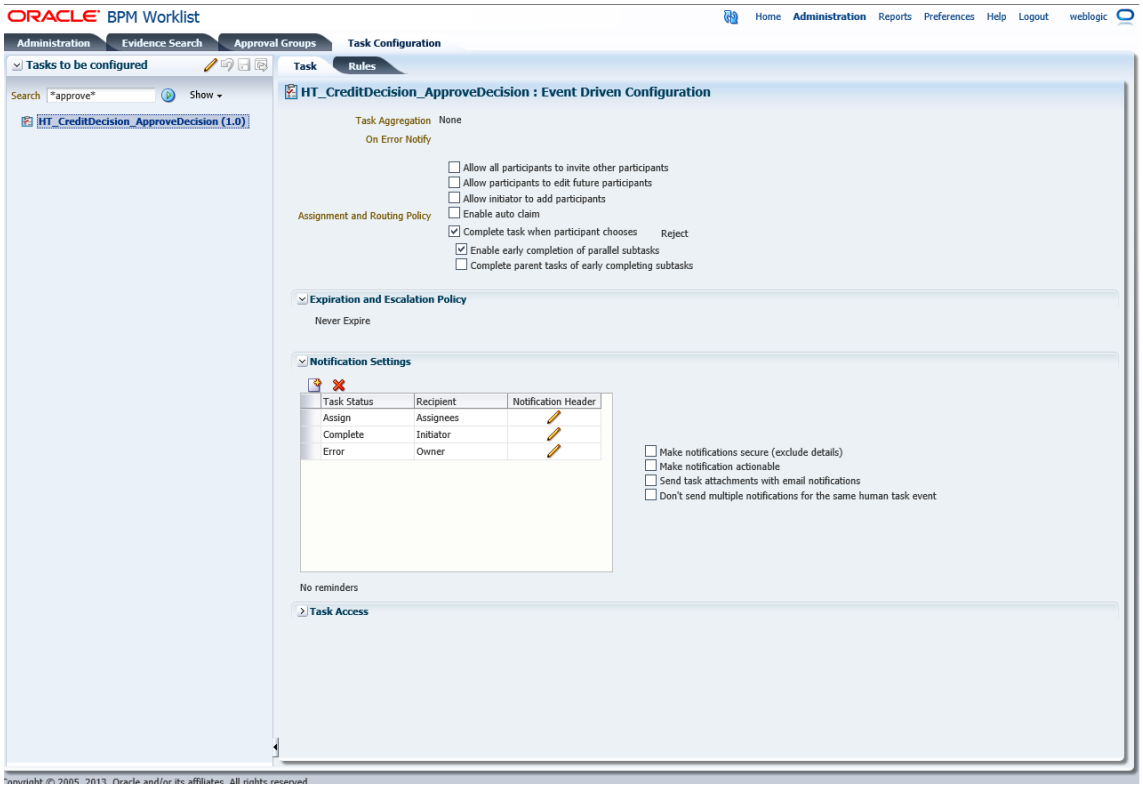
2. Click the **Administration** link in the top left corner.

Figure 2–29 Administration Link



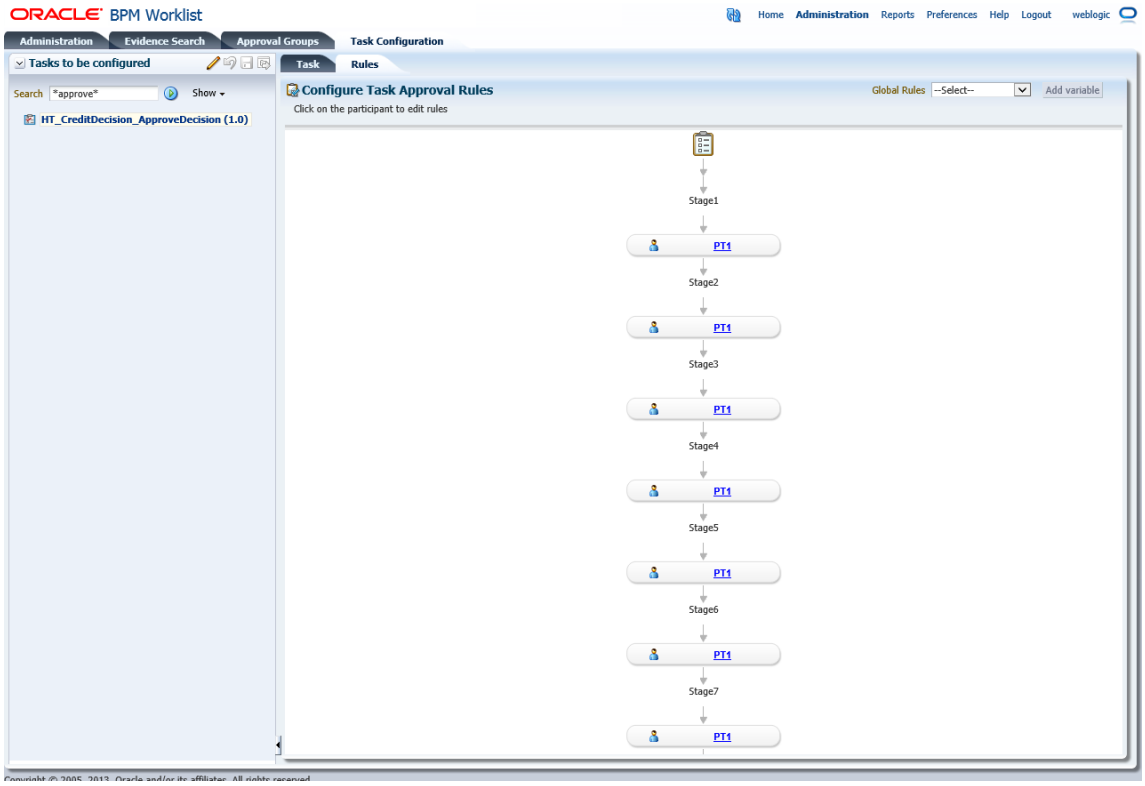
3. Open the **Task Configuration** tab and search for the specific process.

Figure 2–30 Task Configuration



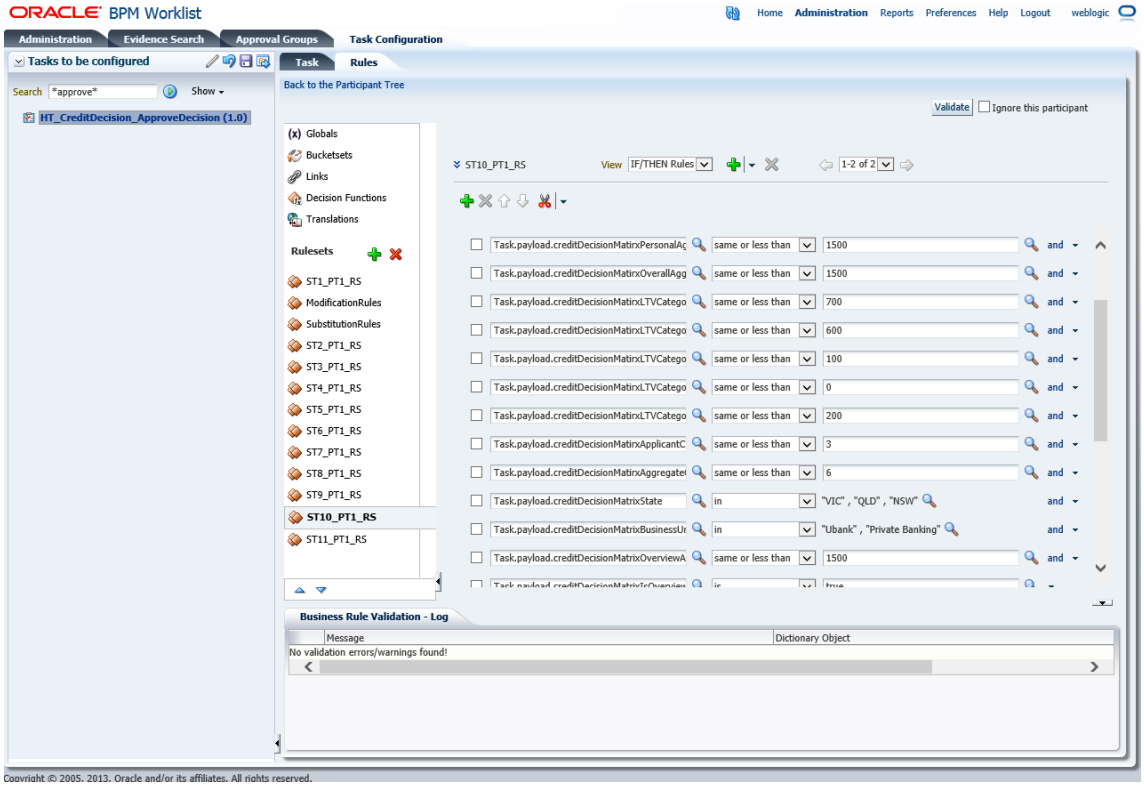
4. Open the **Rules** tab and click the links (PT1, PT2, and so on) provided for any stage of your choice.

Figure 2–31 Rules Link



5. Click the **Edit** (Pencil) icon for creating the rule and the following will be visible.

Figure 2–32 Creating Rule



6. Add rule attributes and commit the rule.

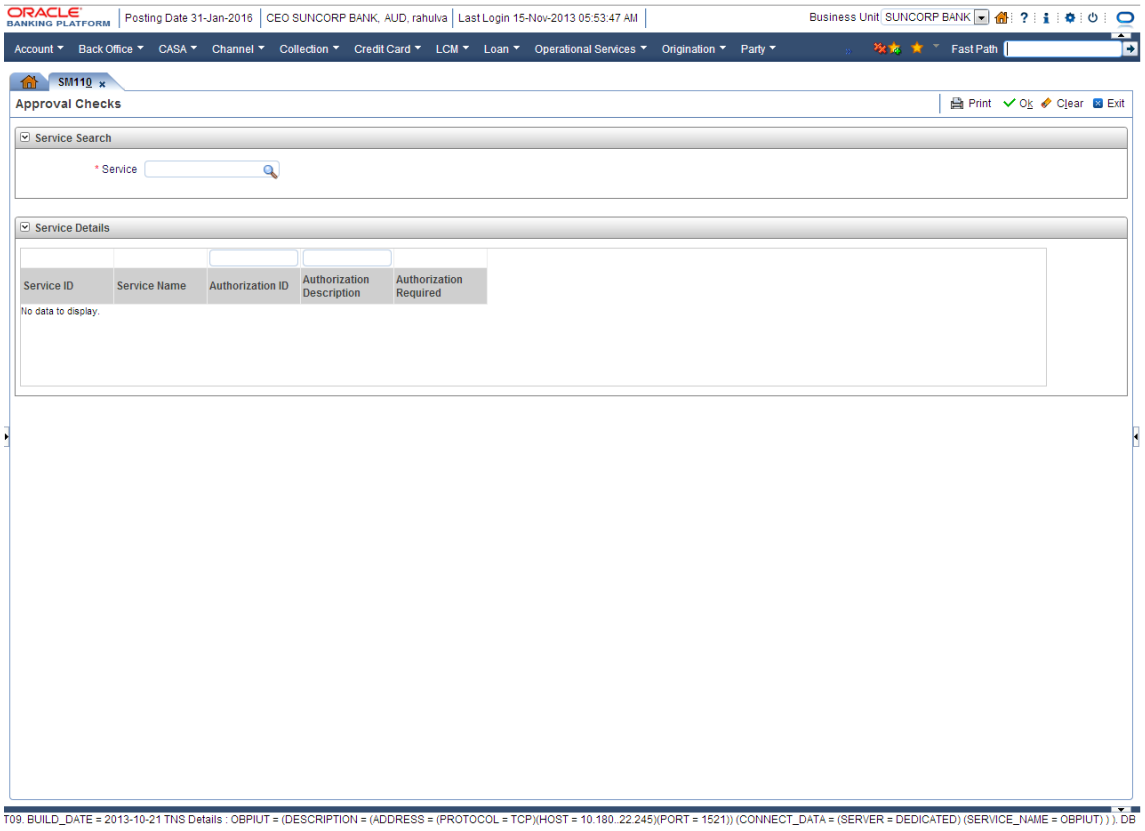
2.2.1.3 SMS Setup

The user also needs to enable matrix-based authorization for the DCA service. This can be done through the **Approval Checks (Fast path: SM110)** page which is used for enabling approval.

Following is the procedure to be followed during SMS setup:

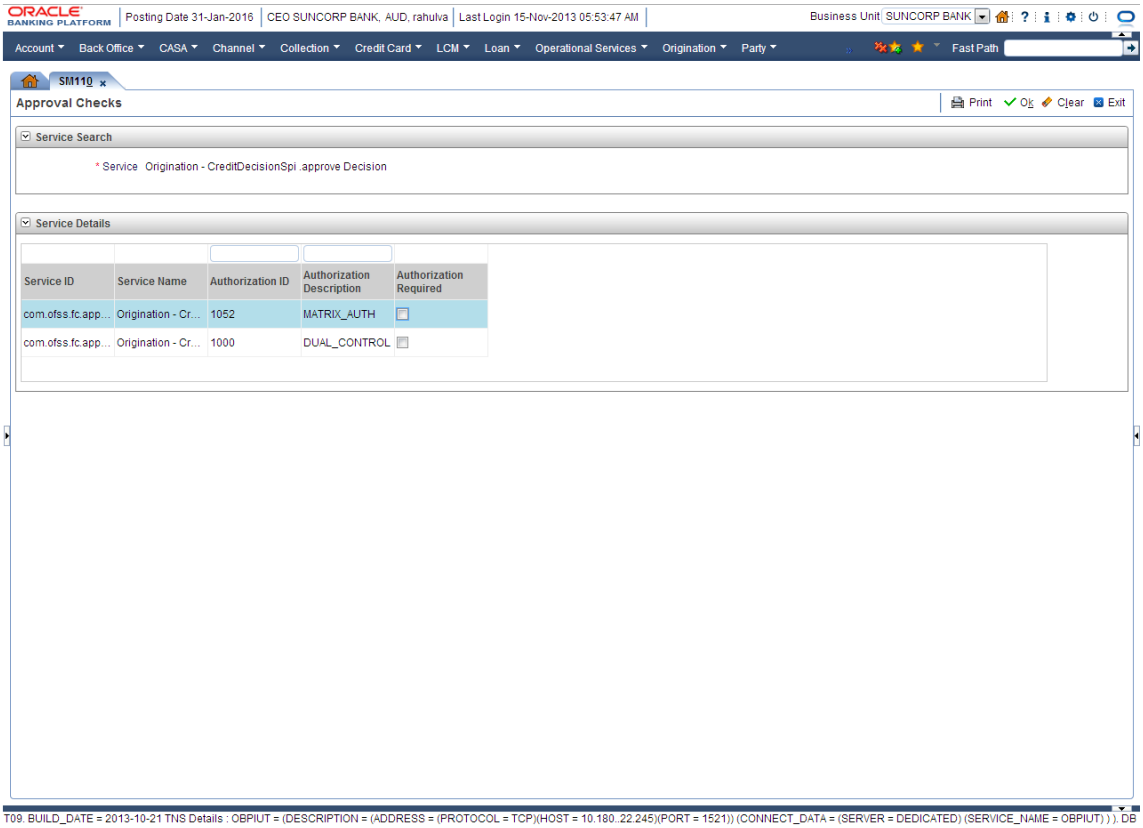
1. Log in to the application as admin user and open the **Approval Checks (Fast path: SM110)** page.

Figure 2–33 Approval Checks



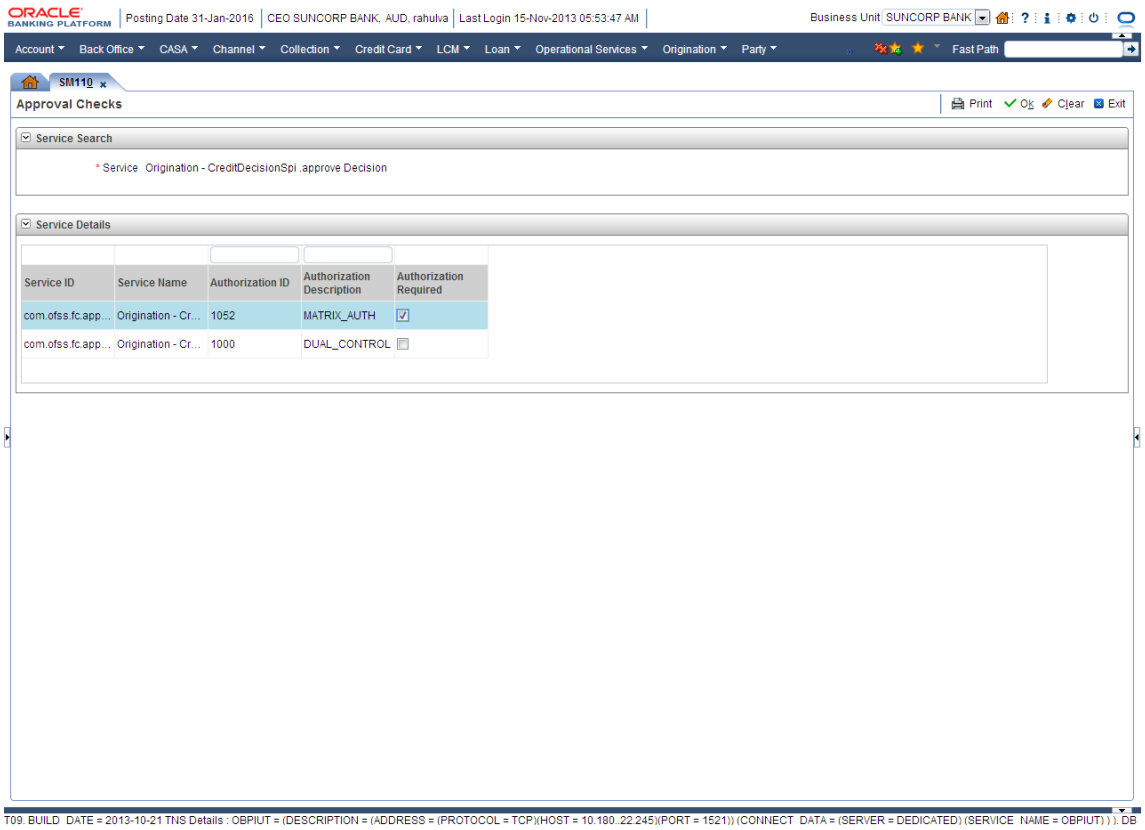
2. Enter the service ID in **Service** field and search.

Figure 2–34 Service ID



3. Check the matrix_auth check box and click **Ok** to save the changes.

Figure 2–35 Checking Matrix Auth

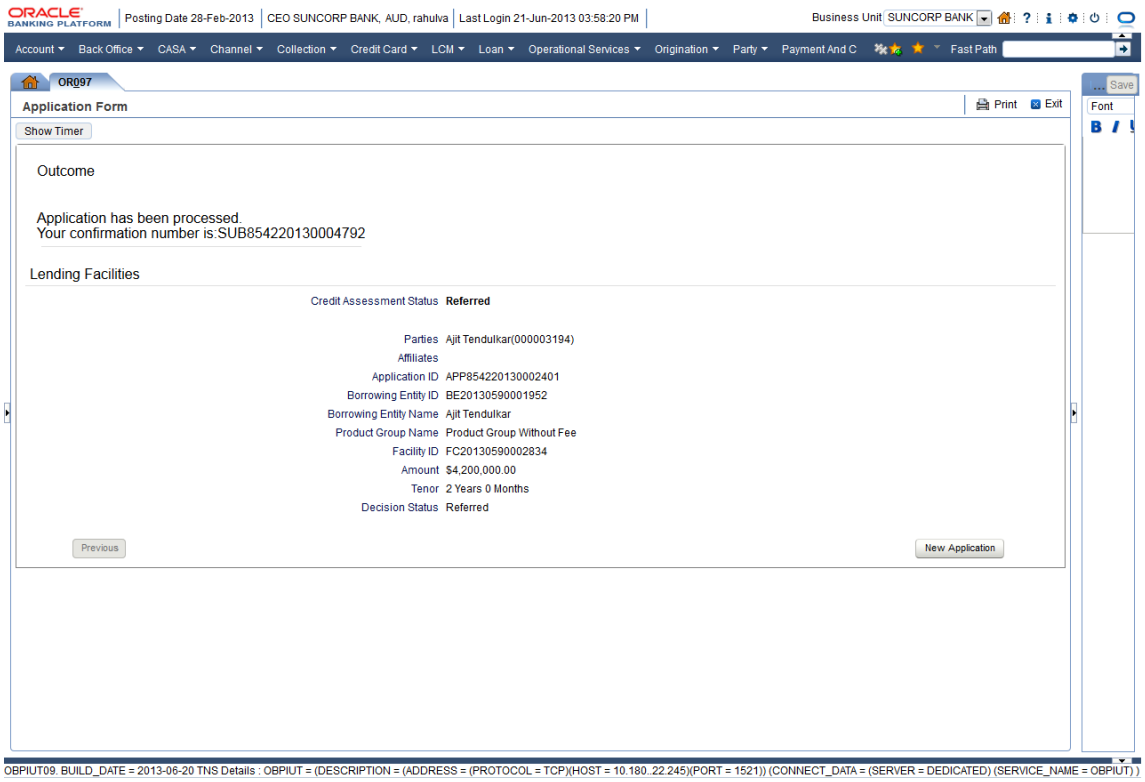


2.2.2 Performing the Transaction

Following procedure explains how to perform a transaction:

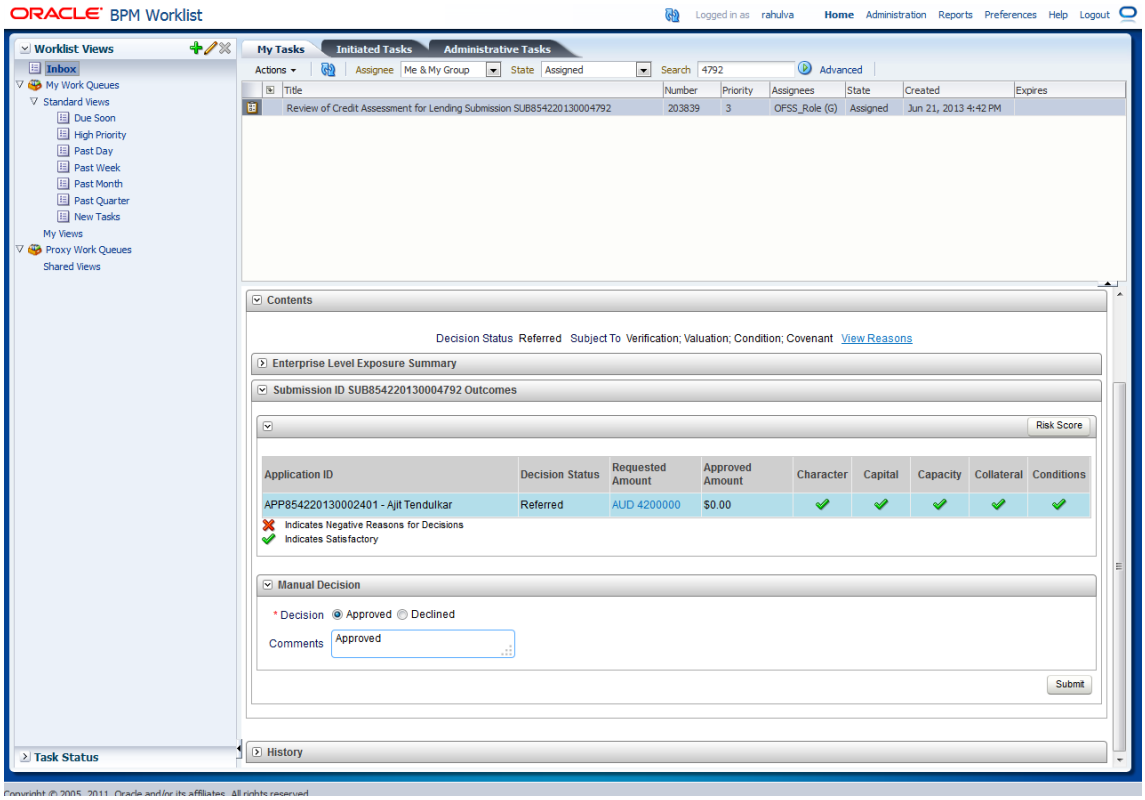
1. Log in to the application as teller user and initiate an origination task flow such that the application goes to referred status.

Figure 2–36 Initiating Origination Task Flow



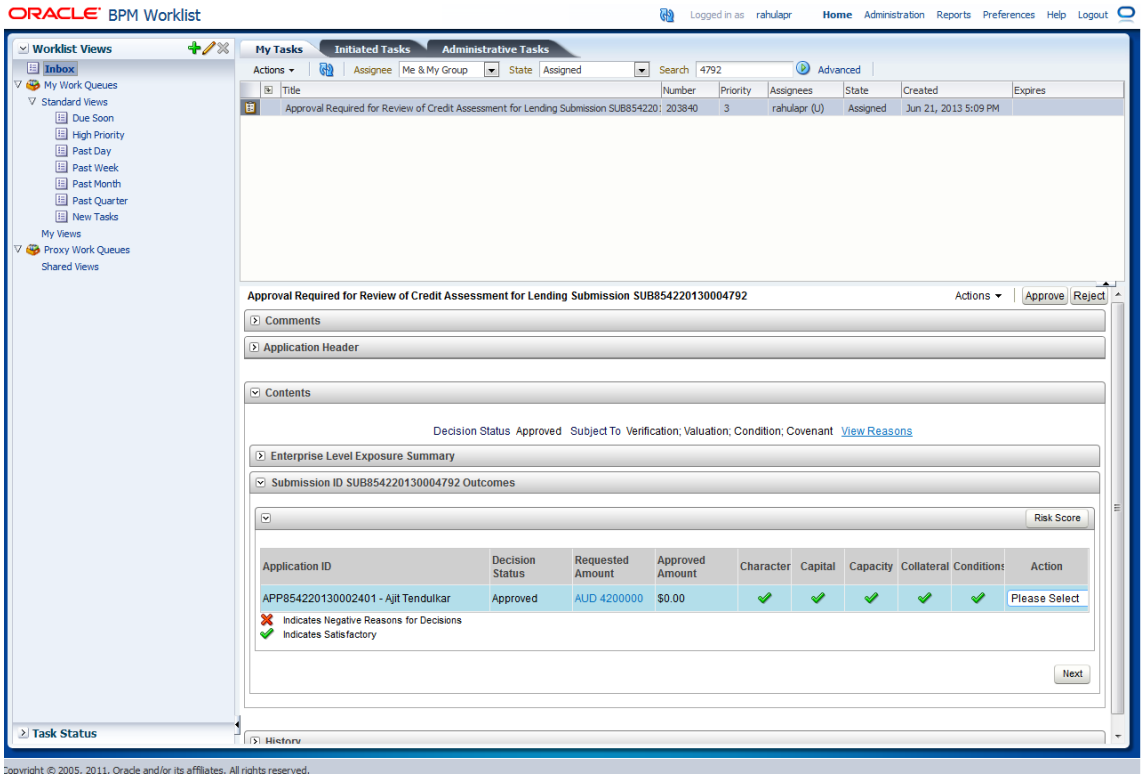
2. Log in to BPM worklist as initiator user who initiated the origination process.
3. Select to approve the credit assessment task created for the him.

Figure 2–37 Approving Credit Assessment



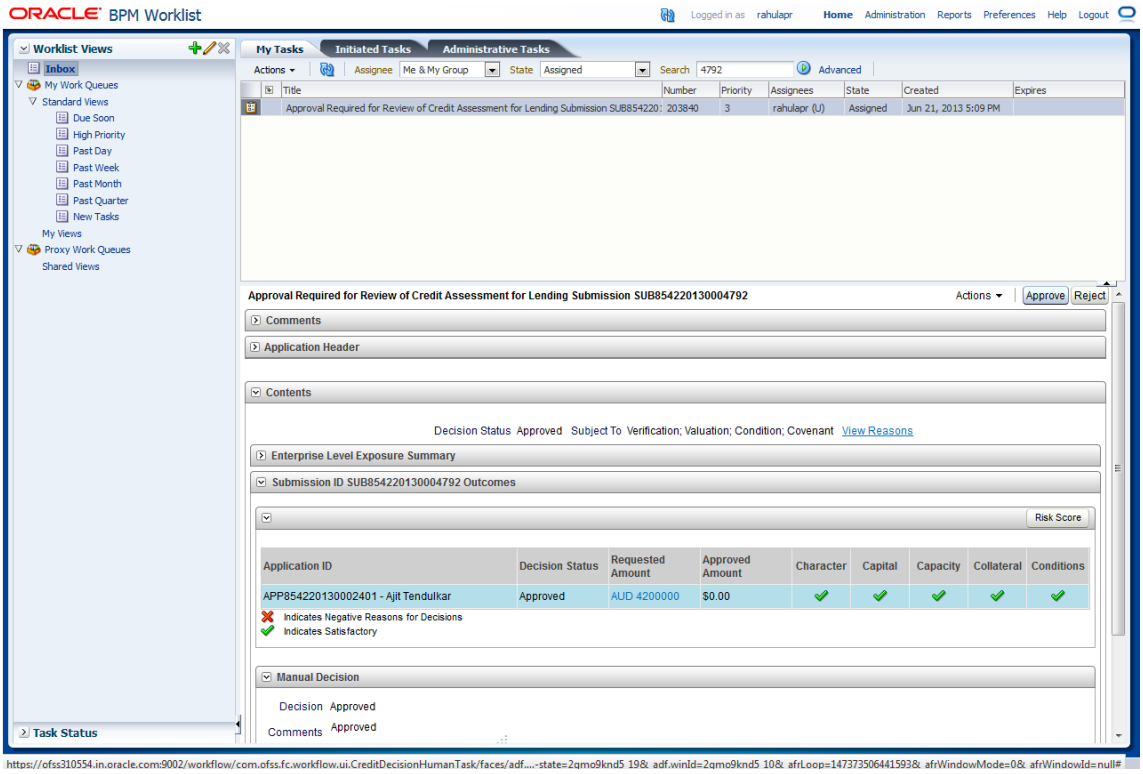
4. Credit assessment task goes to the approver user for approval. Log in as approver user to BPM Worklist

Figure 2–38 Credit Assessment Approval



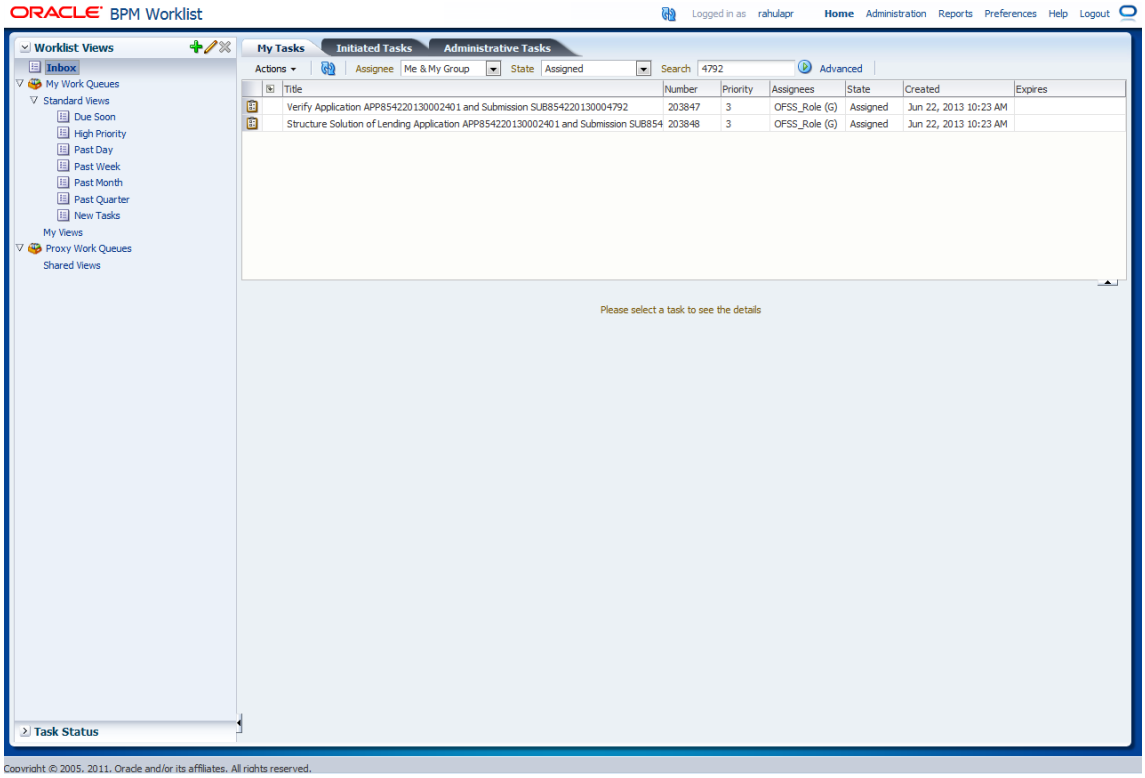
5. User can either approve or reject the work item (approved here) after adding comments.

Figure 2–39 Adding Comments



6. Once approved, the further task flows required for origination will be created for the initiator user.

Figure 2–40 Creation of Further Task Flows



2.3 Enabling Worklist Authorization

This section explains the steps in enabling Worklist authorization. Following are the steps:

Step 1 Identify the Service Name

There can be two services for the same service. A normal 'app' service and 'appx' service. We need to identify both the Services.

For example, Add Transaction Limit will have the following two services:

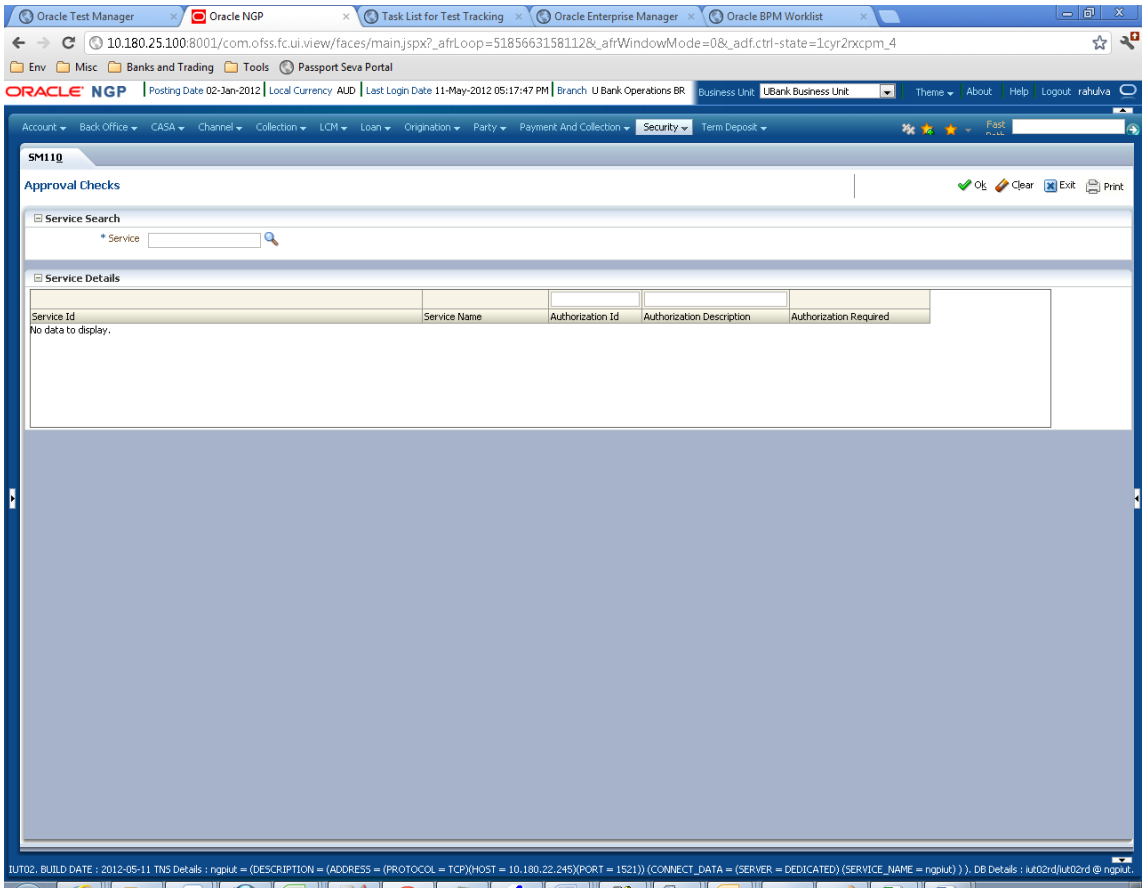
- com.ofss.fc.app.sms.service.limit.TransactionLimitApplicationService.addTransaction Limit
- com.ofss.fc.appx.sms.service.limit.TransactionLimitApplicationServiceSpi.addTransac tionLimit

Step 2 Enable Dual Authorization

Once the services are identified, follow the below steps to enable Dual Authorization

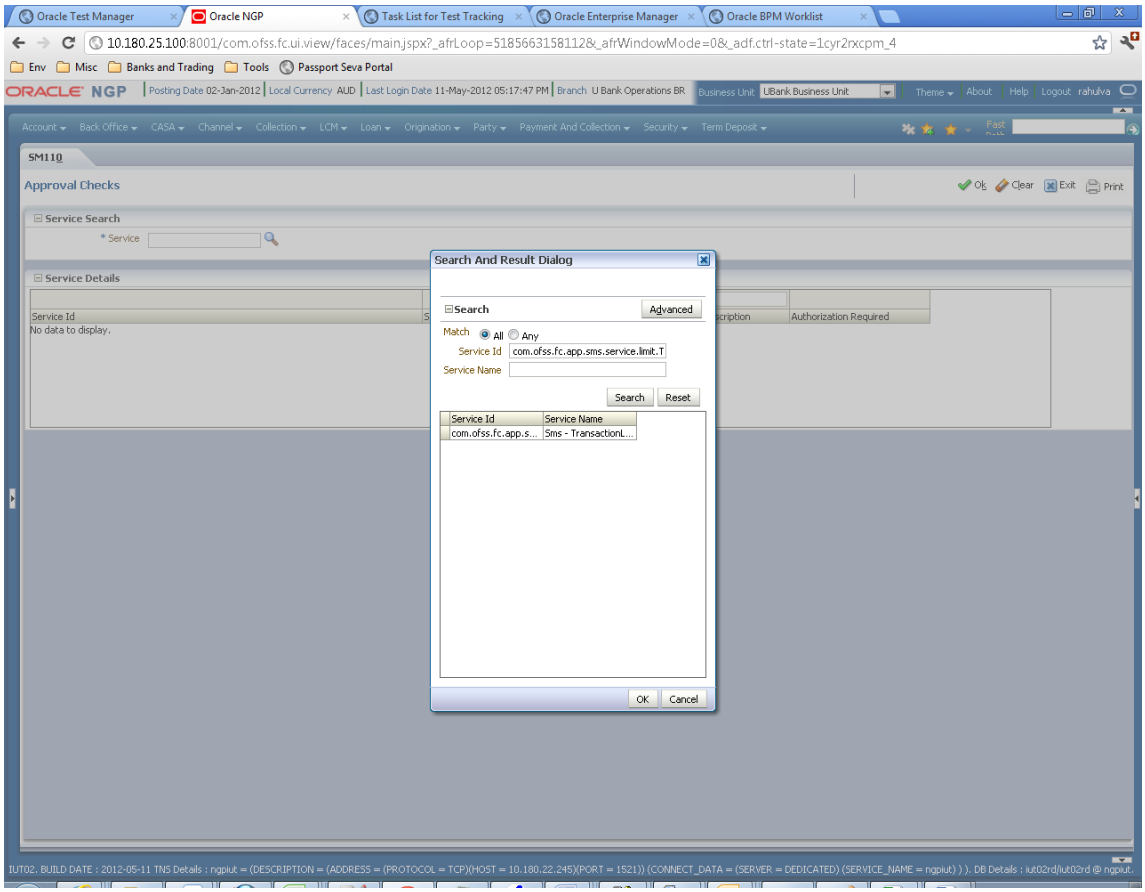
1. Log in to Oracle Banking Platform.
2. Navigate to the page **Approval Checks (Fast path: SM110).**

Figure 2–41 Approval Checks (Fast path: SM110)



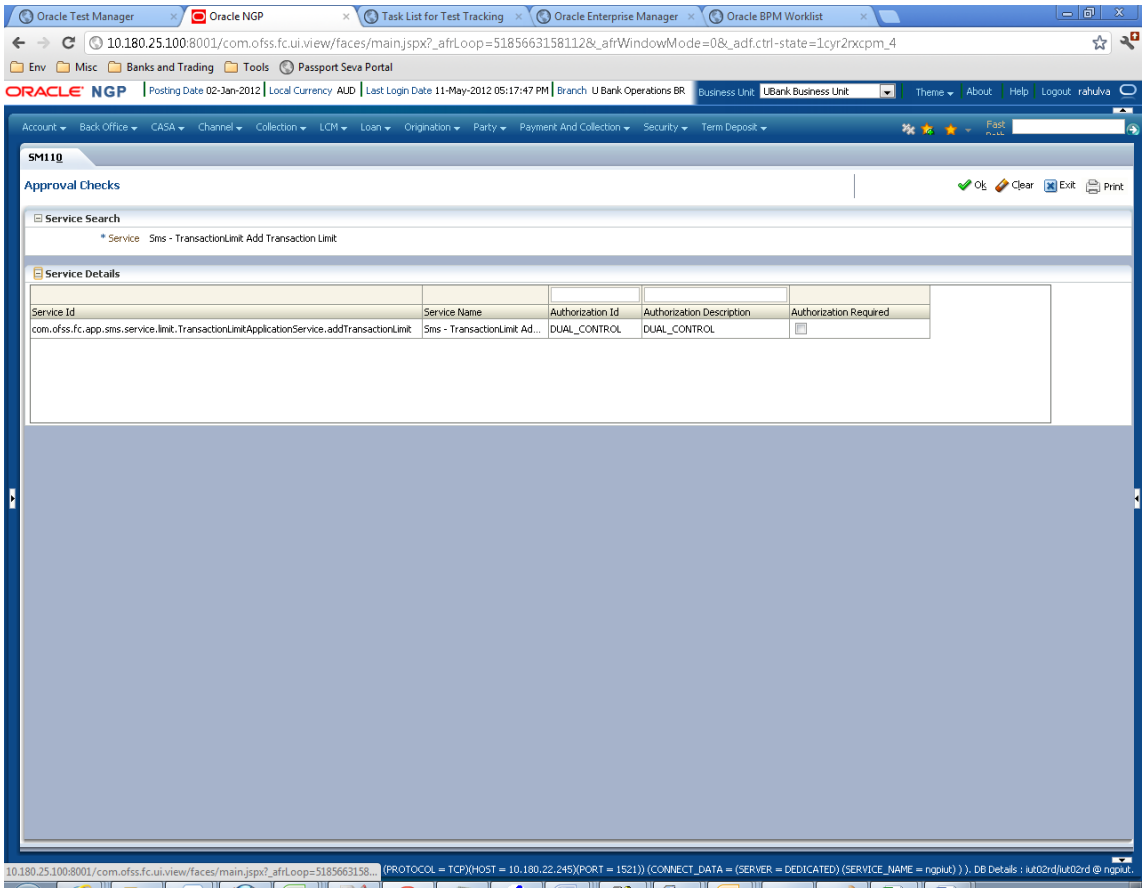
3. Search for the required Service as shown in Figure 2–42.

Figure 2–42 Search Requisite Service



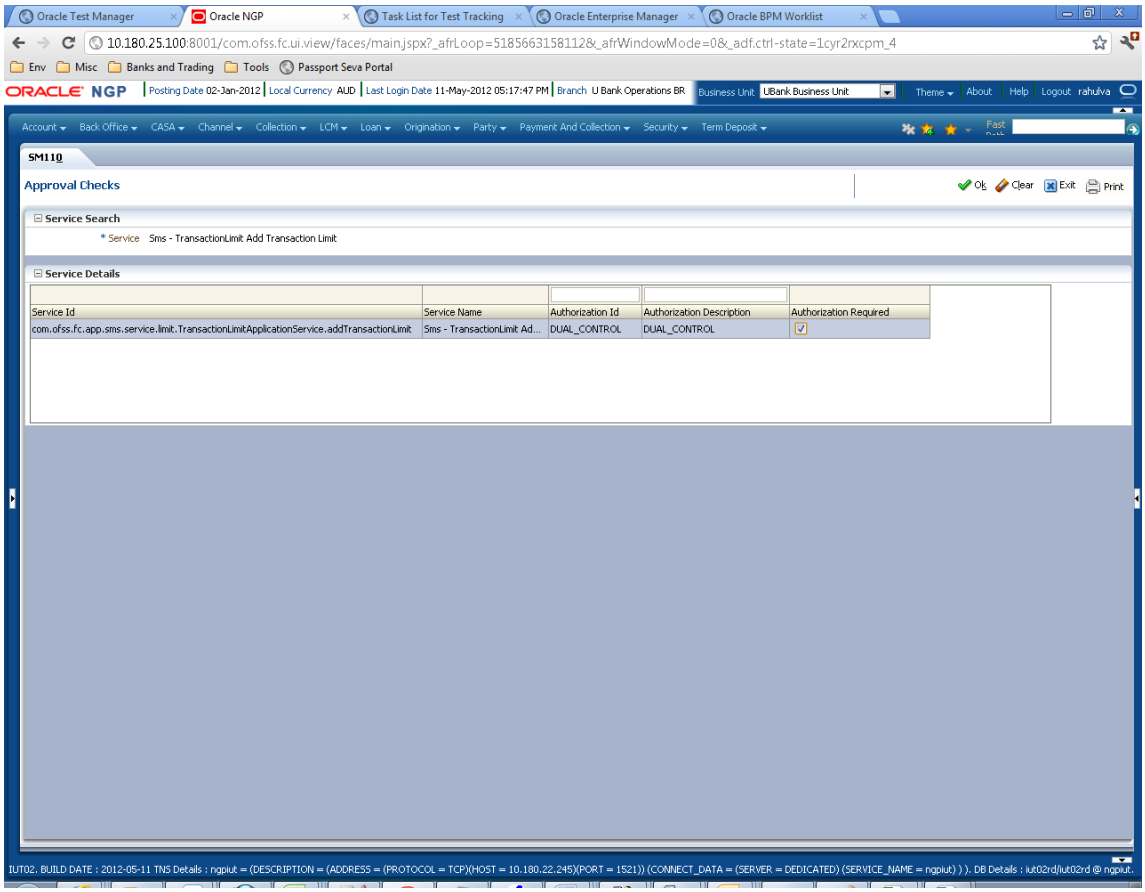
4. On successful search, the required Service Details appear as shown in Figure 2–43.

Figure 2–43 Service Details



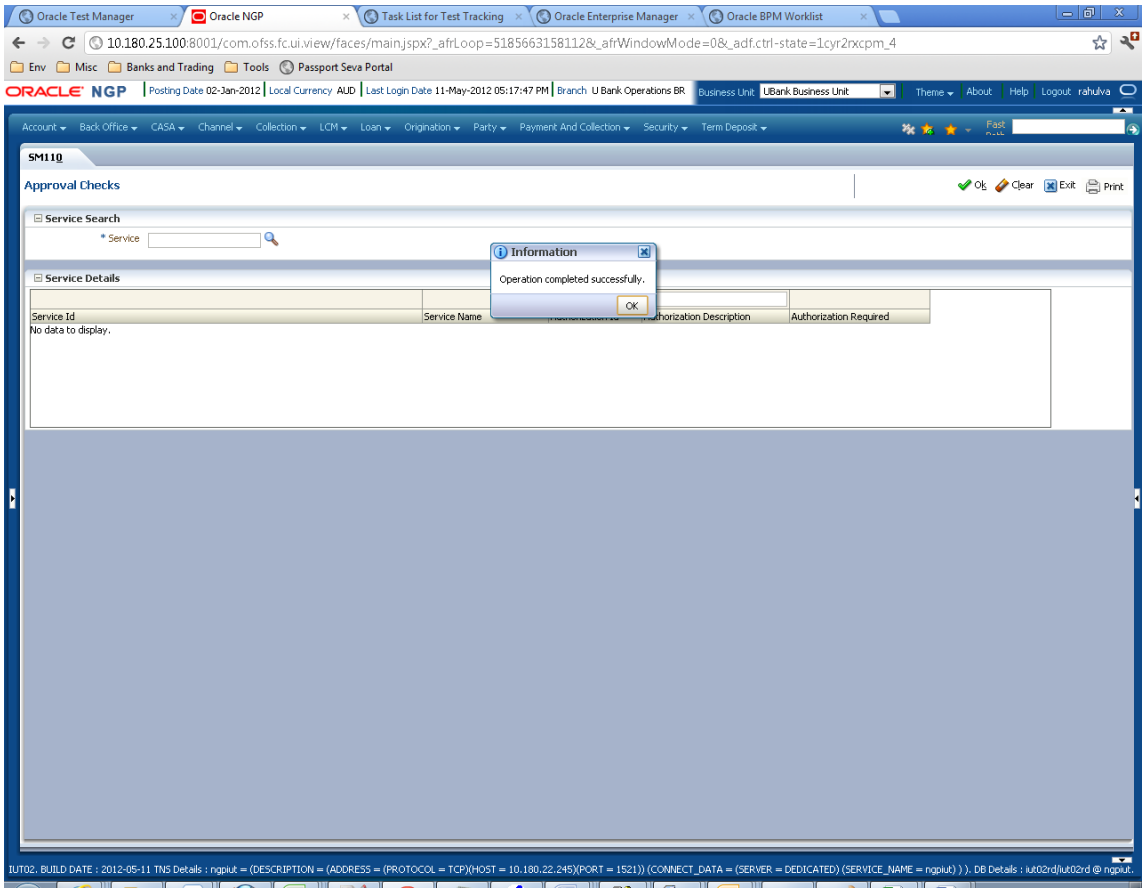
5. Select the **Authorization Required** check box.
6. Click **Ok** to complete the action.

Figure 2–44 Authorization Required Check box



The message appears that *Operation completed successfully.*

Figure 2–45 Approval Checks Success



Step 3 Configure Severity

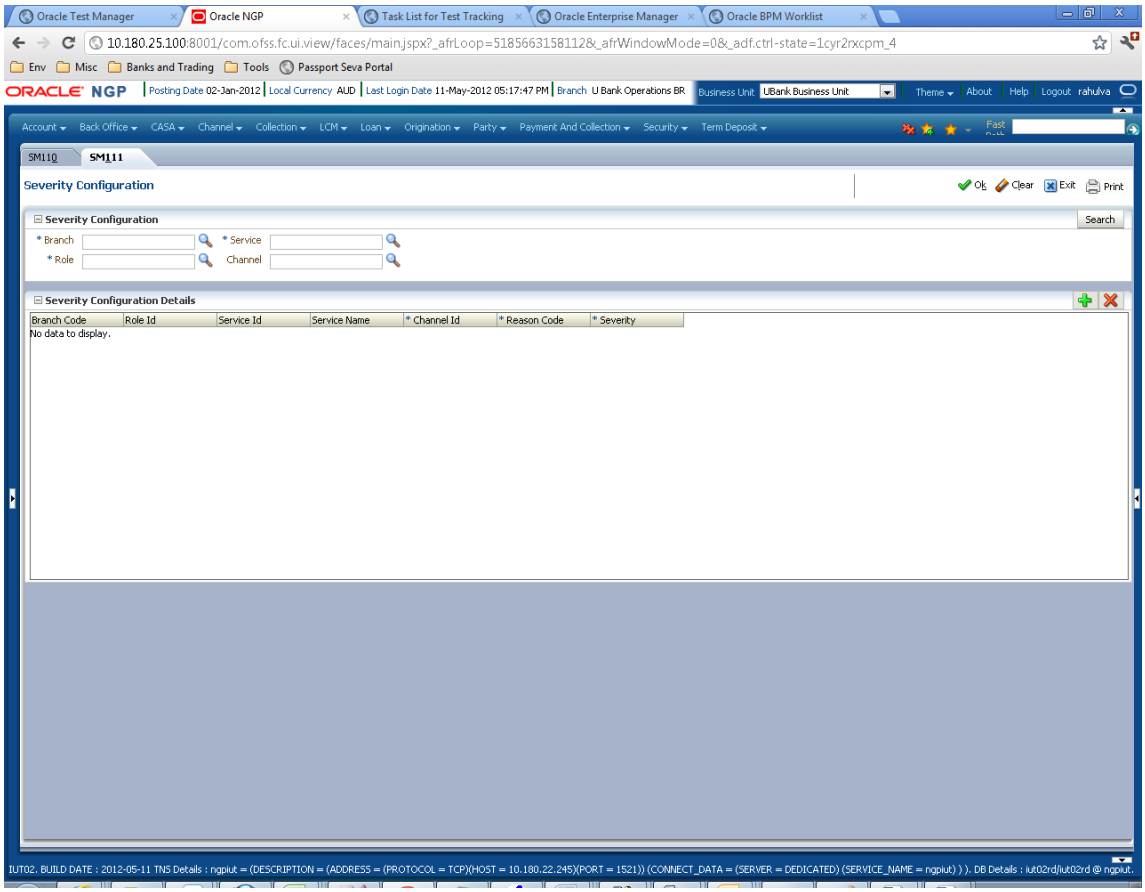
Enabling of dual Authorization alone will not send the transaction for approval. Hence we need to configure Severity for the identified service, to enable the call for approval workflow.

This will be configured from the OBP Application page **Severity Configuration (Fast path: SM111).**

Follow the below steps to configure severity:

1. Log in to Oracle Banking Platform.
2. Navigate to the page **Severity Configuration (Fast path: SM111)** from the Security menu item (Security- - > Severity Configuration).

Figure 2–46 Severity Configuration (Fast path: SM111)

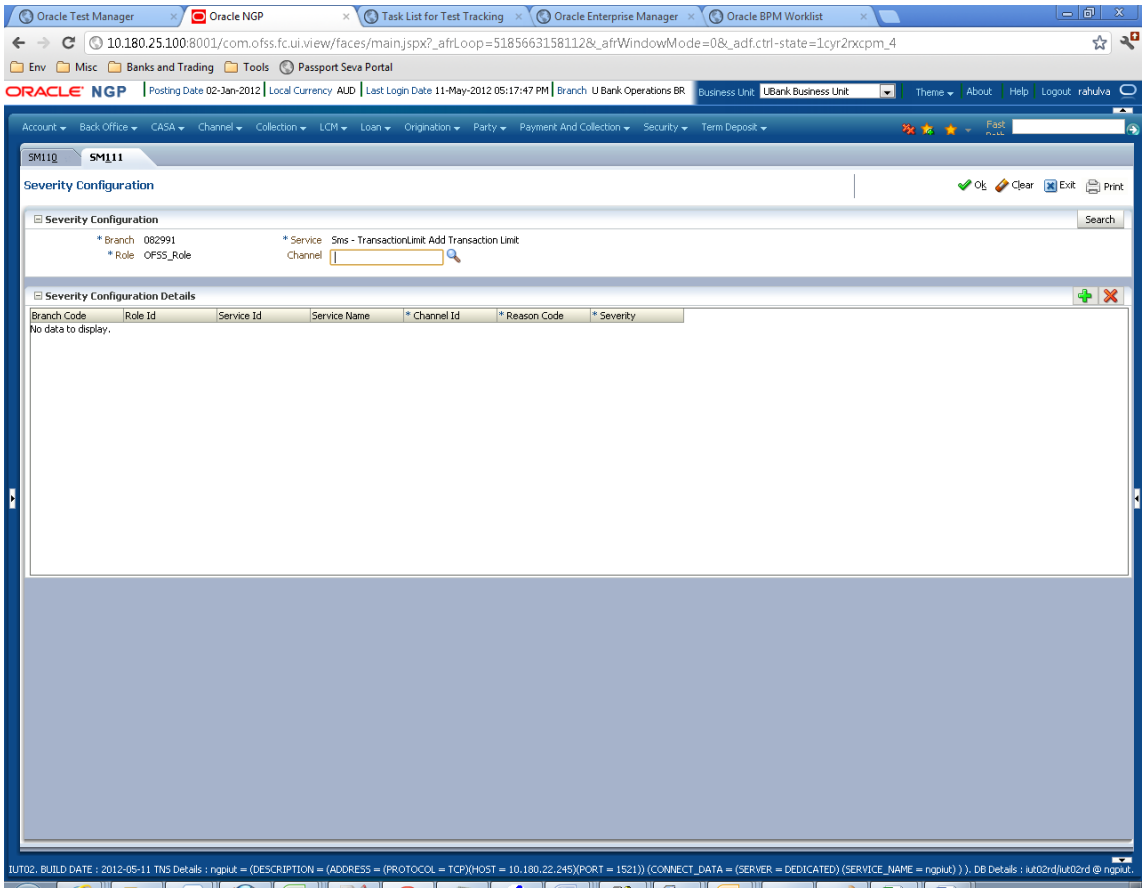


3. Enter the following details in the **Severity Configuration** section.

Branch	Branch code from which the transaction is to be performed. Example: 082991
Role	Security Role to which the user belongs and initiates the transaction.
Service	Enter the Service ID identified. Here, for both APP and APPX services, the configuration for severity needs to be done.
Channel	This is optional. You get to select this option when you proceed to configuration.

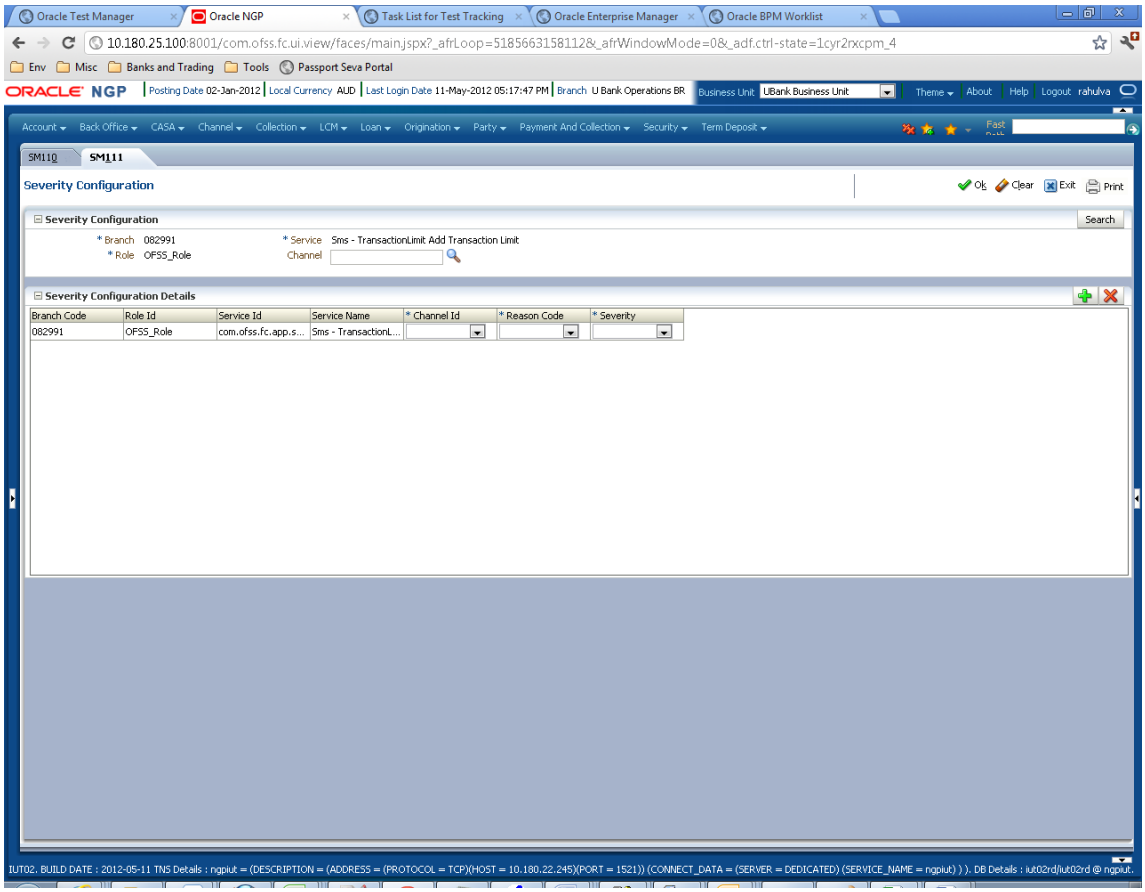
Note: At times you may also notice that the Severity Configuration is already set up. Do not change that.

Figure 2–47 Severity Configuration Details



4. Click **Search** on entering the required data. Existing Severity configuration, if any, appears.
5. In **Severity Configuration Details** section, configure the severity by clicking **Add**, if the desired severity is not configured. This inserts a row as shown in Figure 2–48.

Figure 2–48 Severity Configuration - Search



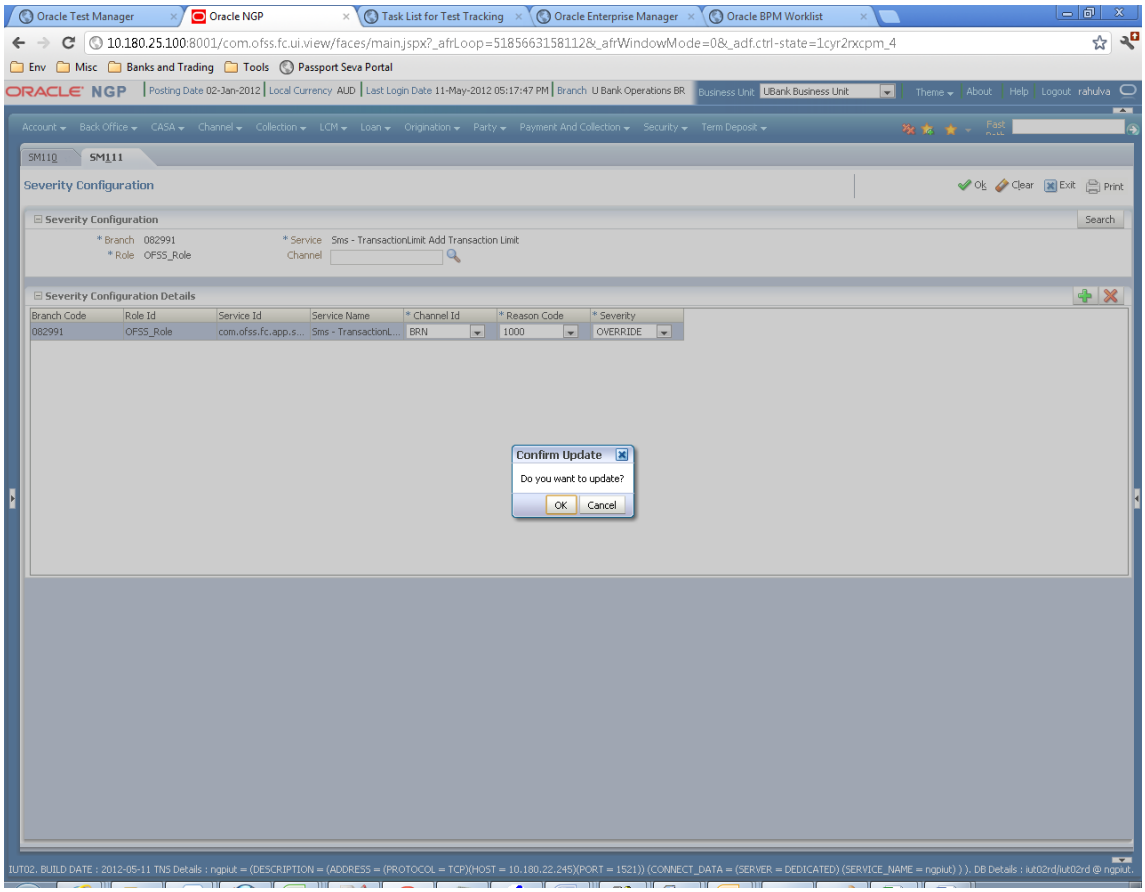
6. Enter the following details in the newly added row in the **Severity Configuration Details** section for configuring the severity:

Channel	Channels such as BRN, ATM, IB and so on, through which the transaction is performed.
Reason Code	Select Reason Code 1000 normal approval flow. See Section 2.4, "Configuring Approvals for Reason Codes Other Than 1000" for more details.
Severity	This field contains four values as detailed below: ■ **Ignore:** Allows transaction to complete without any authorization, that is Auto Authorization. ■ **Override:** Transaction will be sent for Authorization. ■ **Notify:** In this case, the task is not sent for authorization, but the user is expected to confirm the transaction for proceeding ahead. This option is not applicable in case of Dual Authorization. ■ **Reject:** System doe not allow to proceed with transaction.

To enable Dual Authorization, select **Override** option.

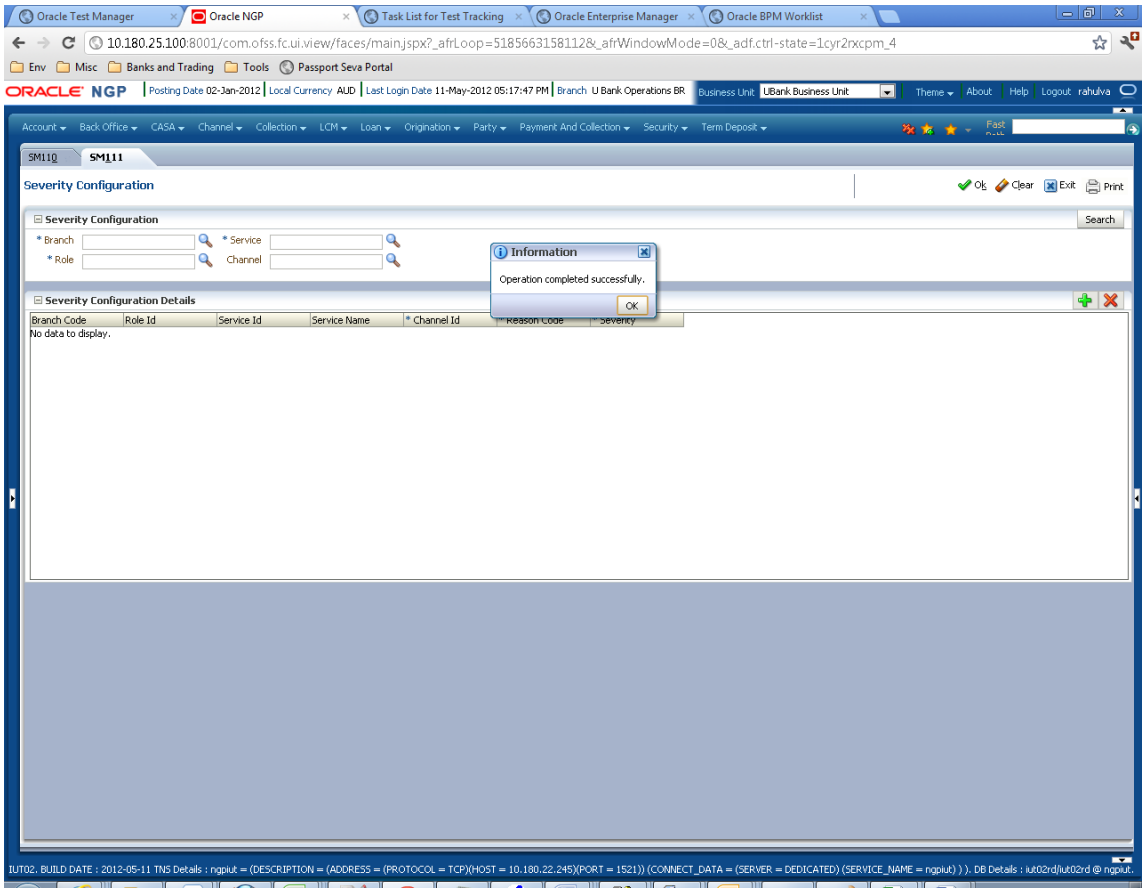
7. Click **Ok** after entering the required data.
8. Next click **Ok** to confirm the update as shown in Figure 2–49.

Figure 2–49 Severity Configuration - Confirm Modification



The message appears that the *Operation completed successfully.*

Figure 2–50 Severity Configuration Success



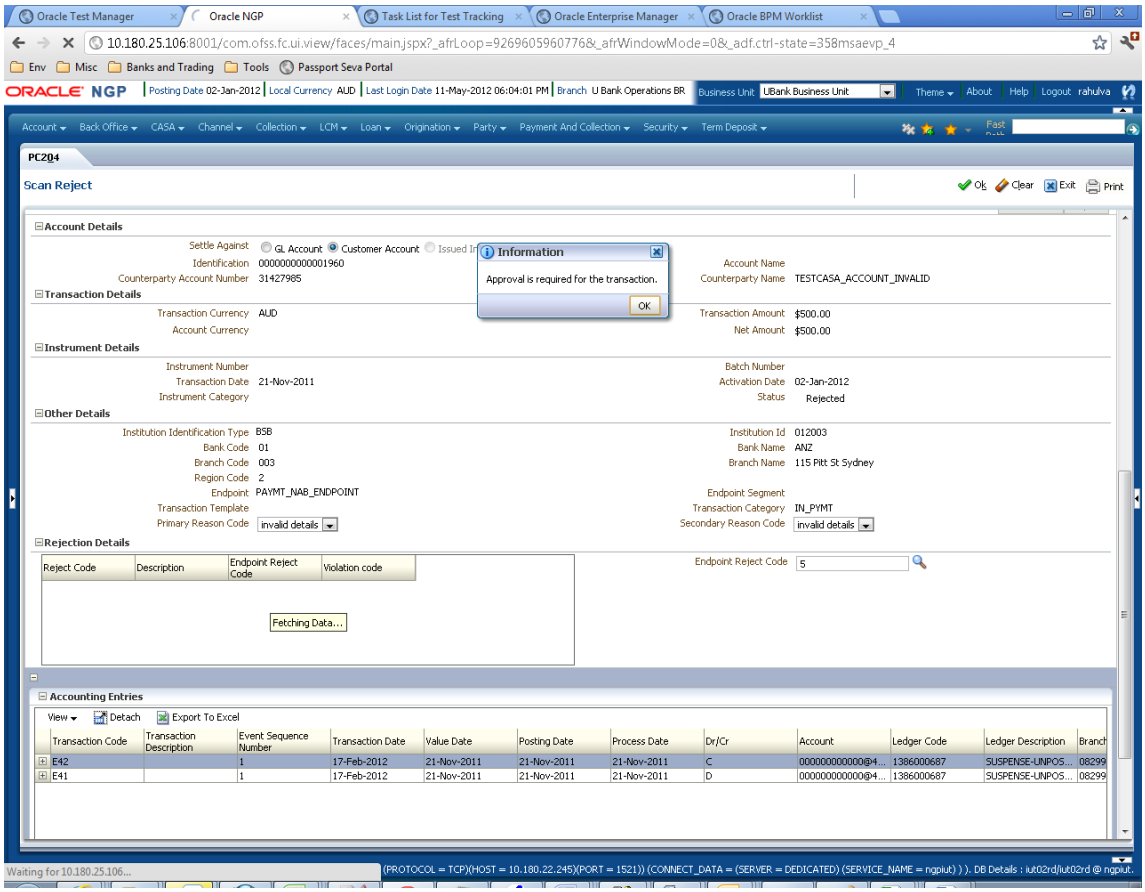
Step 4 Testing Approvals

Once the configurations are done, navigate to the screen for which approval is enabled.

Let us consider **Scan Reject (Fast path: PC204)** for example:

1. Log in to Oracle Banking Platform.
2. Navigate to the page **Scan Reject (Fast path: PC204).**
3. Enter the required details and click **Ok.** The message *Approval is required for the transaction* appears.

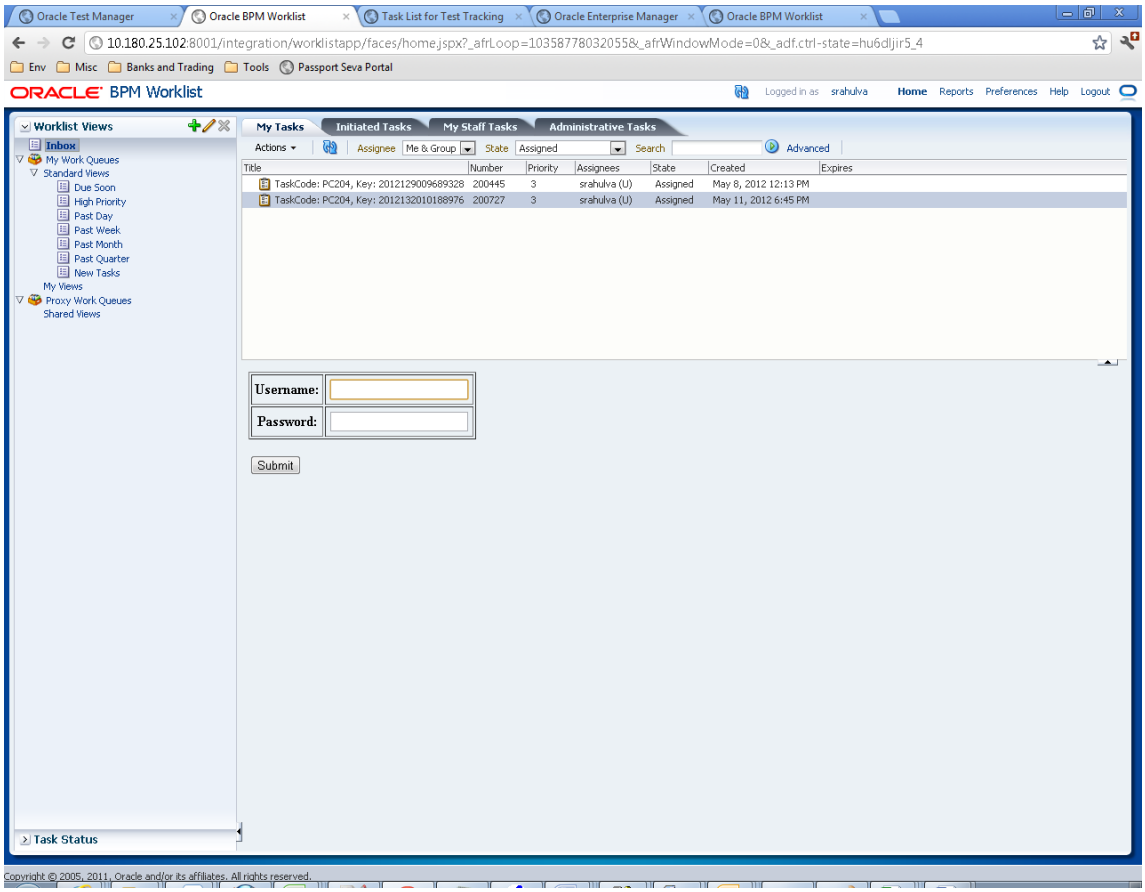
Figure 2–51 Scan Reject (Fast path: PC204)



4. Now log in to the Oracle BPM Worklist application (BPEL Worklist) of the respective environment using the supervisor credential.
5. In **My Tasks**, select the work item assigned against your name.

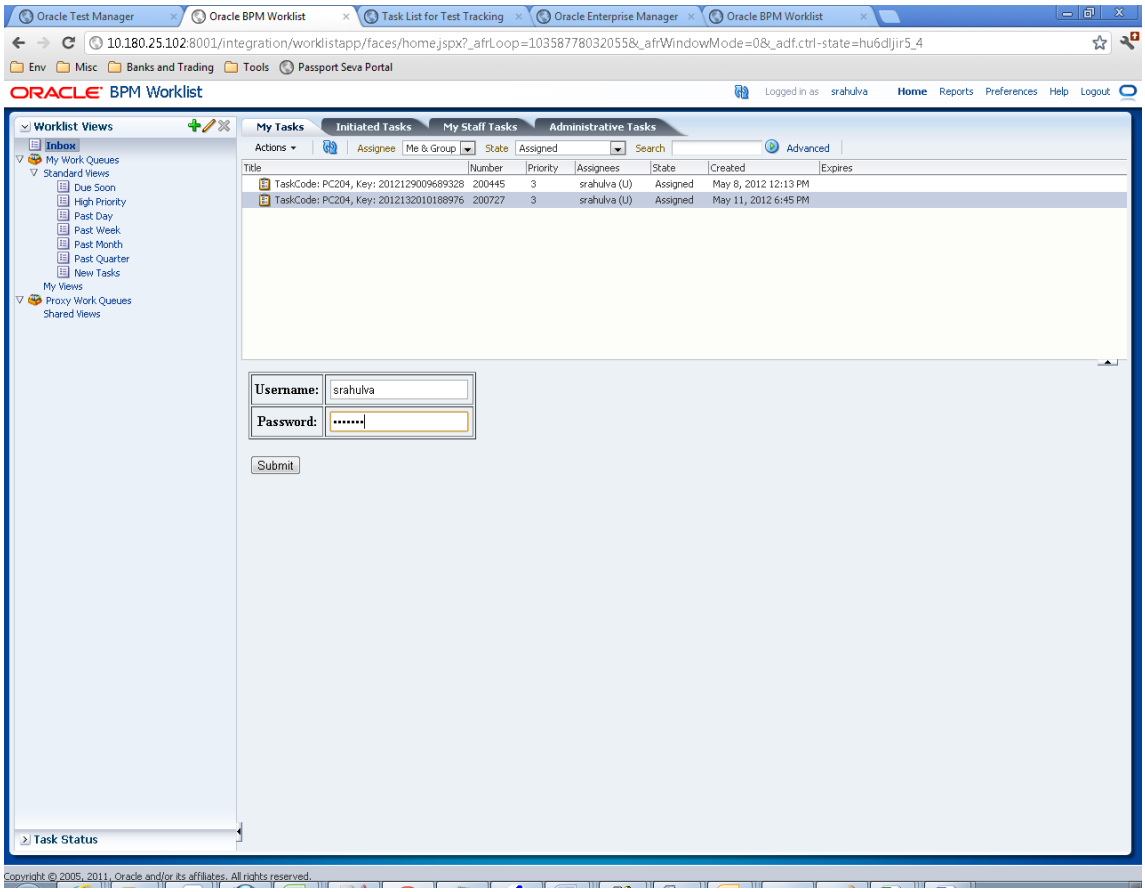
   The application prompts for Log in credentials again as shown in Figure 2–52.

Figure 2–52 Oracle BPM Worklist Log in



6. Enter the same user name and password to open the work item page.

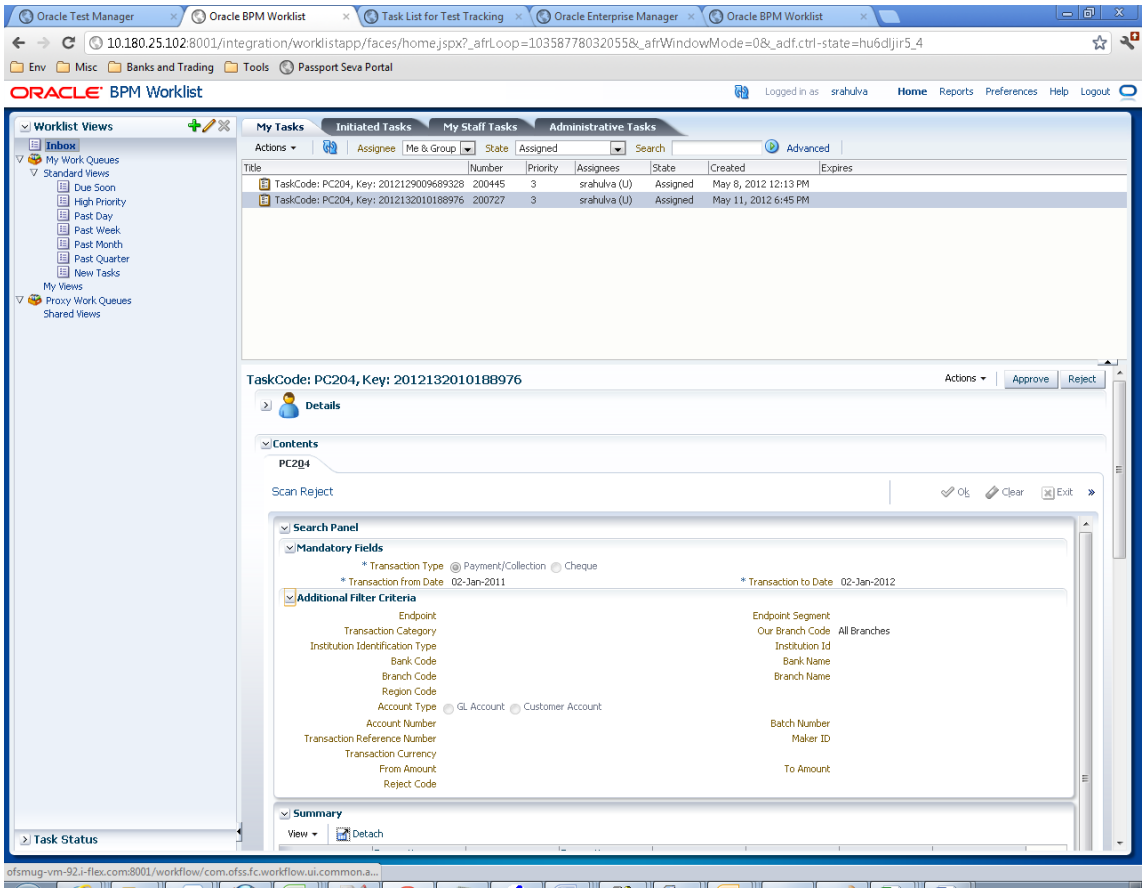
Figure 2–53 Enter Log in Credentials



7. On successful submission of the Log in credentials, you can see the snapshot of the OBP page (Scan Reject in this case) from which the transaction occurred and was sent for approval.
8. Verify whether the page displays the data entered in the OBP application when it was sent for approval.

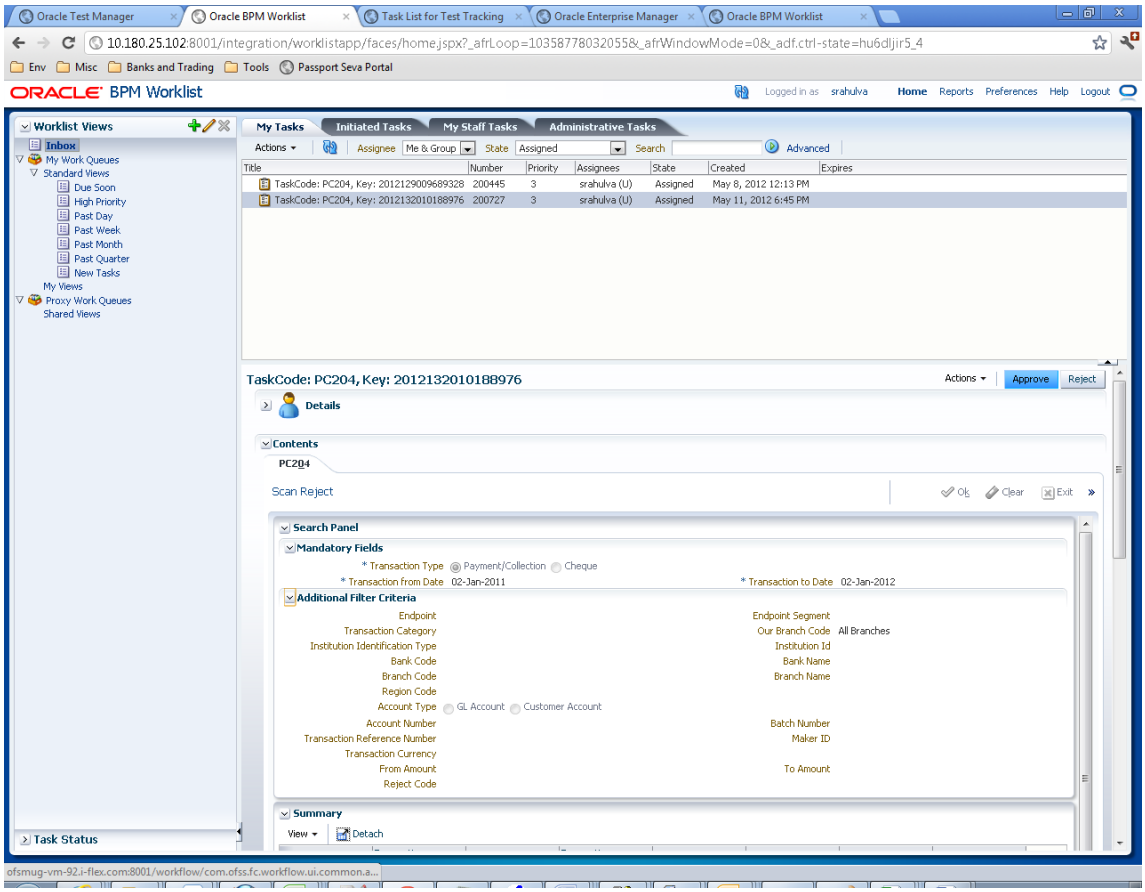
   You can see Approve and Reject option for the work item.

Figure 2–54 My Tasks - Work Item Details



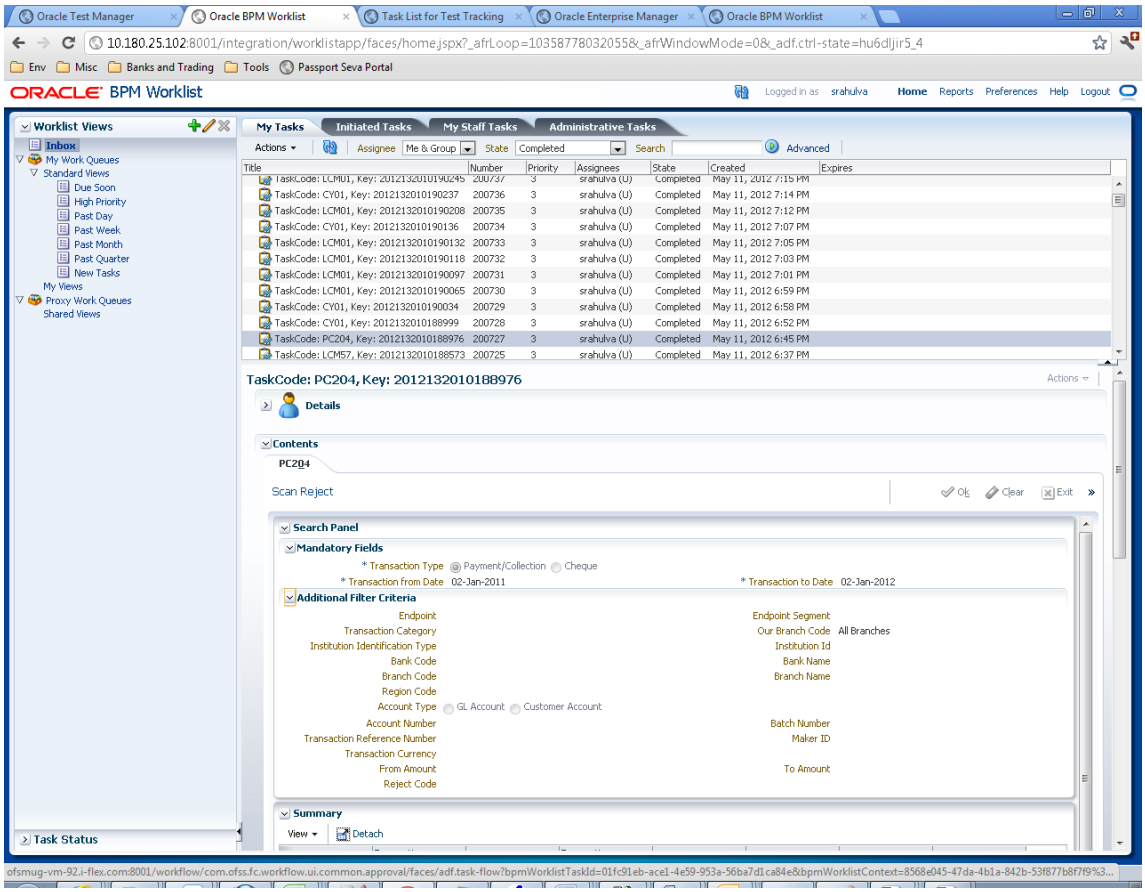
9. Click either **Approve** or **Reject** to perform the appropriate action.

Figure 2–55 Approve or Reject Task



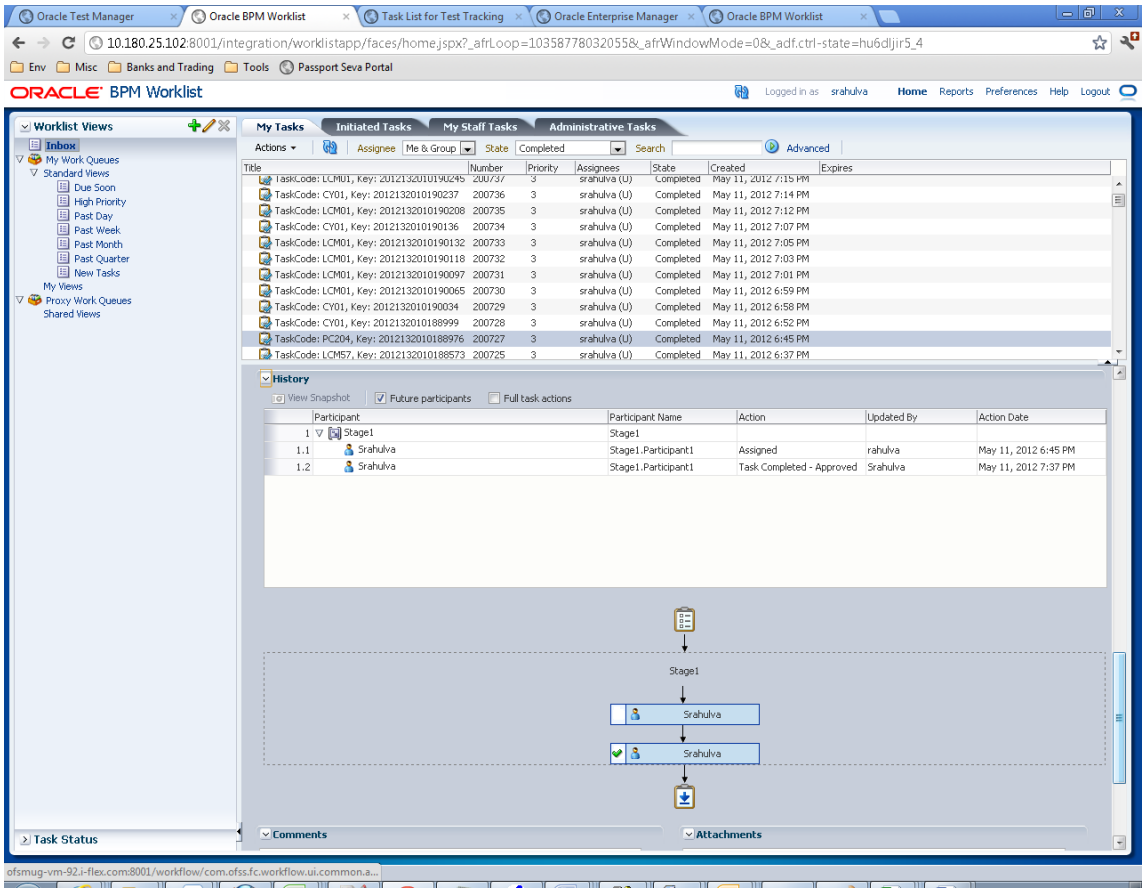
Once the transaction is approved, you can see the status of transaction as *Completed*. This can also be seen by selecting the option *Completed* from **State** list as shown in Figure 2–56.

Figure 2–56 Task Complete



10. Check the history of the work item by scrolling down.

Figure 2–57 Worklist Task History



2.4 Configuring Approvals for Reason Codes Other Than 1000

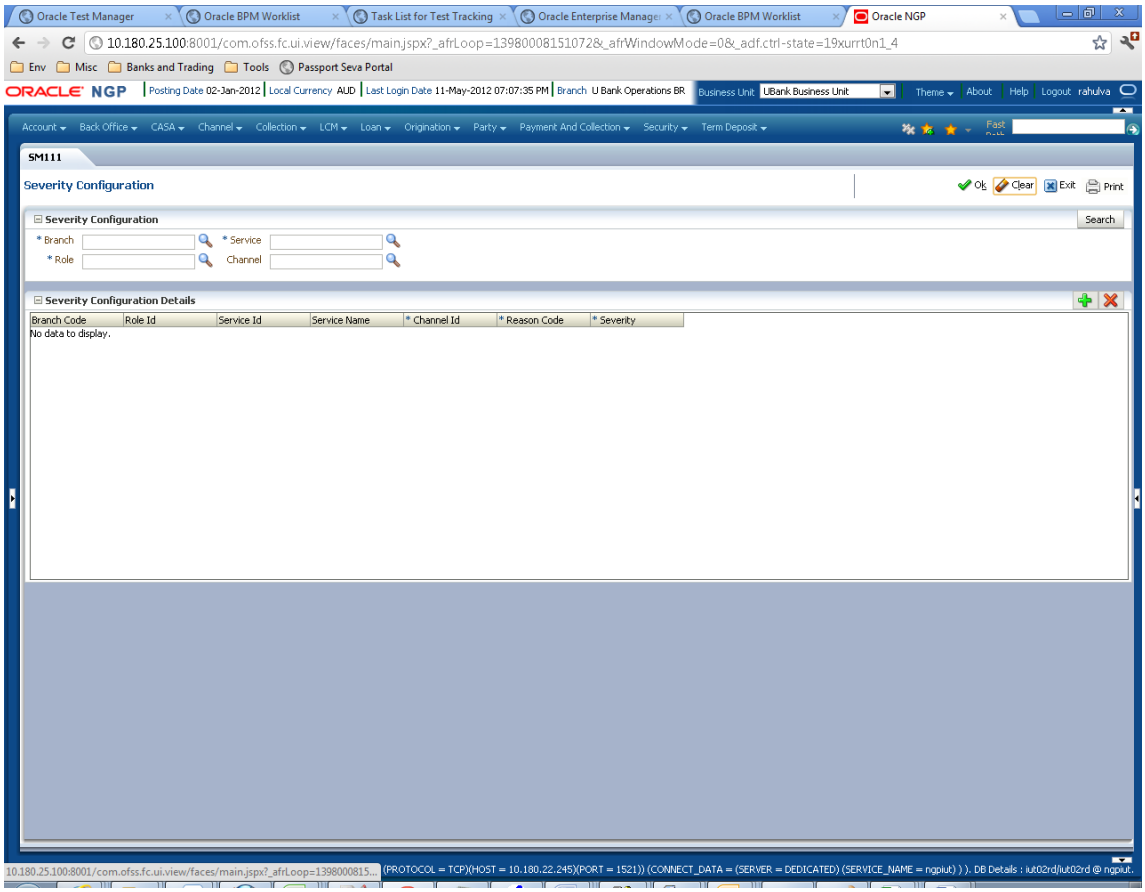
This section explains the steps in configuring approvals for reason codes other than 1000.

For configuring approvals for reason code other than 1000, like High severity memo and so on, you need to set up the same in OBP page **Severity Configuration (Fast path: SM111)**.

Following are the steps to set up approval for other reason codes:

1. Log in to Oracle Banking Platform.
2. Navigate to the page **Severity Configuration (Fast path: SM111)** from the Security menu item (Security- - > Severity Configuration).

Figure 2–58 Severity Configuration (Fast path: SM111)

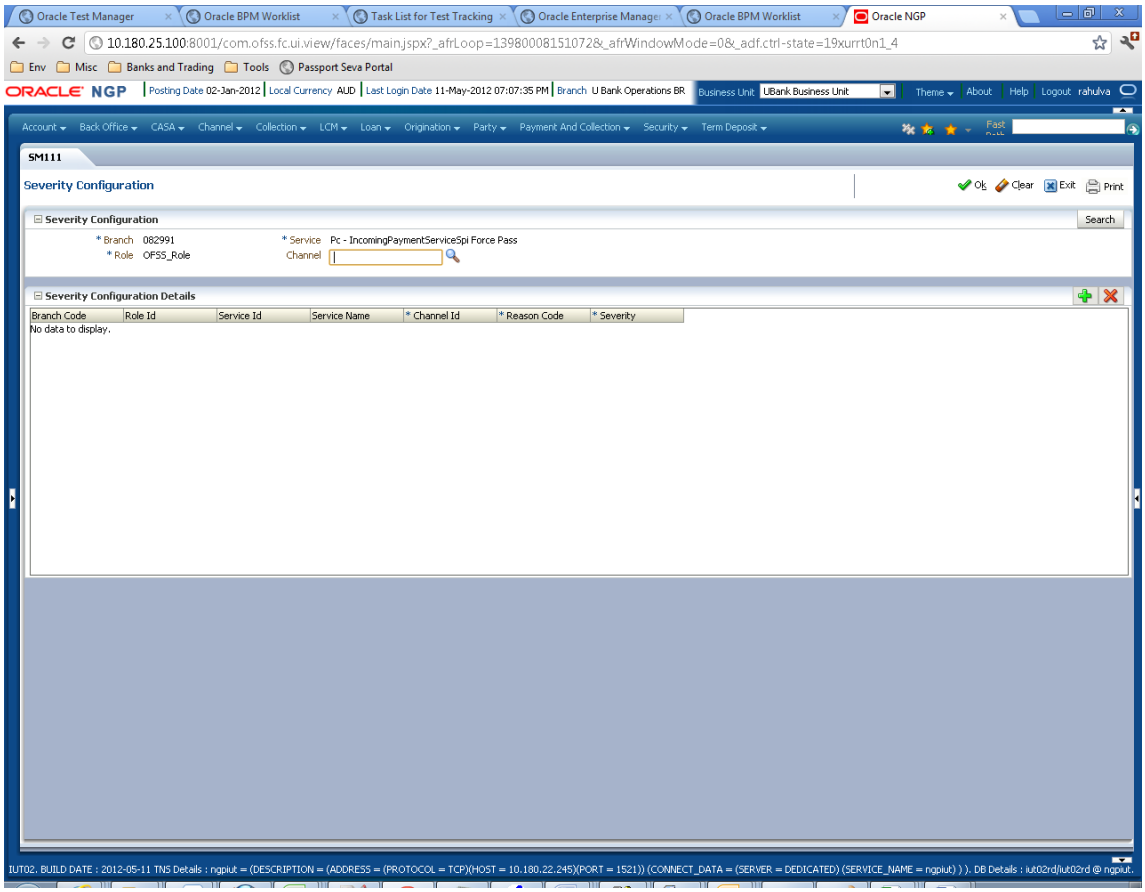


3. Enter the following details in the **Severity Configuration** section.

Branch	Branch code from which the transaction is to be performed. Example: 082991
Role	Security Role to which the user belongs and initiates the transaction.
Service	Enter the Service ID identified. Here, for both APP and APPX services, the configuration for severity needs to be done.
Channel	This is optional. You get to select this option when you proceed to configuration.

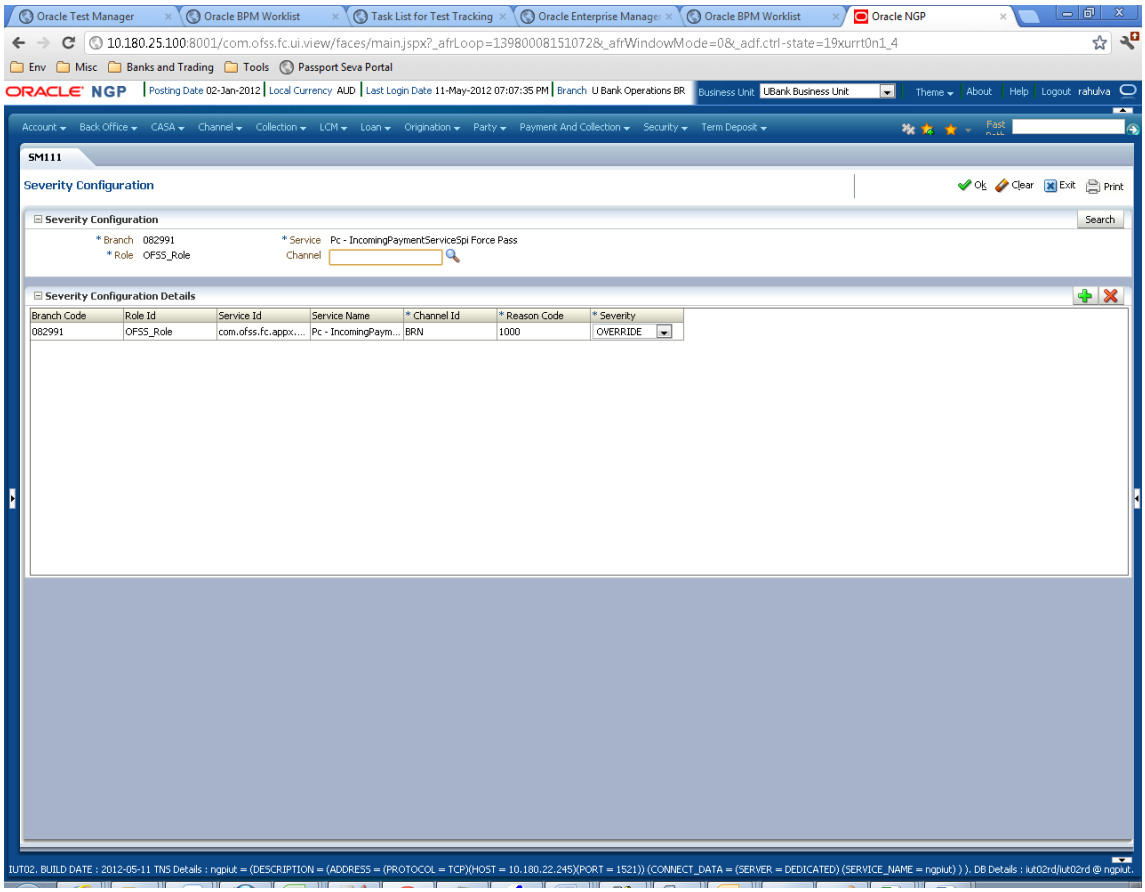
Note: At times you may also notice that the Severity Configuration is already set up. Do not change that.

Figure 2–59 Severity Configuration Details



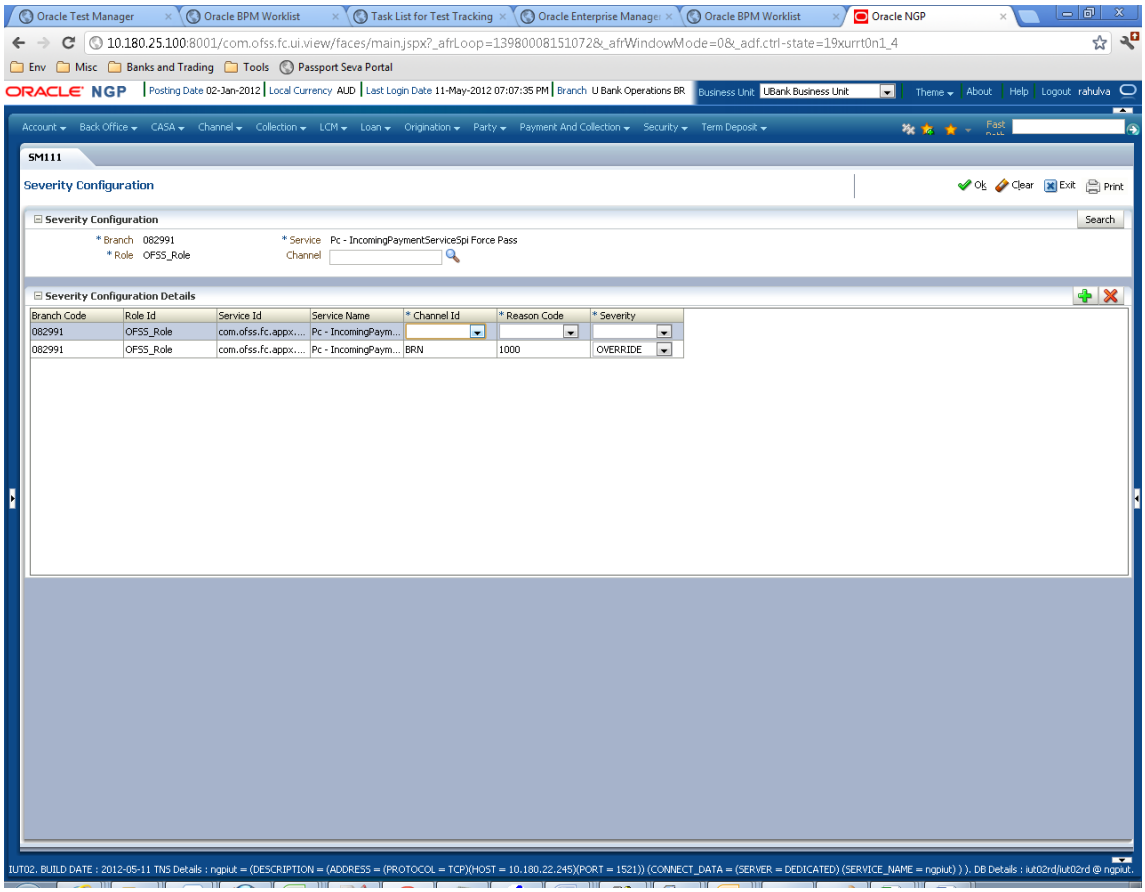
4. Click **Search** on entering the required data. Existing Severity configuration, if any, appears.

Figure 2–60 Severity Configuration - Search



5. In **Severity Configuration Details** section, configure the severity by clicking **Add**, if the desired severity is not configured. This inserts a row as shown in Figure 2–61.

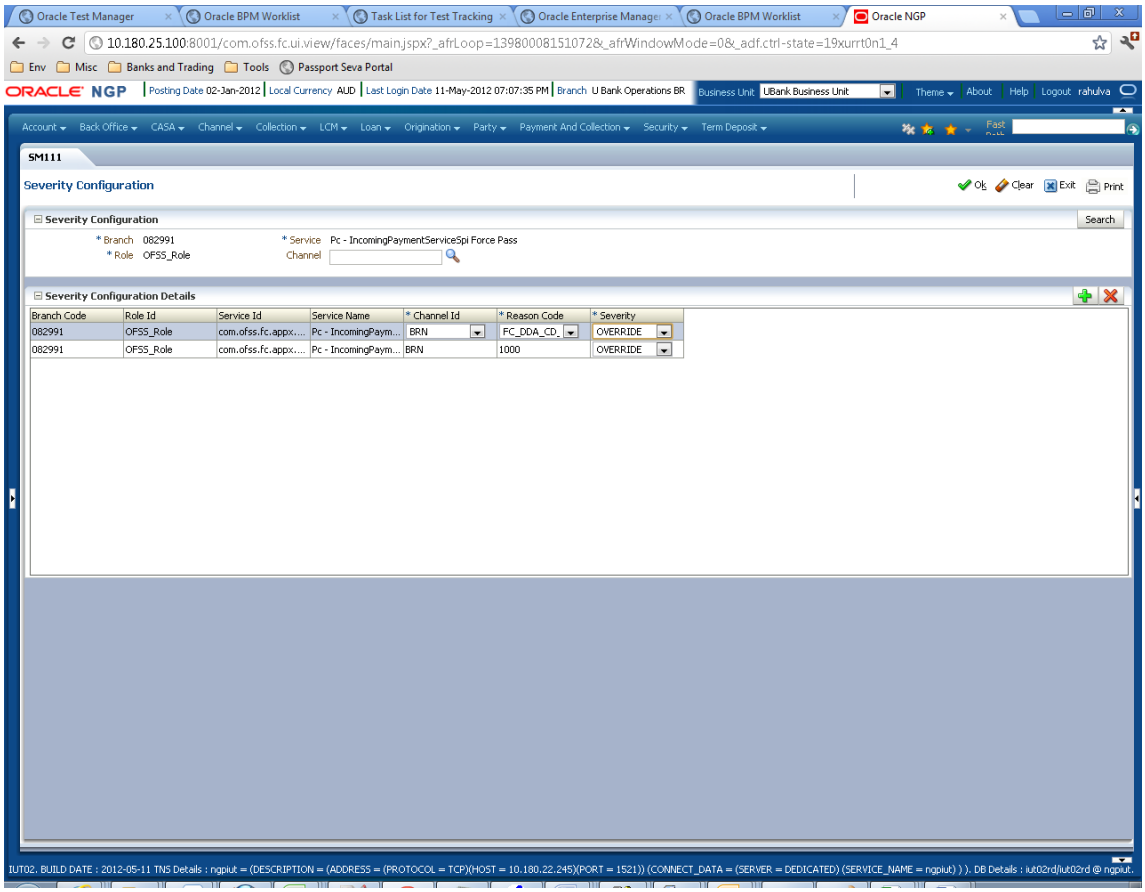
Figure 2–61 Severity Configuration - Add Details



6. Enter the following details in the newly added row in the **Severity Configuration Details** section for configuring the severity:

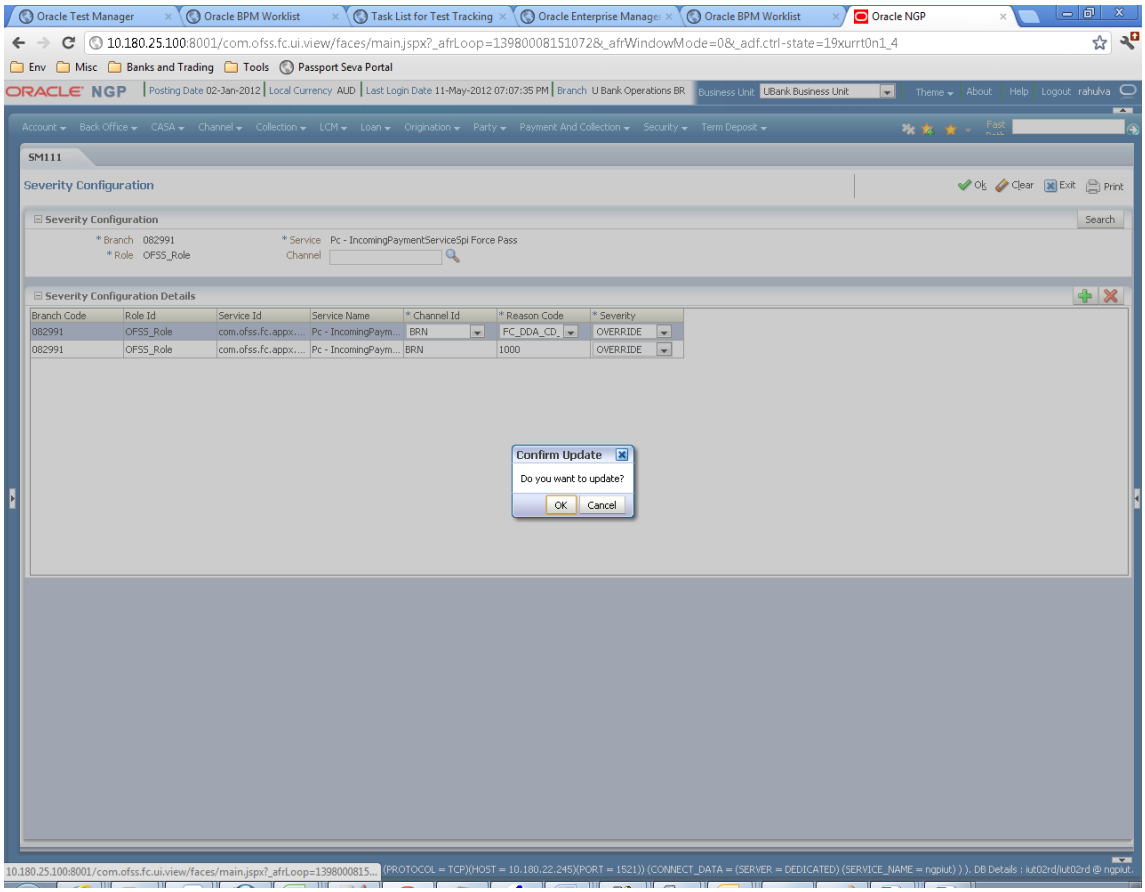
Channel	Channels such as BRN, ATM, IB and so on, through which the transaction is performed.
Reason Code	Select the desired reason code for which configuration needs to be done.
Severity	This field contains four values as detailed below: ■ **Ignore:** Allows transaction to complete without any authorization, that is Auto Authorization. ■ **Override:** Transaction will be sent for Authorization. ■ **Notify:** In this case, the task is not sent for authorization, but the user is expected to confirm the transaction for proceeding ahead. This option is not applicable in case of Dual Authorization. ■ **Reject:** System doe not allow to proceed with transaction.

Figure 2–62 Severity Configuration - Details



7. Click **Ok** after entering the required data.
8. Next click **Ok** to confirm the update as shown in Figure 2–63.

Figure 2–63 Severity Configuration - Confirm Modification



The message appears that the *Operation completed successfully.*

Figure 2–64 Severity Configuration Success

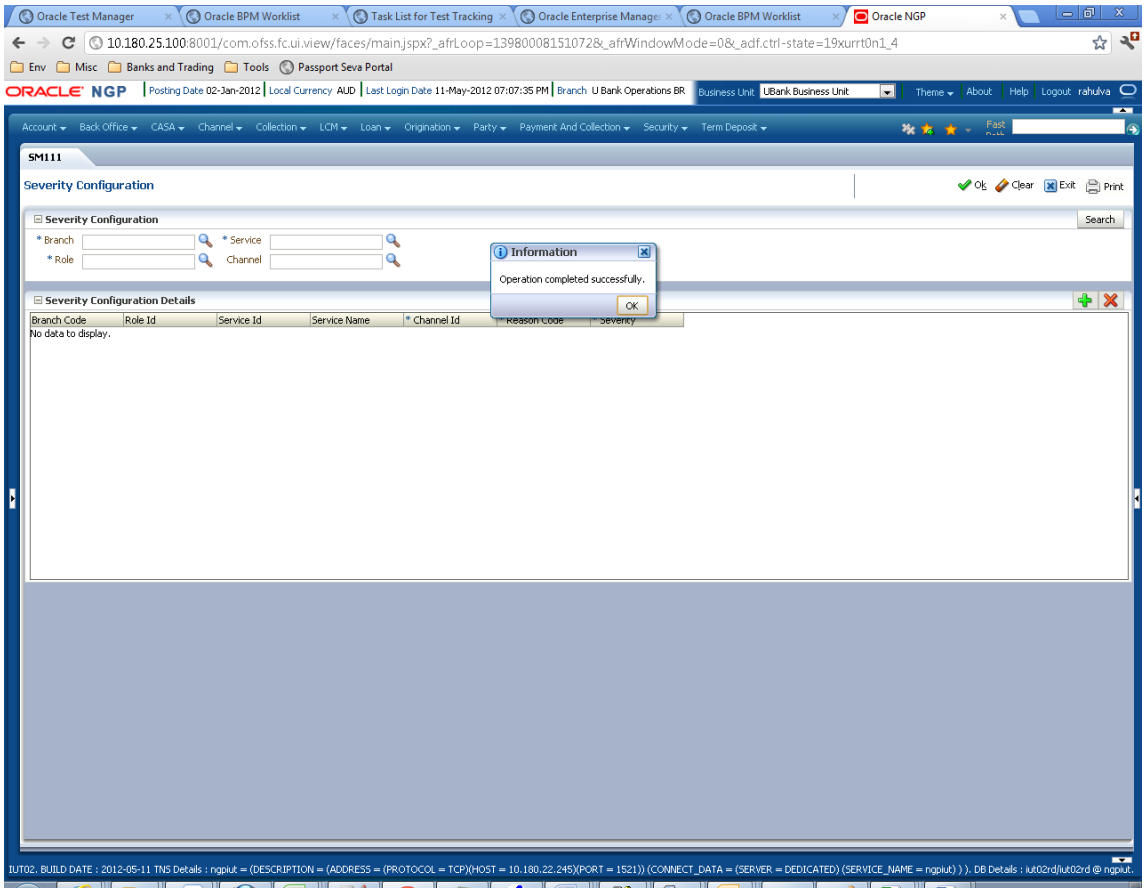
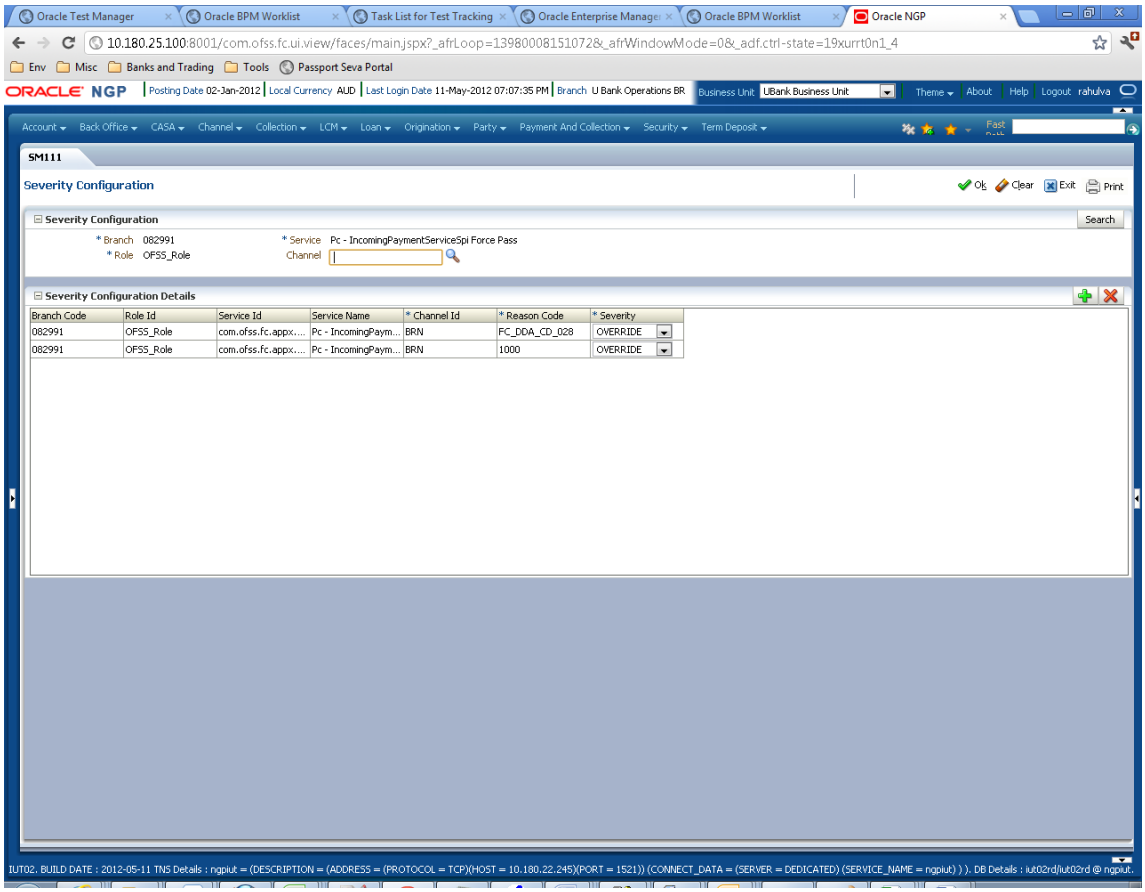


Figure 2–65 Severity Configuration Completed

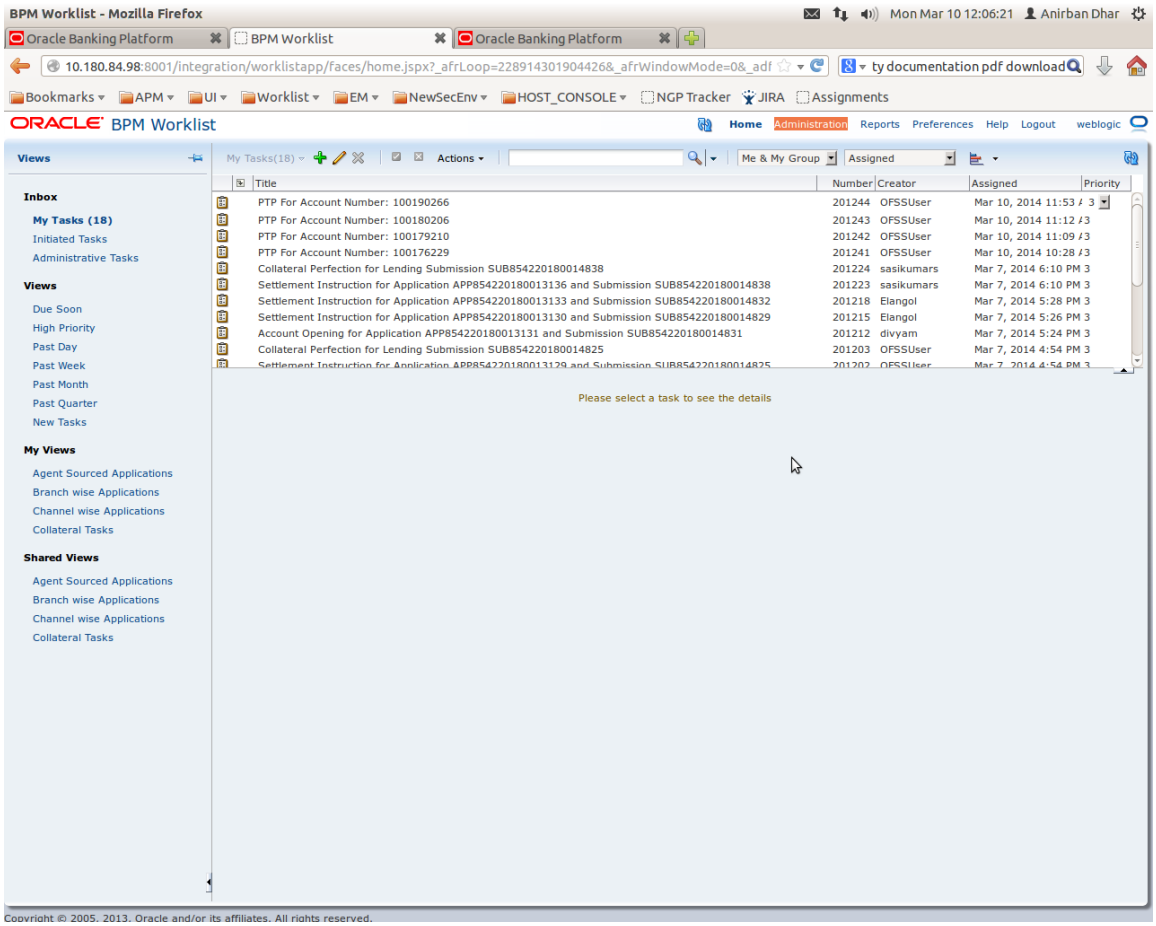


2.5 BPM Routing Rules Setup

This section describes the steps and the screens required to configure APM rules for DPA approval. Same can be referred for configuring any other routing rules based on facts available in human task payload.

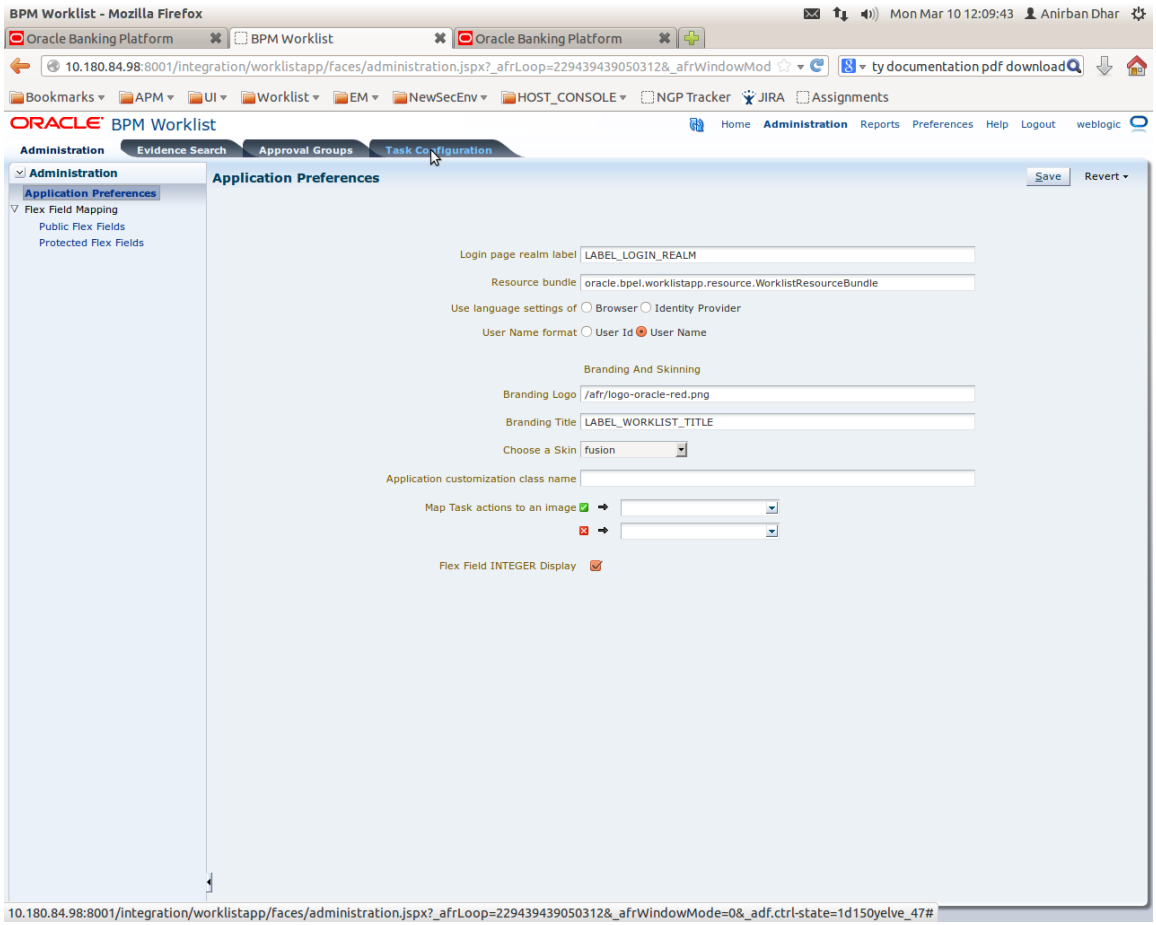
1. Log in to worklist App using Weblogic / Weblogic1 [or any other user with administrators as role]
2. Click **Administration** link.

Figure 2–66 Worklist App - Administration Link



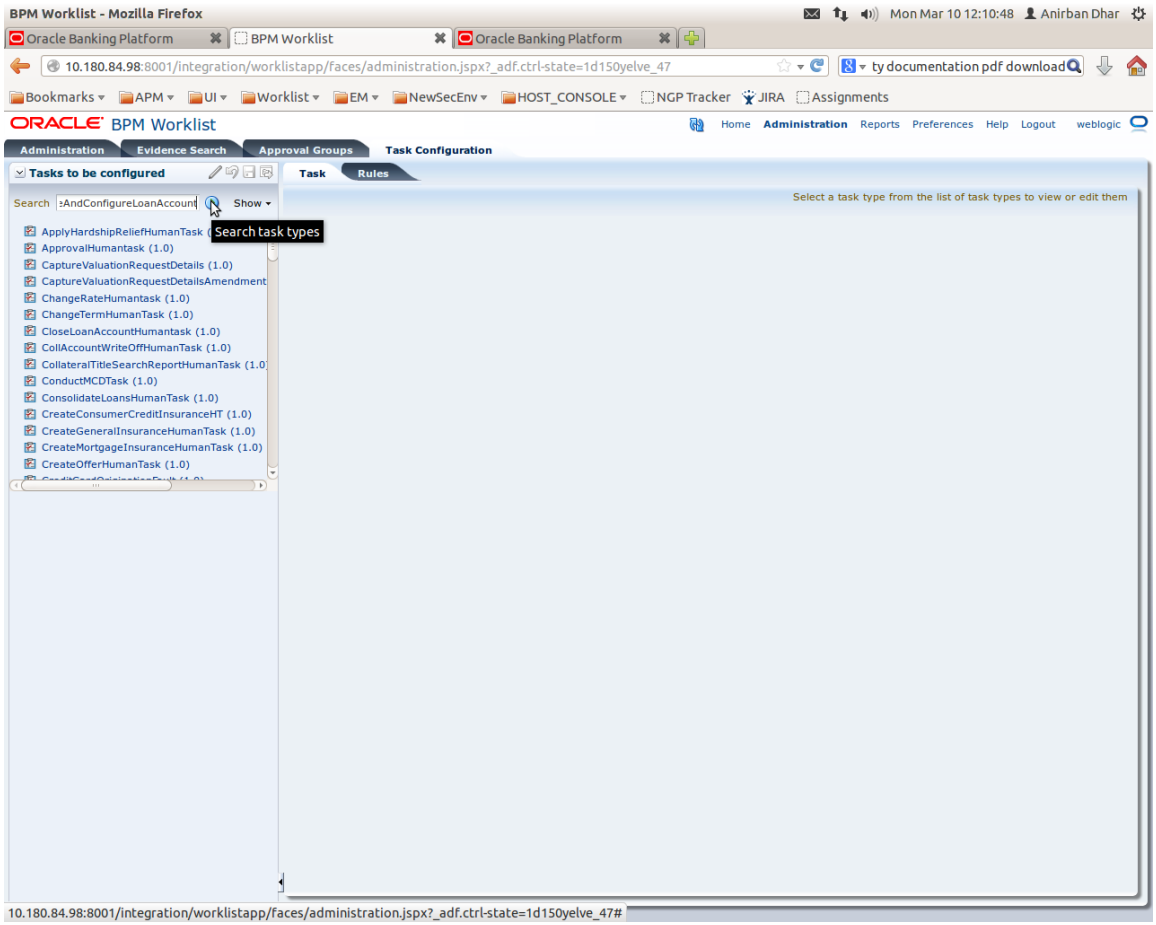
3. Navigate to **Task Configuration** tab.

Figure 2–67 Task Configuration



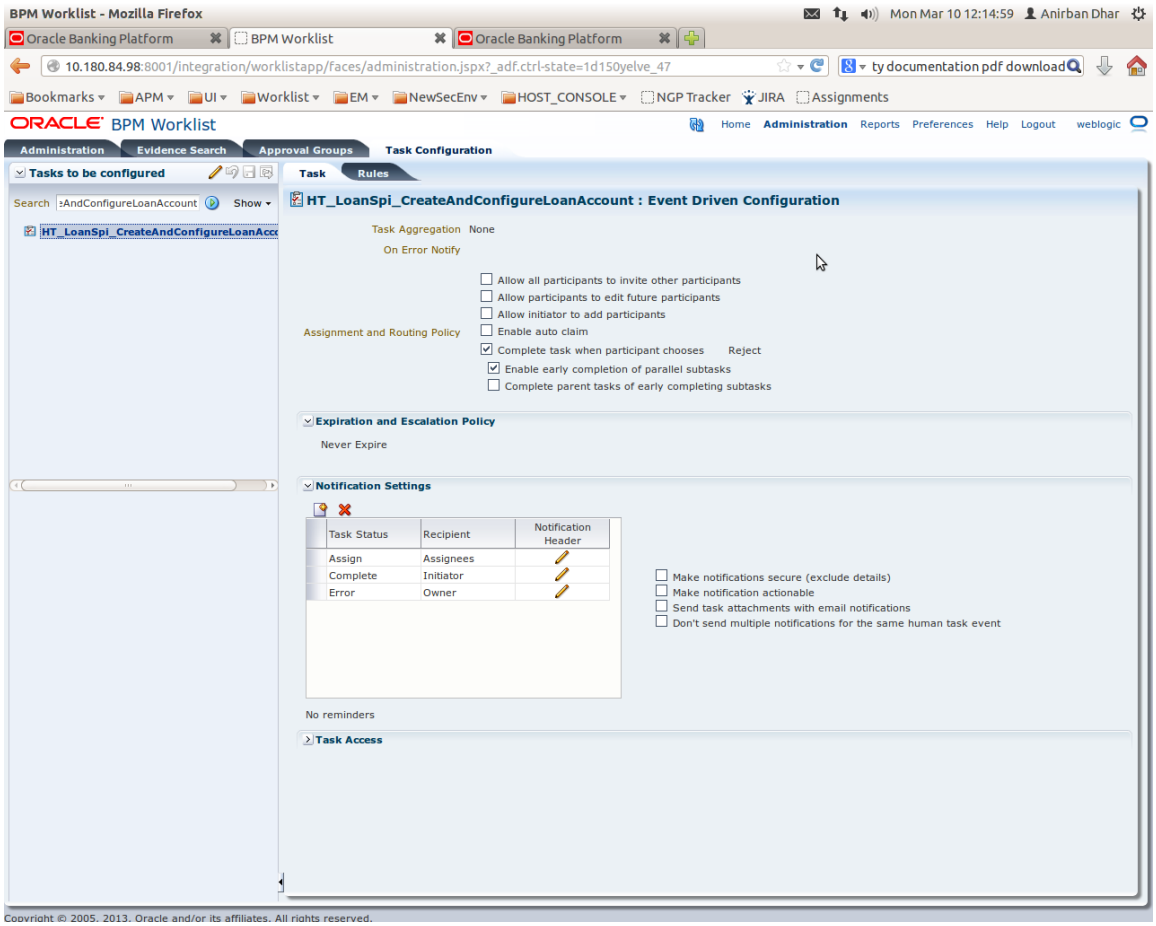
4. Search the task by inputting the name of the task in the search bar. For example, HT_LoanSpi_CreateAndConfigureLoanAccount or *CreateAndConfigureLoan*

Figure 2–68 Search Task Types



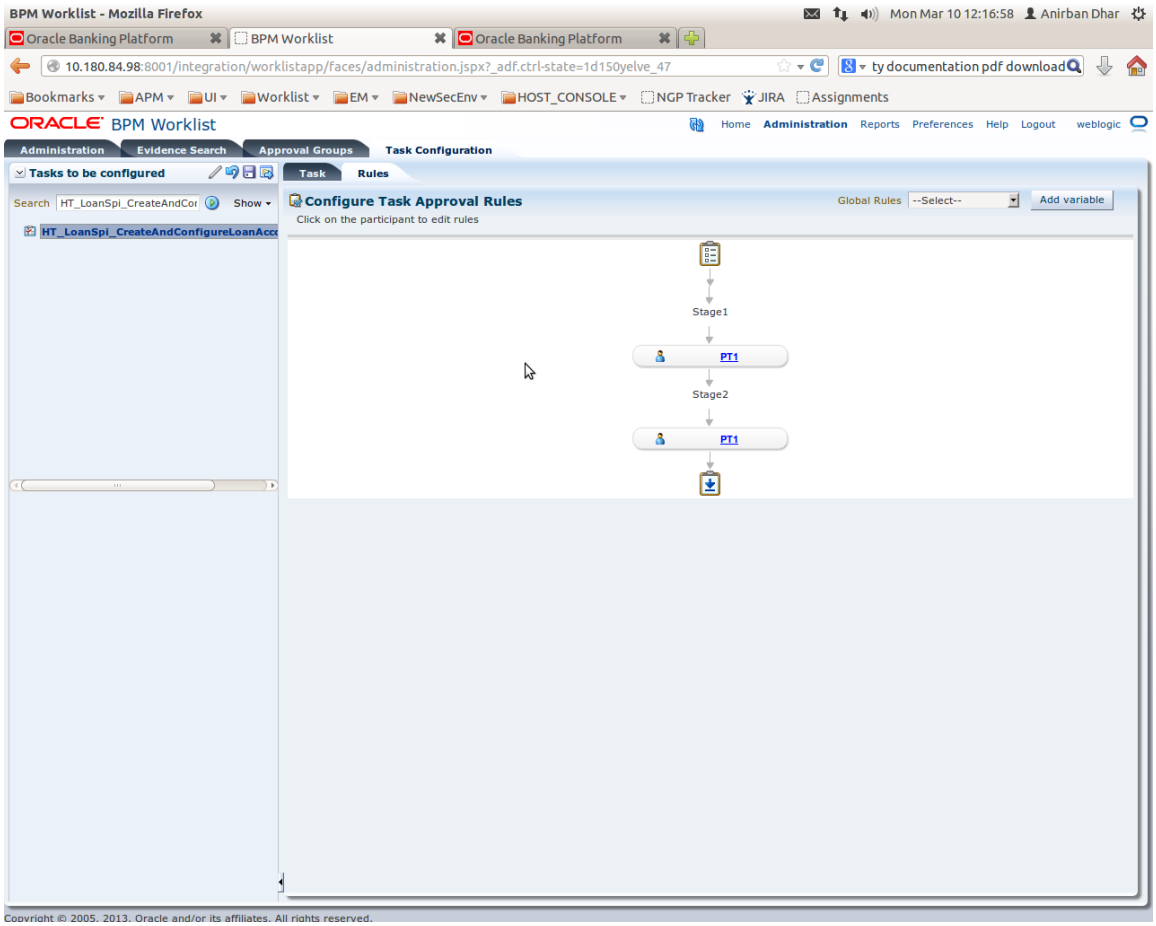
5. Click the result to open the task for editing.

Figure 2–69 Task Editing



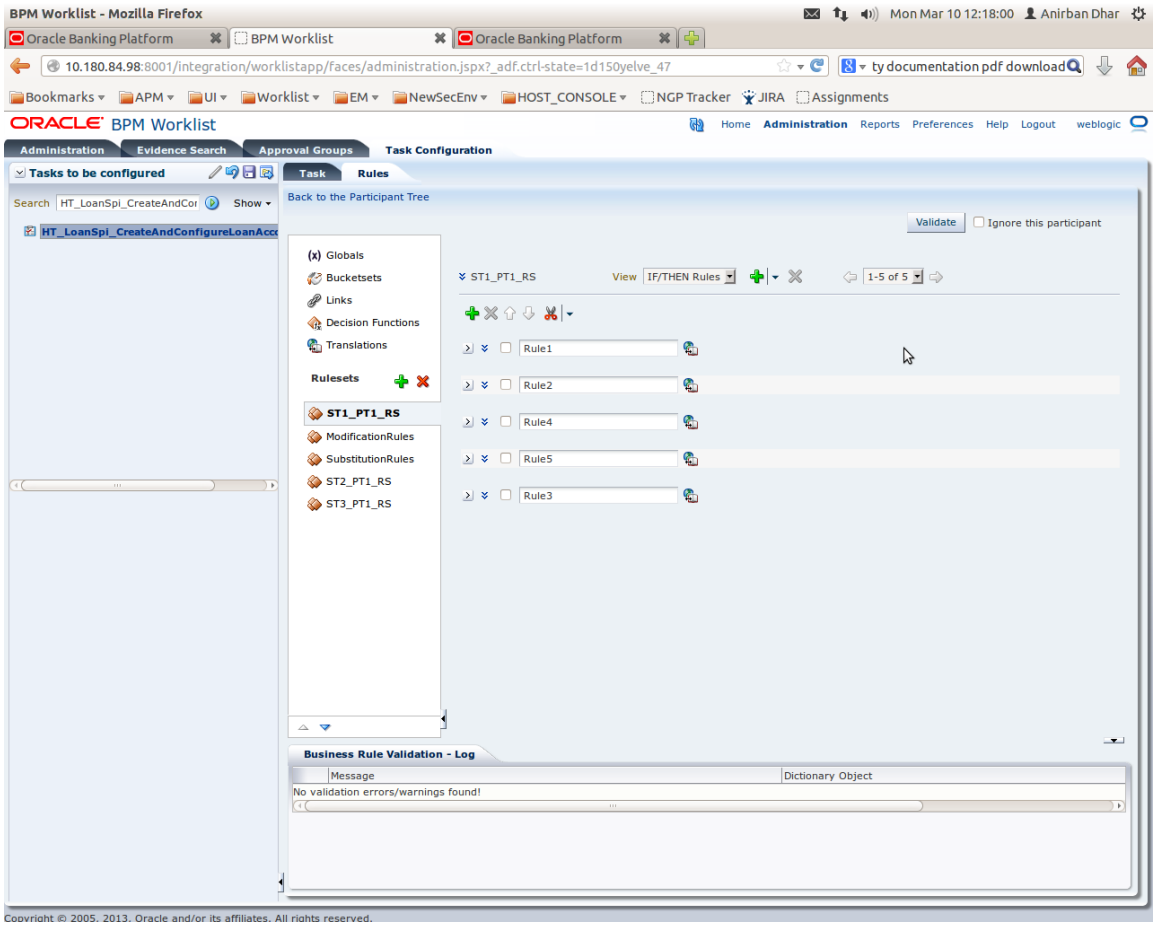
6. Click the Edit icon.

Figure 2–70 Configuring Rules - Edit Icon



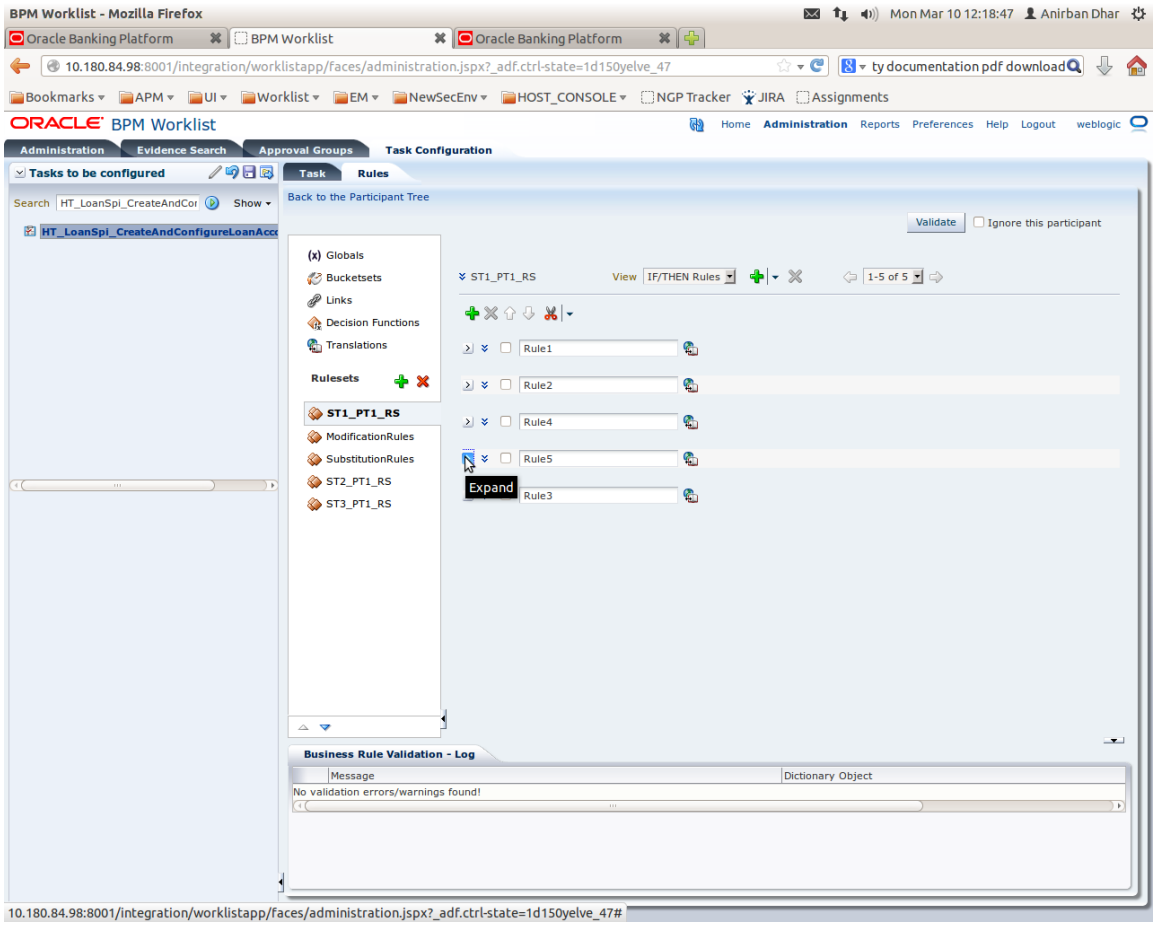
7. Navigate to **Rules** tab for configuring rules.
8. Click participant to edit rule for that stage. For example, stage 1 -> PT1)

Figure 2–71 Configuring Rules - Rules tab



9. Expand the **Rule** to configure. For example, Rule 5.

Figure 2–72 Expand Rule



10. Expand the **Advance Settings** panel.

Figure 2–73 Advanced Settings

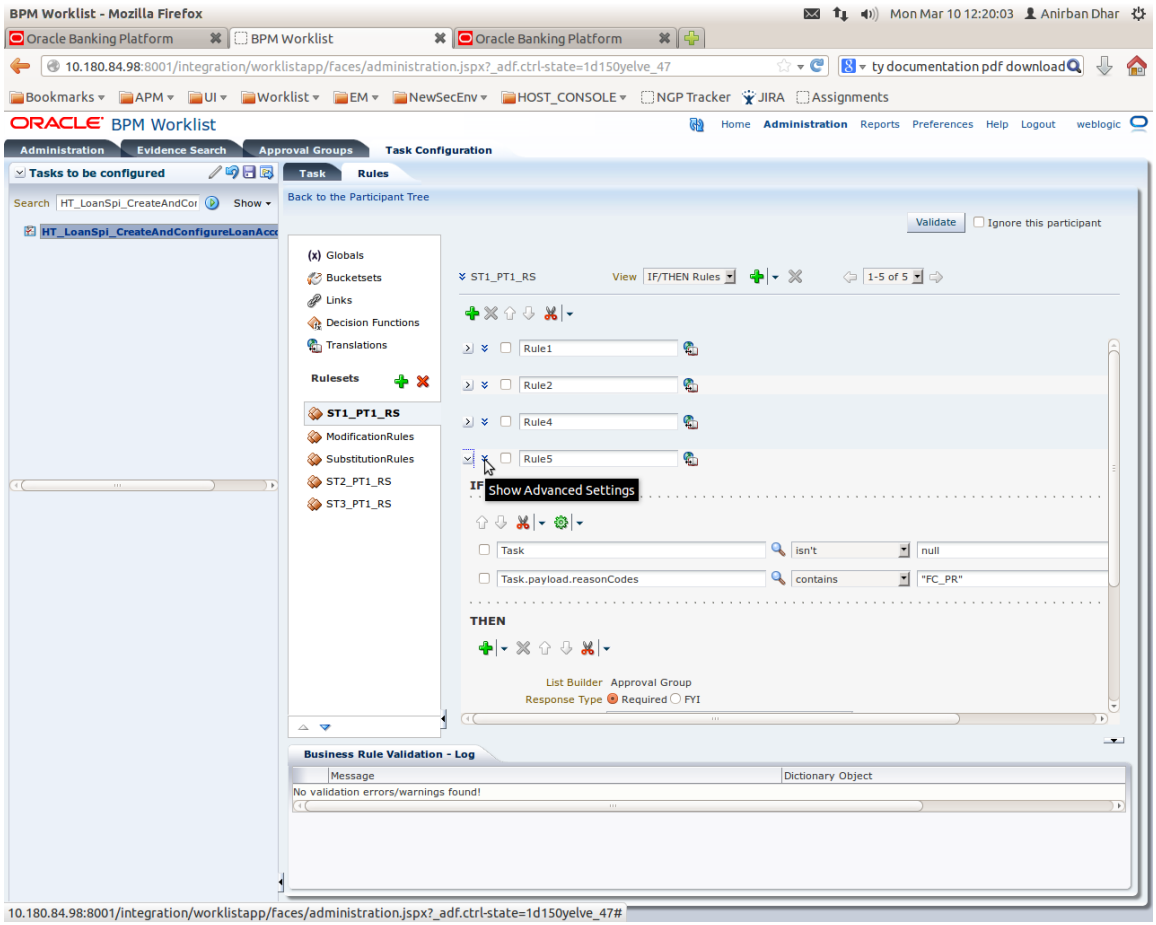
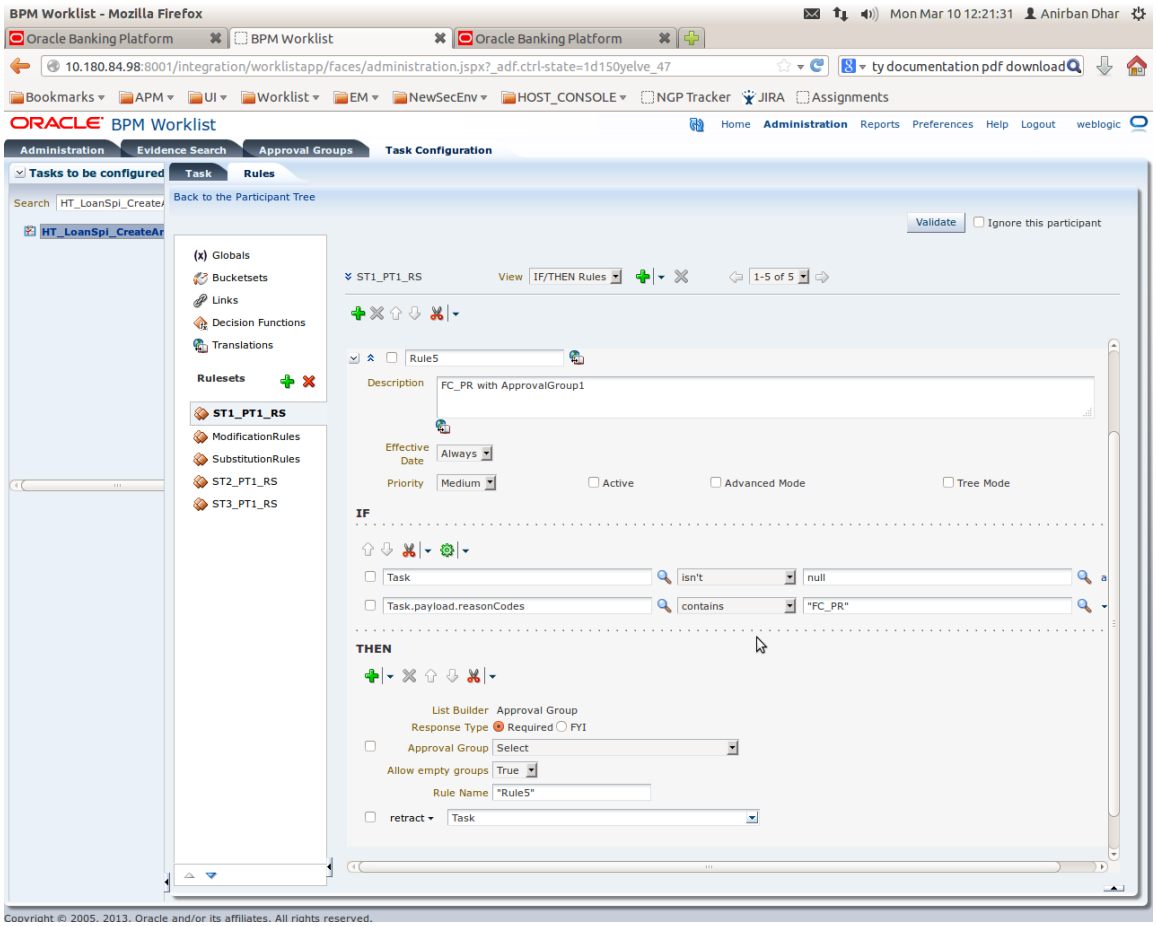
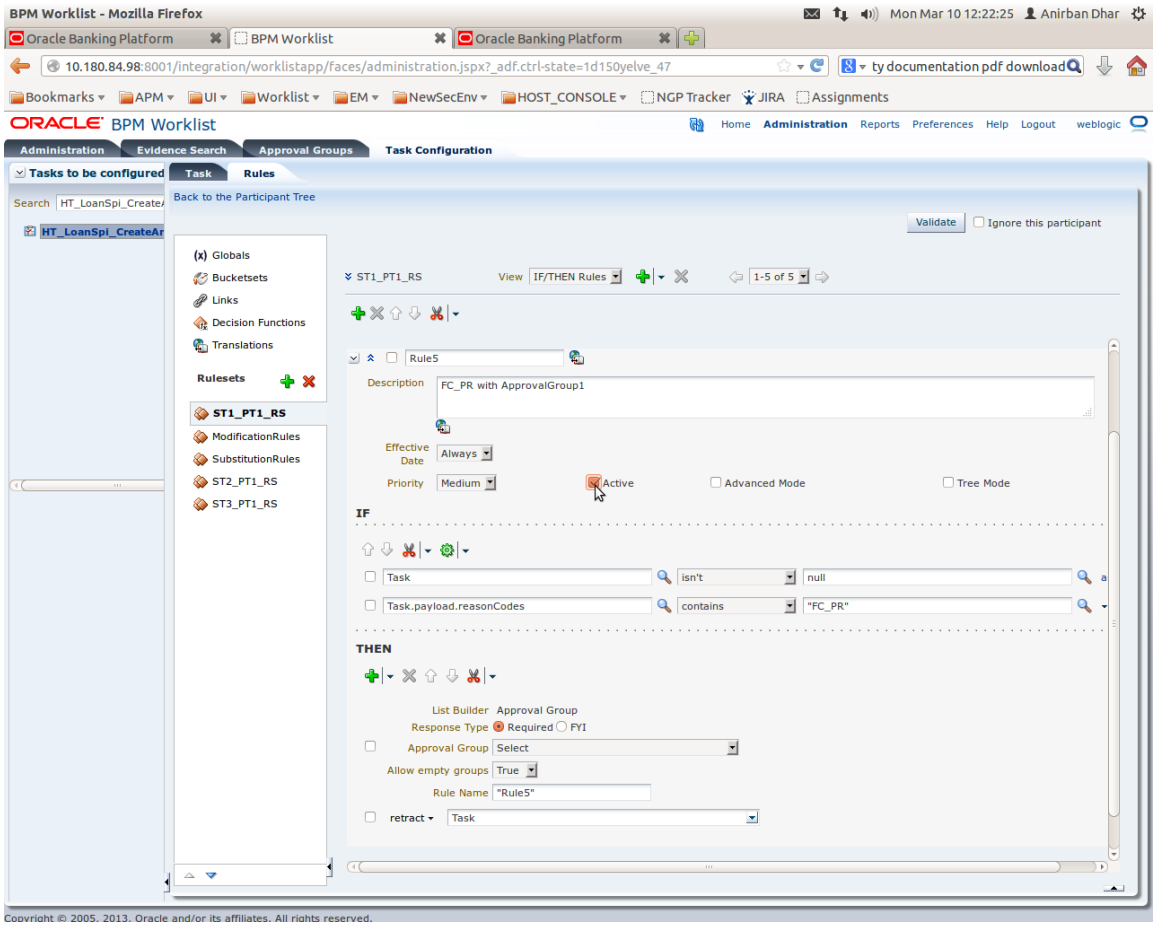


Figure 2–74 Rule Expansion



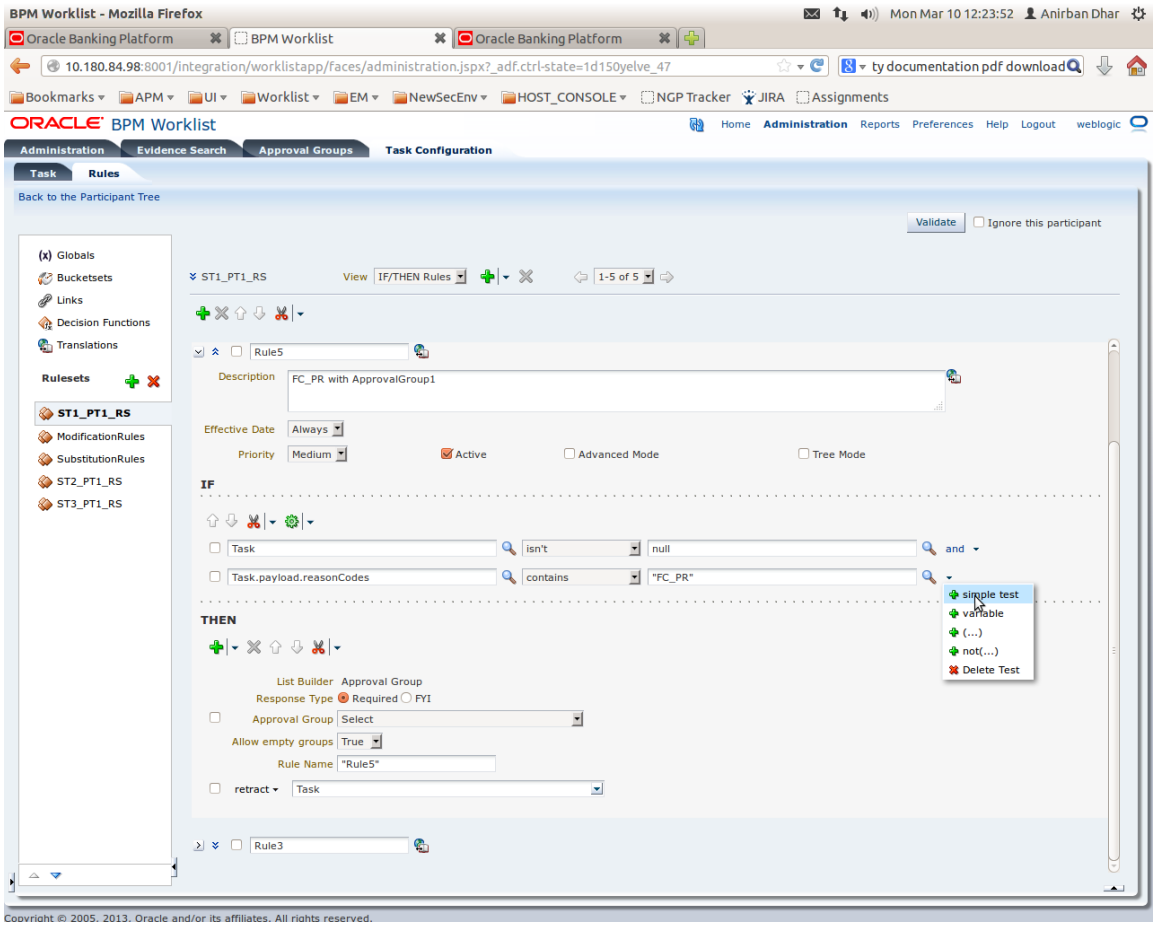
11. Click **Active** check box to enable a particular rule.

Figure 2–75 Enabling Rule



12. Click the drop-down arrow to add a new row of condition and select **Simple Test**.

Figure 2–76 Adding New Row of Condition



13. Expand **Task** ->**Payload** and then select the **fact**.

Figure 2–77 Expanding Payload

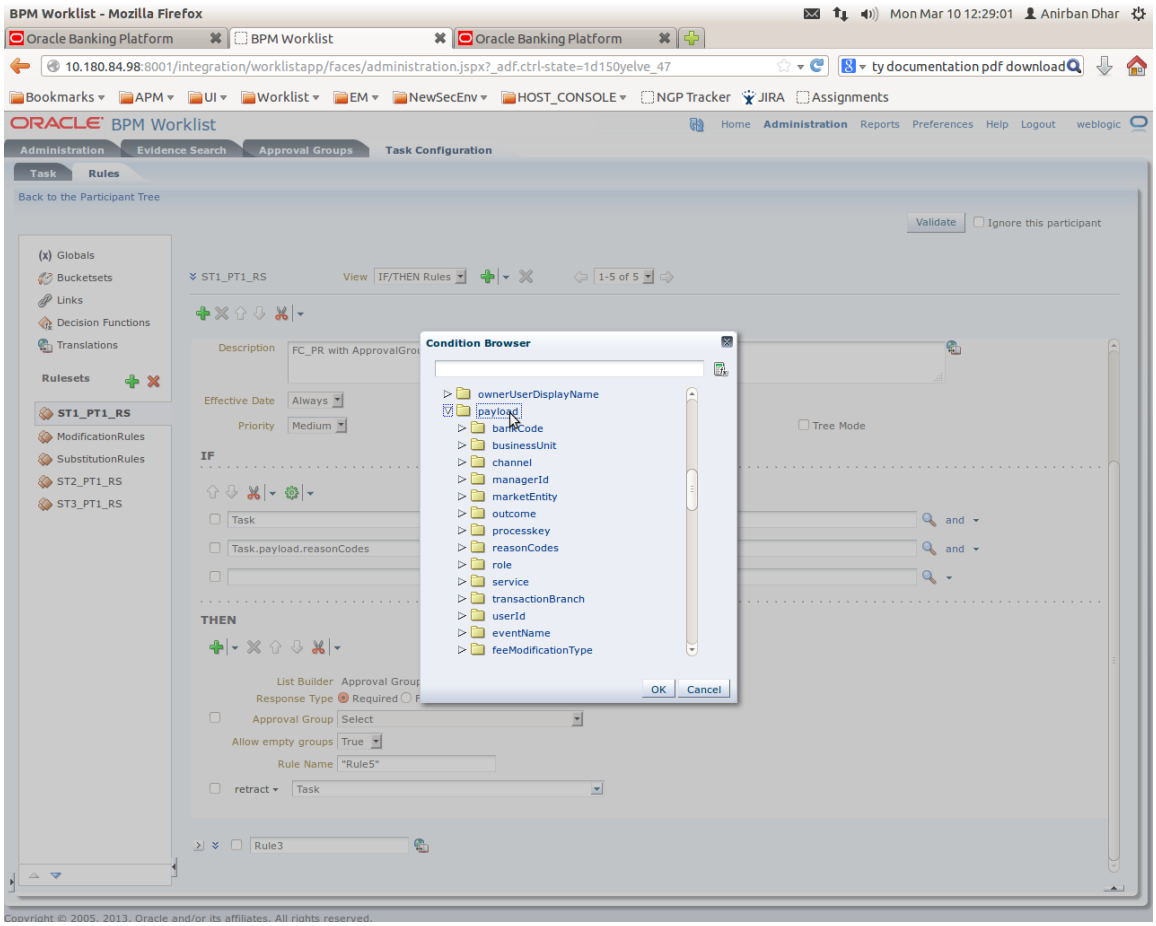
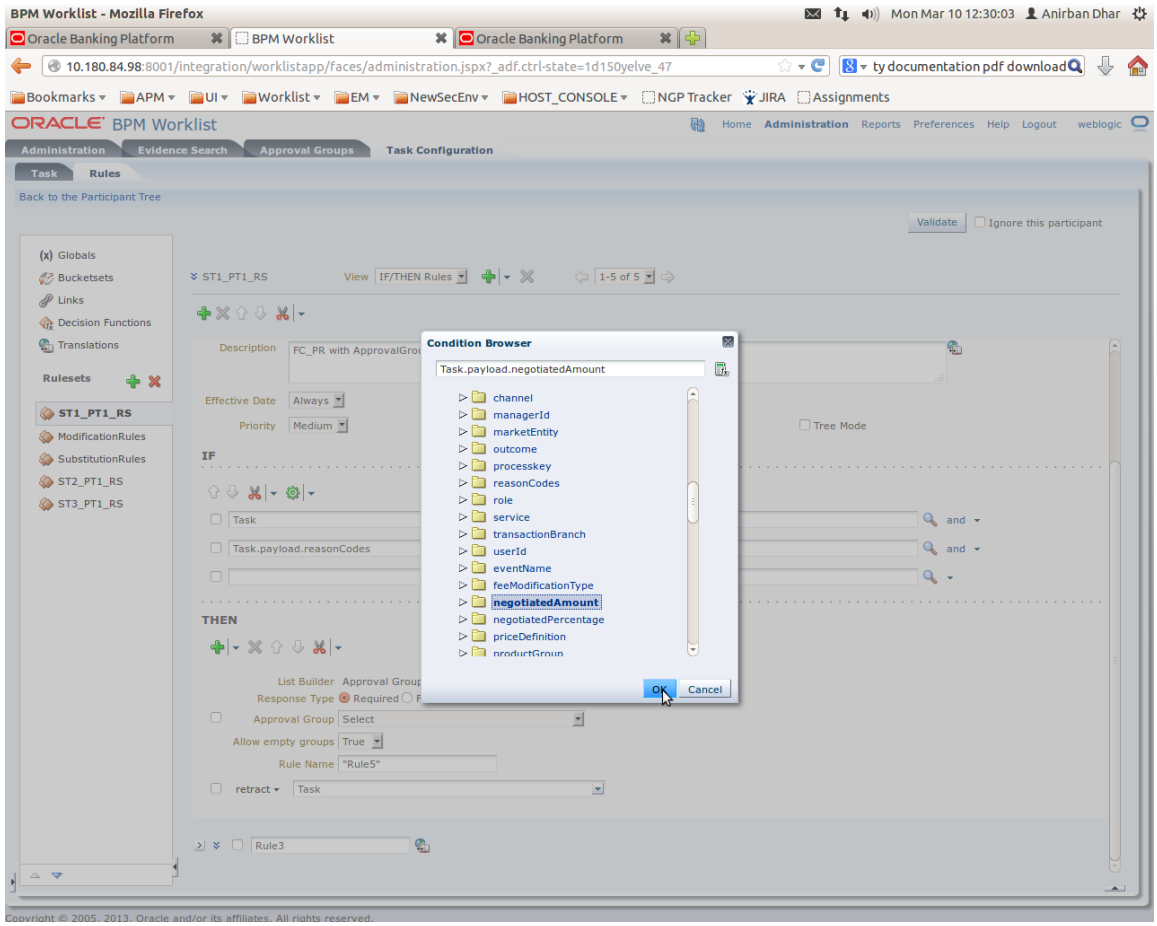
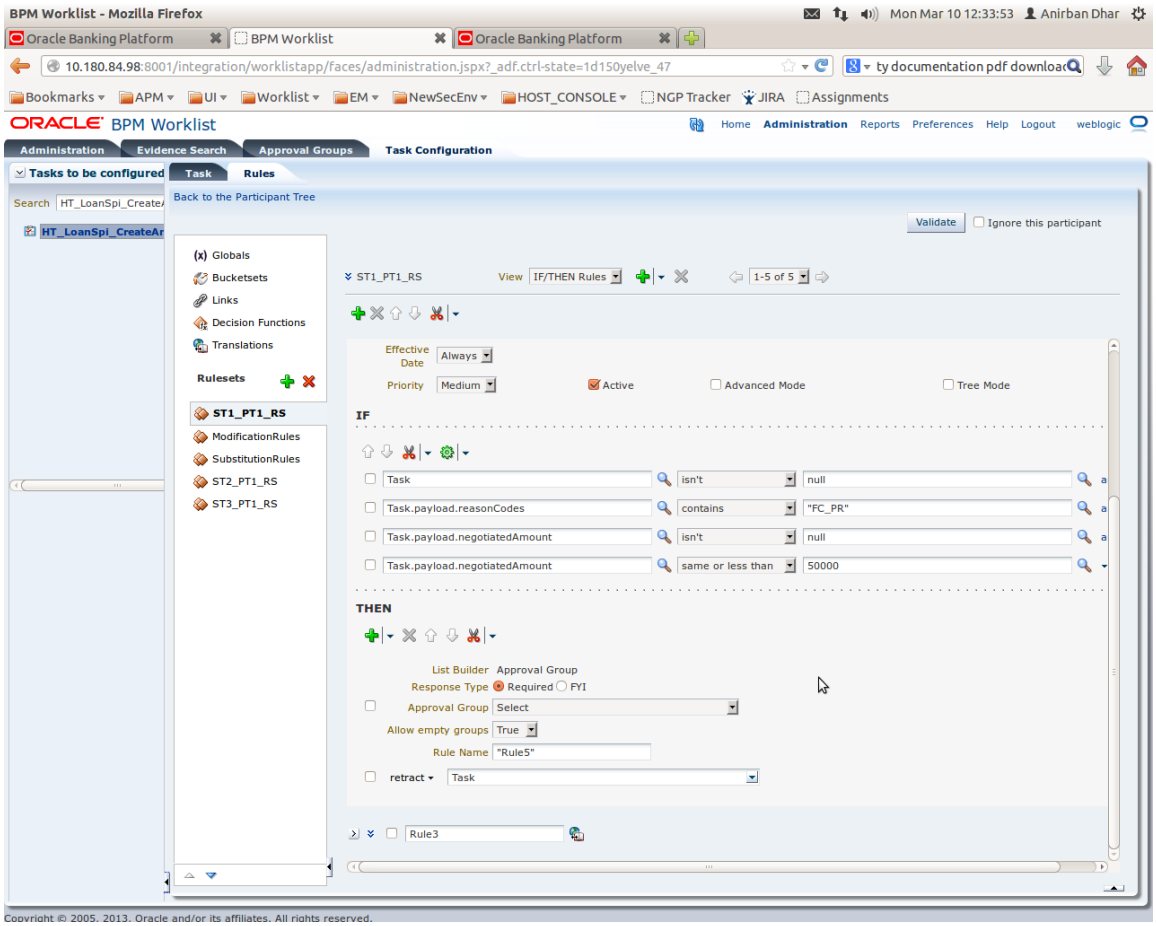


Figure 2–78 Selecting Fact



14. Update the fact value as per condition required for routing.

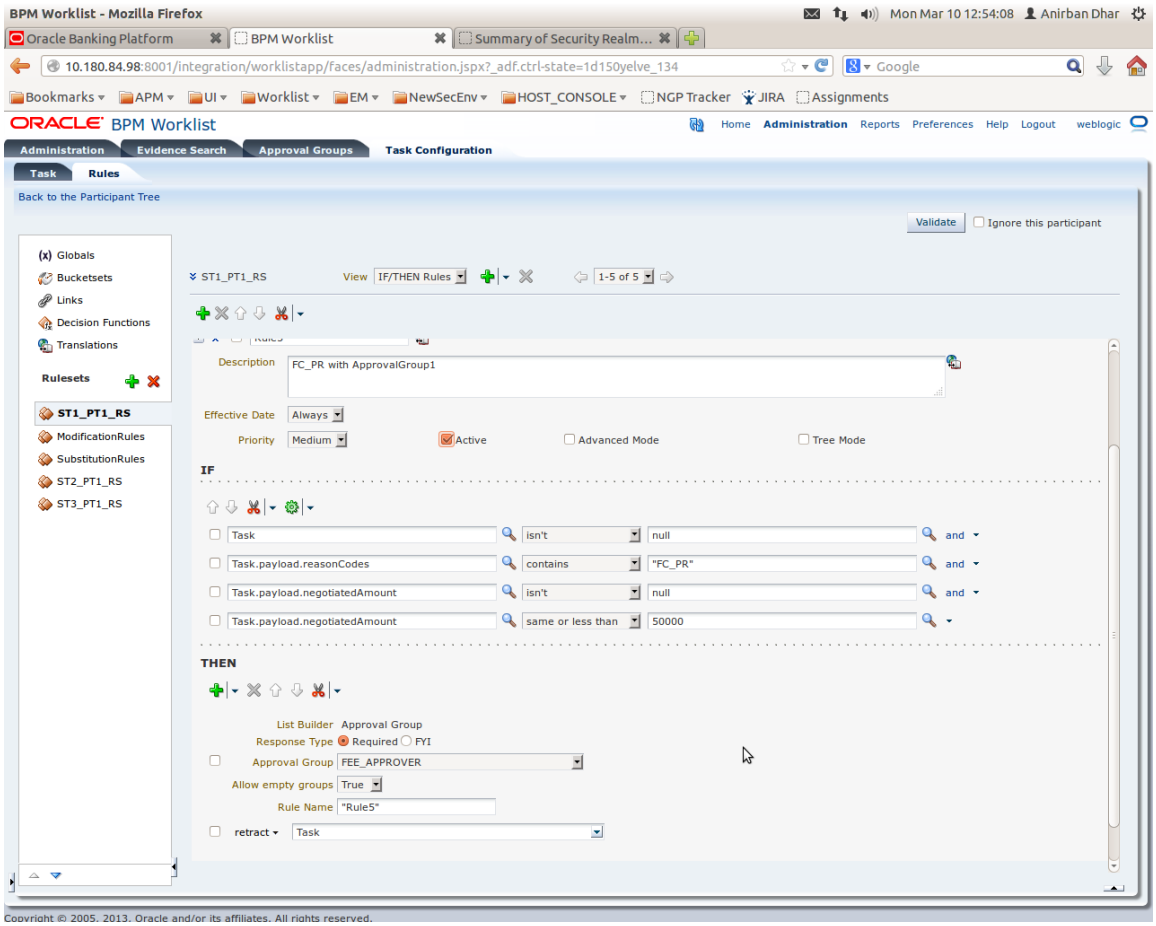
Figure 2–79 Updating Fact Value



Note: Negotiated Amount should be multiplied by 10000. For example, if you want to set limit for $5 negotiated amount, then in BPM rule you need to put fact value as 50000.

15. Select **Approver Group** from the option list. For example, select FEE_APPROVER.

Figure 2–80 Select Approver Group



16. Save and commit the rule and the task.

    You can follow similar steps to configure rules on different stages. Ensure the desired rule is active.

    If no rule is evaluated, then the task will be auto rejected by workflow system.

3

Data Management

This chapter describes data related activities to be performed as an administrator.

3.1 Oracle Banking Platform Batch Execution

Oracle Banking Platform Batch Execution refers to bulk processing of records to perform business operations in real-time environment. Business operations include complex processing of large volumes of information, that is most efficiently processed with minimal or no user interaction using OBP Batch Execution.

In OBP, the batch process is run through the **End of Day (Fast path: EOD10)** page with a varied combination of category, job code and job type for a particular business day.

This section explains the steps involved in Oracle Banking Platform Batch Execution.

> **Note:** To view the detailed procedure to be followed in the application page **End of Day (Fast Path: EOD10)**, see its context-sensitive help in the application.

3.1.1 Database Backup

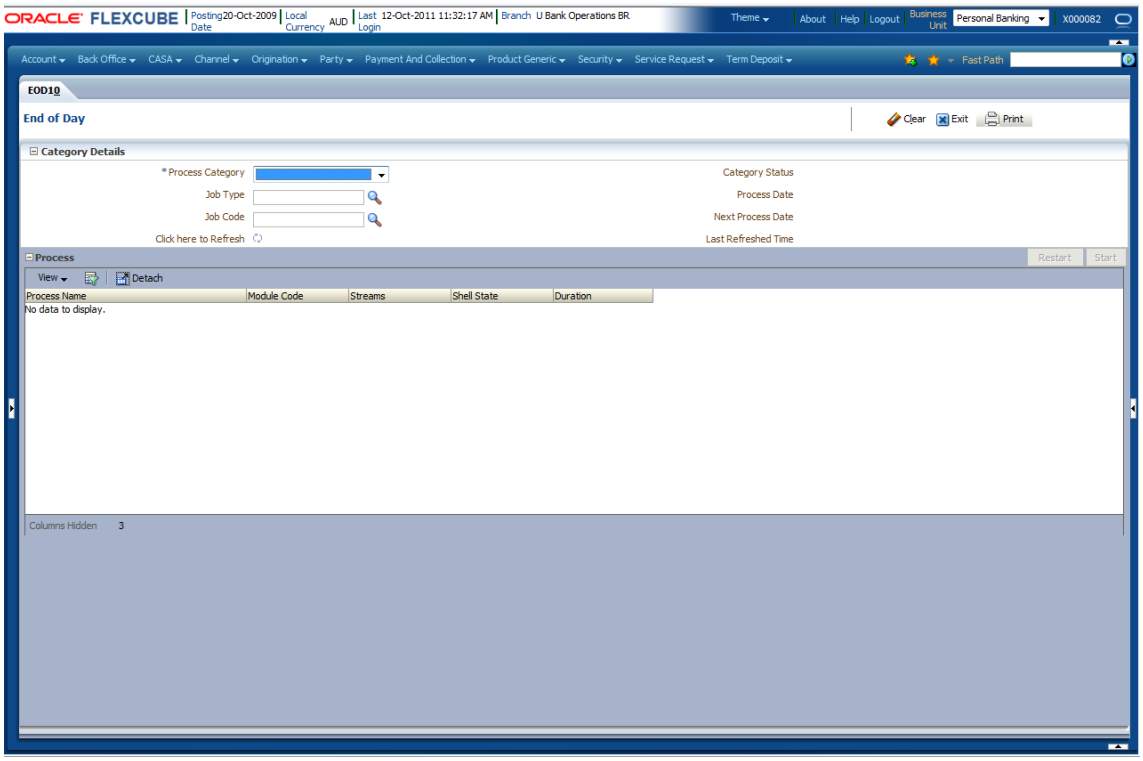
Perform Database Backup before starting with the Batch Execution.

3.1.2 Navigate to End of Day Page

To navigate to the End of Day page:

1. Log in to OBP.
2. Navigate to *End of Day* page either by entering the Fast path **EOD10** or through the menu **Back Office --> Global --> End of Day.**

Figure 3–1 End of Day (Fast path:EOD10)



3.1.3 Cutoff Category Execution

To execute the Cutoff category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	Cutoff
Job Type	GROUP
Job Code	BRN_GRP_1

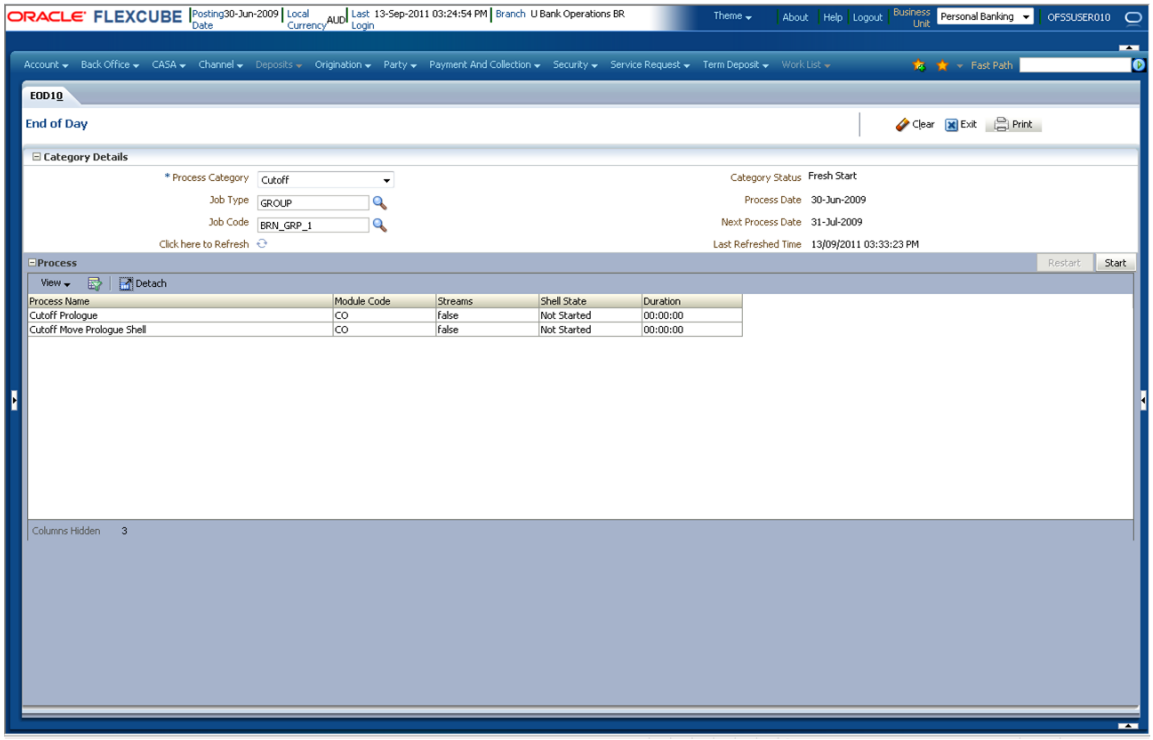
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started*.

   The **Category Status** is *Fresh Start*.

3. Verify the **Process Date** and the **Next Process Date.**

Figure 3–2 Cutoff



4. Click the **Start** button to begin the execution.

   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress.*

5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed.*

3.1.4 End of Day (EOD) Category Execution

To execute the End of Day category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	End of Day
Job Type	GROUP
Job Code	BRN_GRP_1

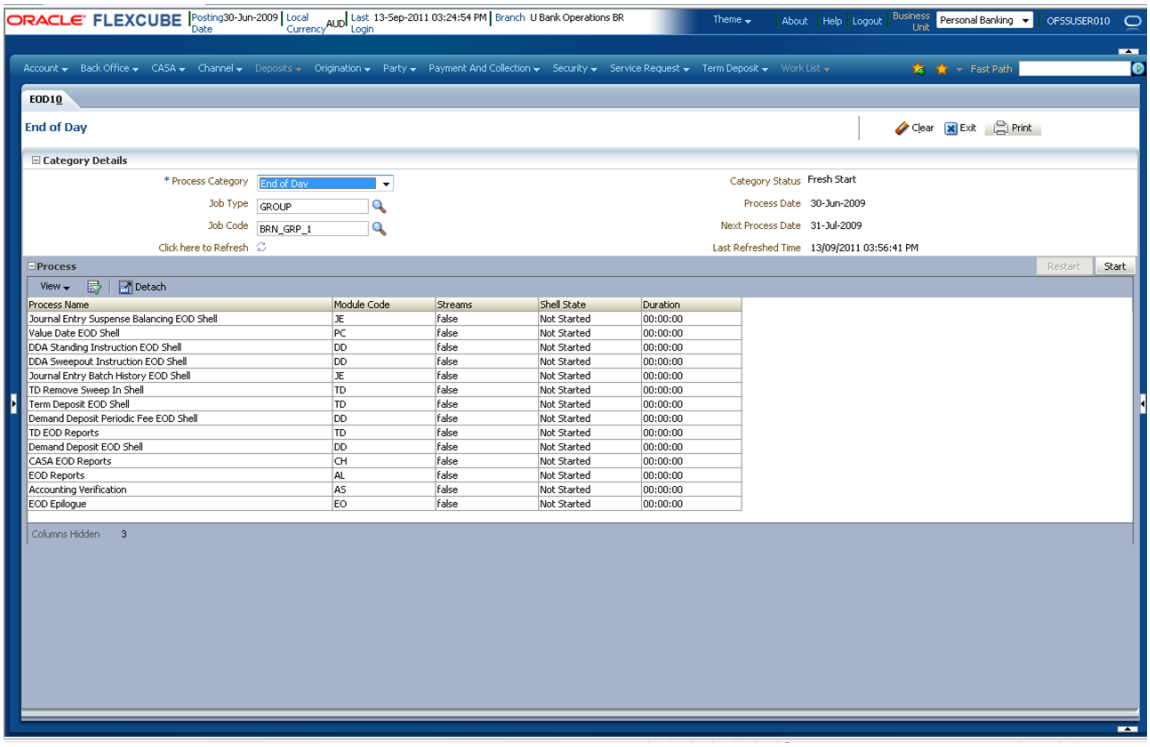
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started.*

   The **Category Status** is *Fresh Start.*

3. Verify the **Process Date** and the **Next Process Date.**

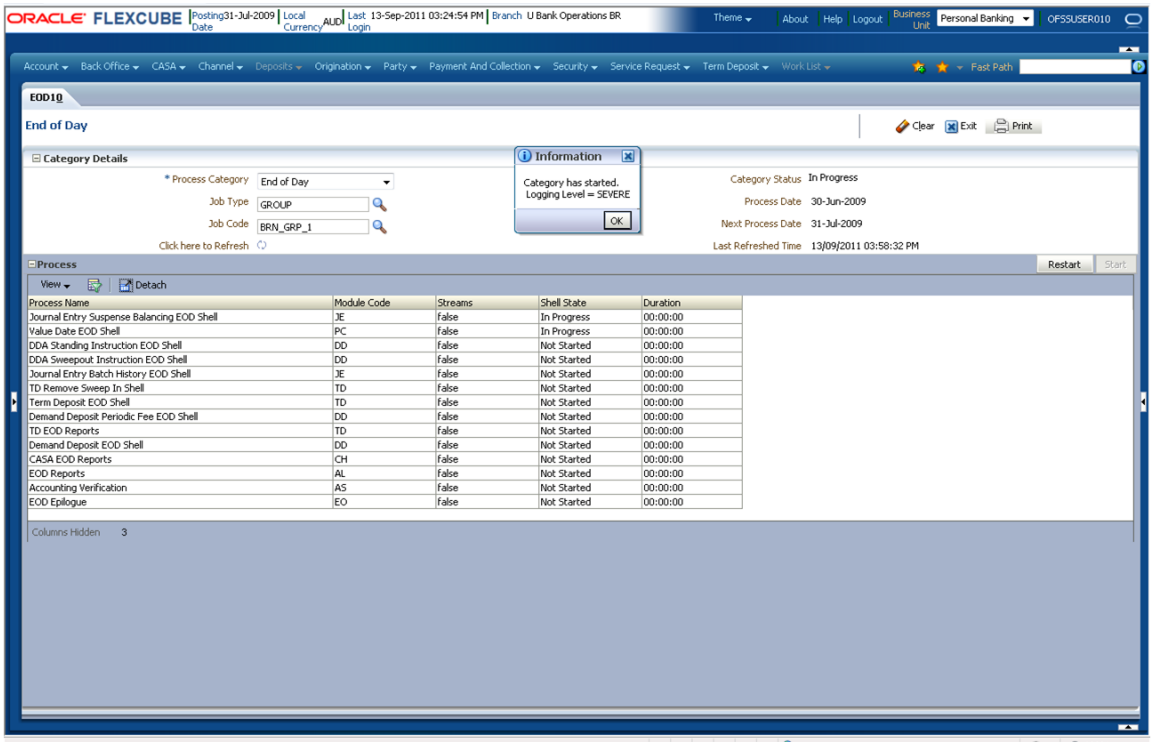
Figure 3–3 End of Day



4. Click the **Start** button to begin the execution.

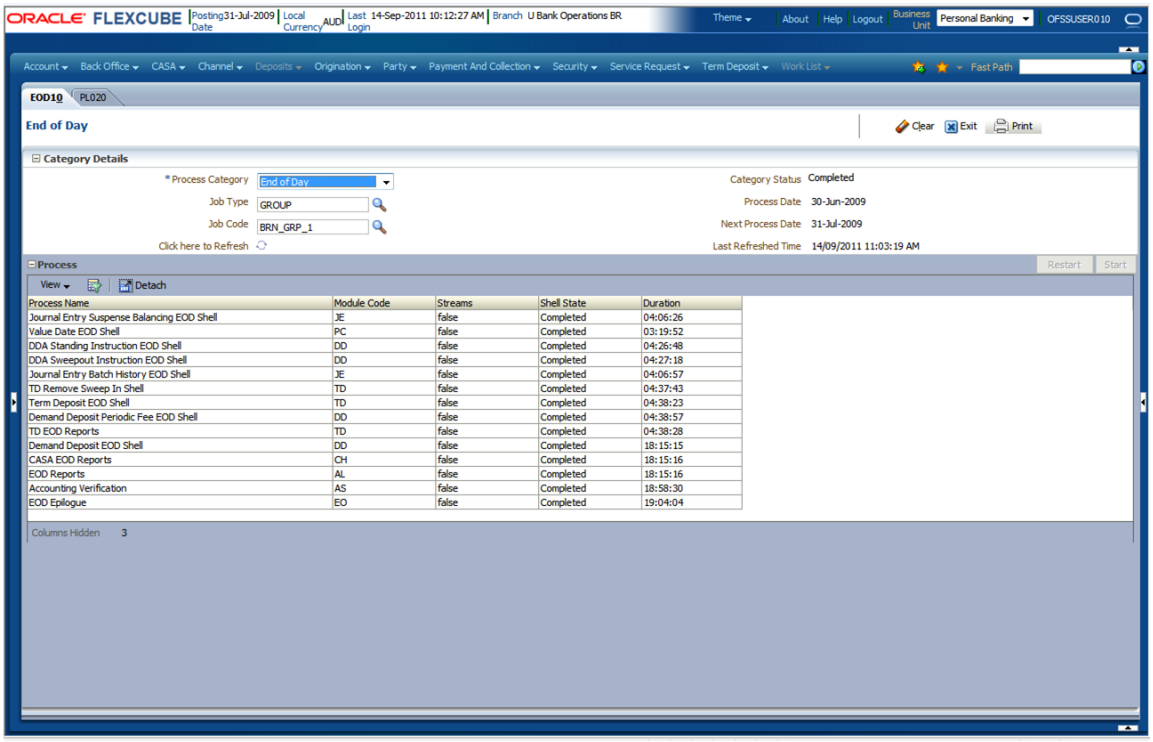
   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress*.

Figure 3–4 End of Day in Progress



5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed*.

Figure 3–5 End of Day Process Completed



3.1.5 Internal System EOD Category Execution

To execute the Internal System EOD category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	Internal System EOD
Job Type	GROUP
Job Code	BRN_GRP_1

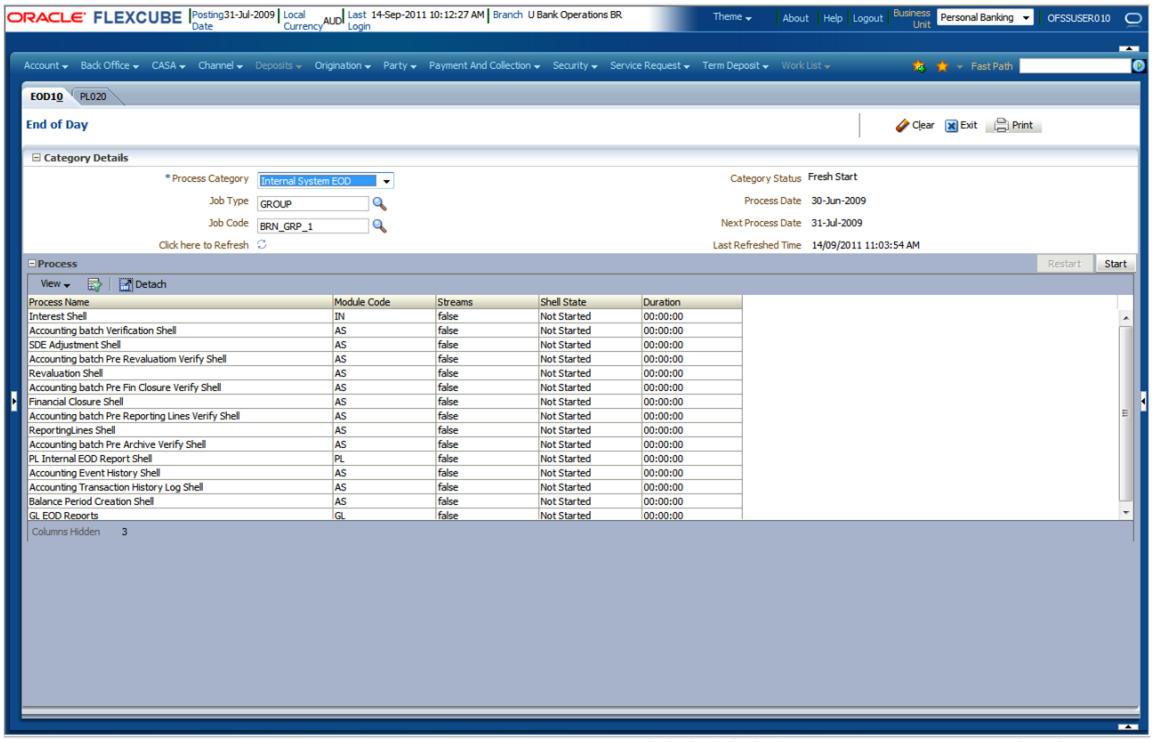
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started.*

   The **Category Status** is *Fresh Start.*

3. Verify the **Process Date** and the **Next Process Date.**

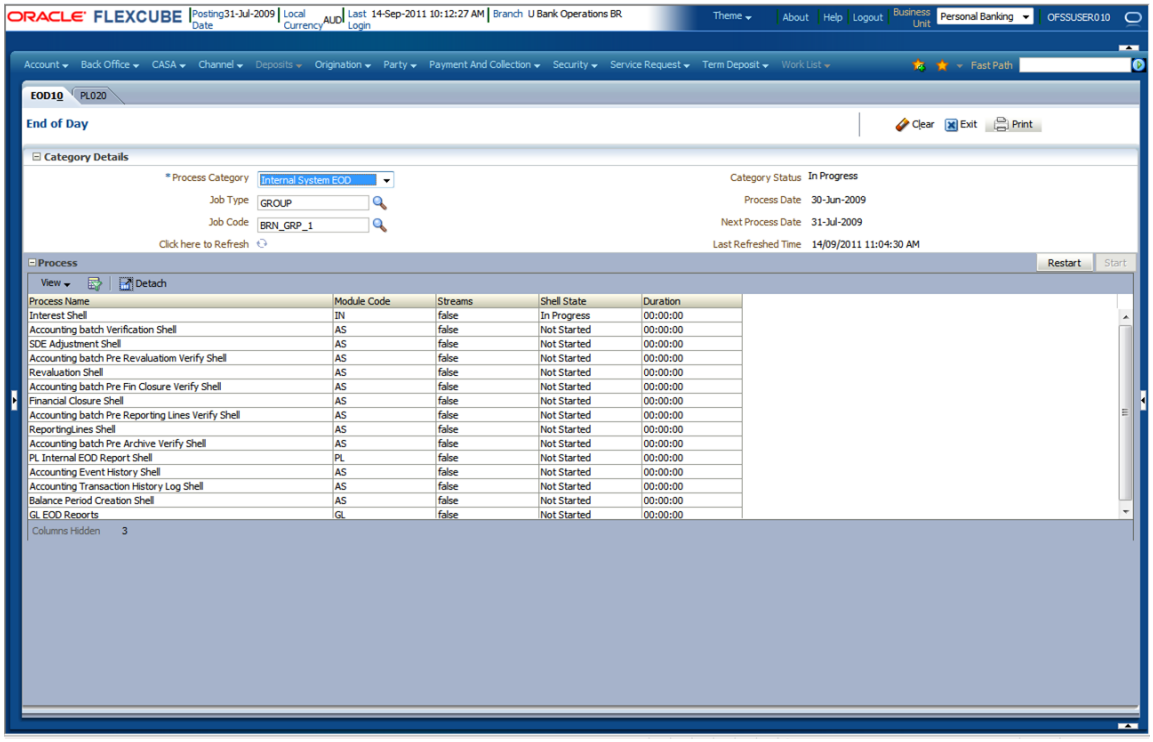
Figure 3–6 Internal System EOD



4. Click the **Start** button to begin the execution.

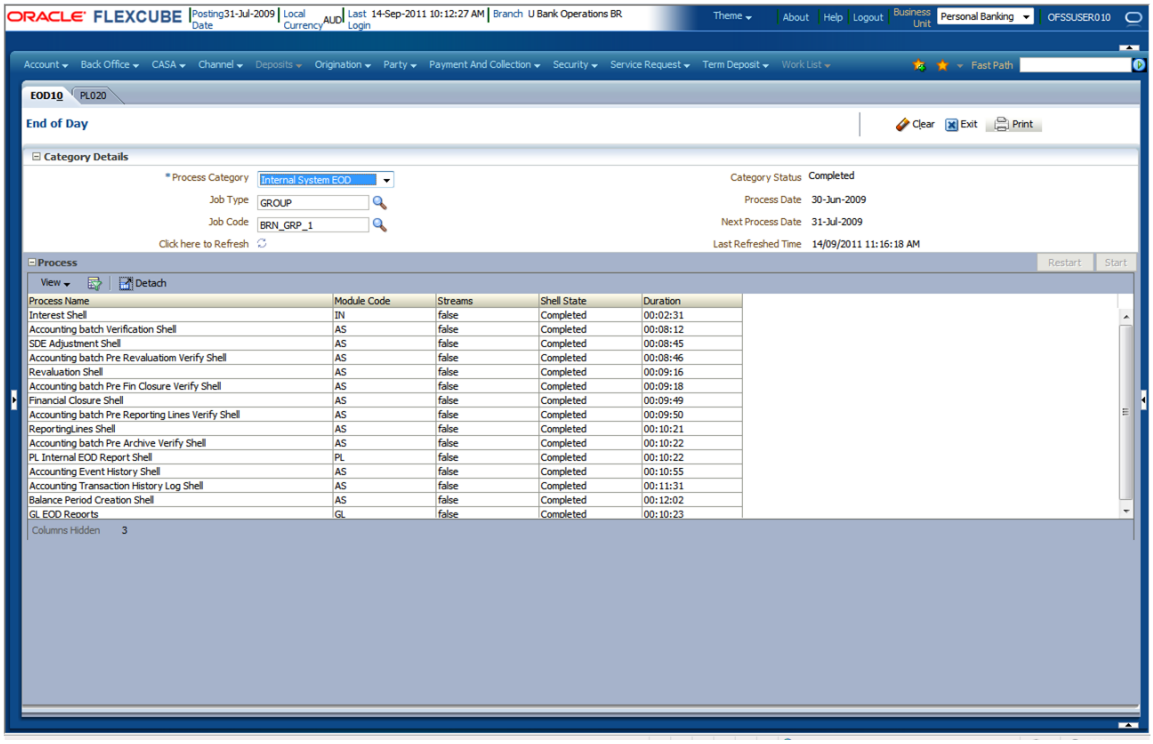
   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress.*

Figure 3–7 Internal System EOD in Progress



5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed*.

Figure 3–8 Internal System EOD Completed



3.1.6 Common Staging Area (CSA) Handoff Category Execution

To execute the Common Staging Area (CSA) Handoff category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	Common Staging Area Handoff
Job Type	GROUP
Job Code	BRN_GRP_1

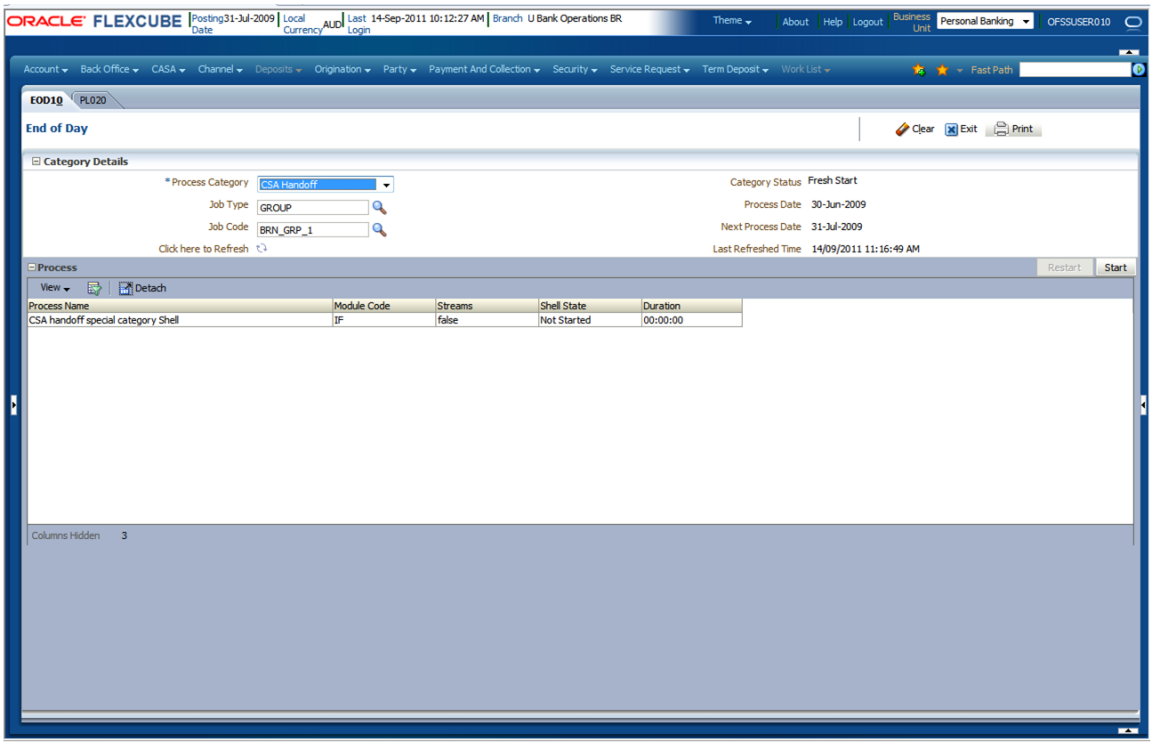
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started.*

   The **Category Status** is *Fresh Start.*

3. Verify the **Process Date** and the **Next Process Date.**

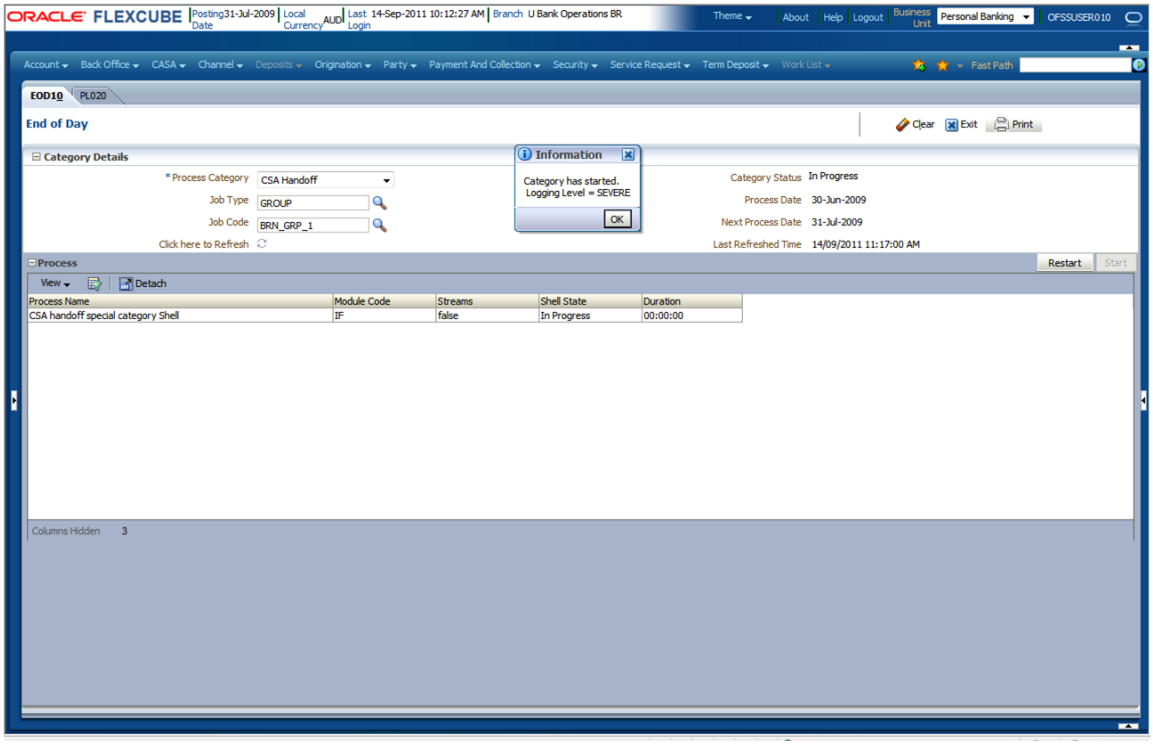
Figure 3–9 Common Staging Area (CSA) Handoff



4. Click the **Start** button to begin the execution.

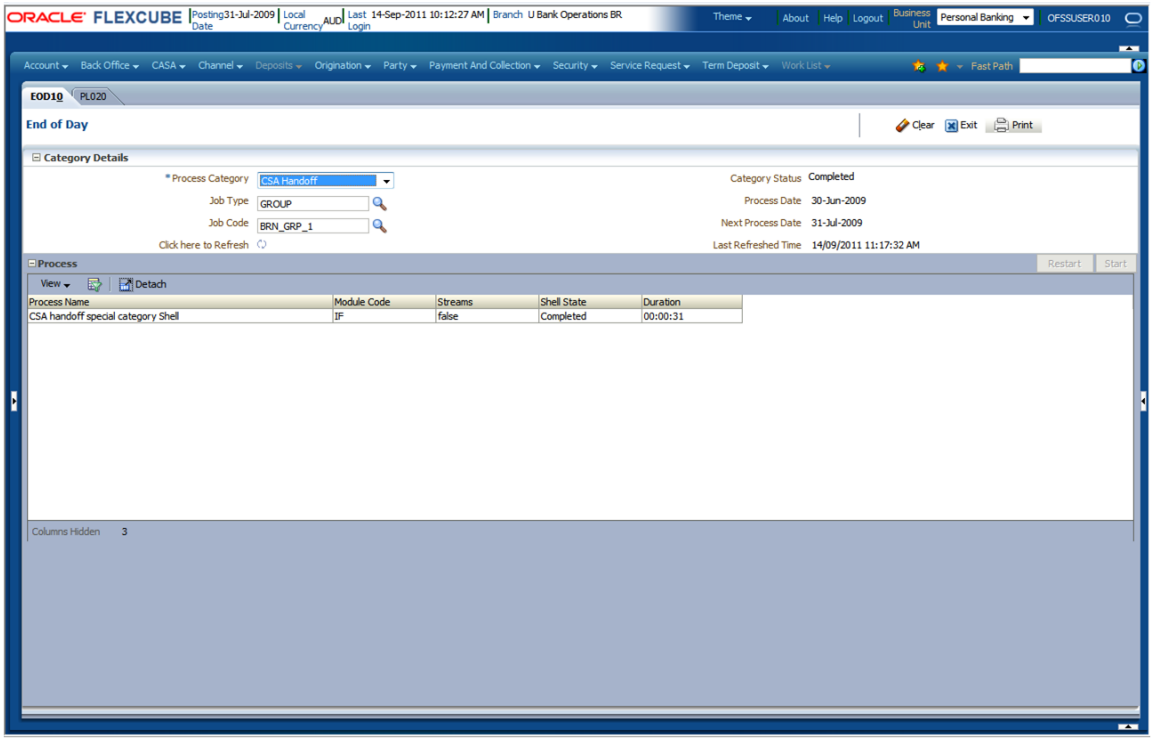
   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress.*

Figure 3–10 CSA Handoff in Progress



5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed*.

Figure 3–11 CSA Handoff Completed



3.1.7 Beginning of Day (BOD) Category Execution

To execute the Beginning of Day category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	Beginning of Day
Job Type	GROUP
Job Code	BRN_GRP_1

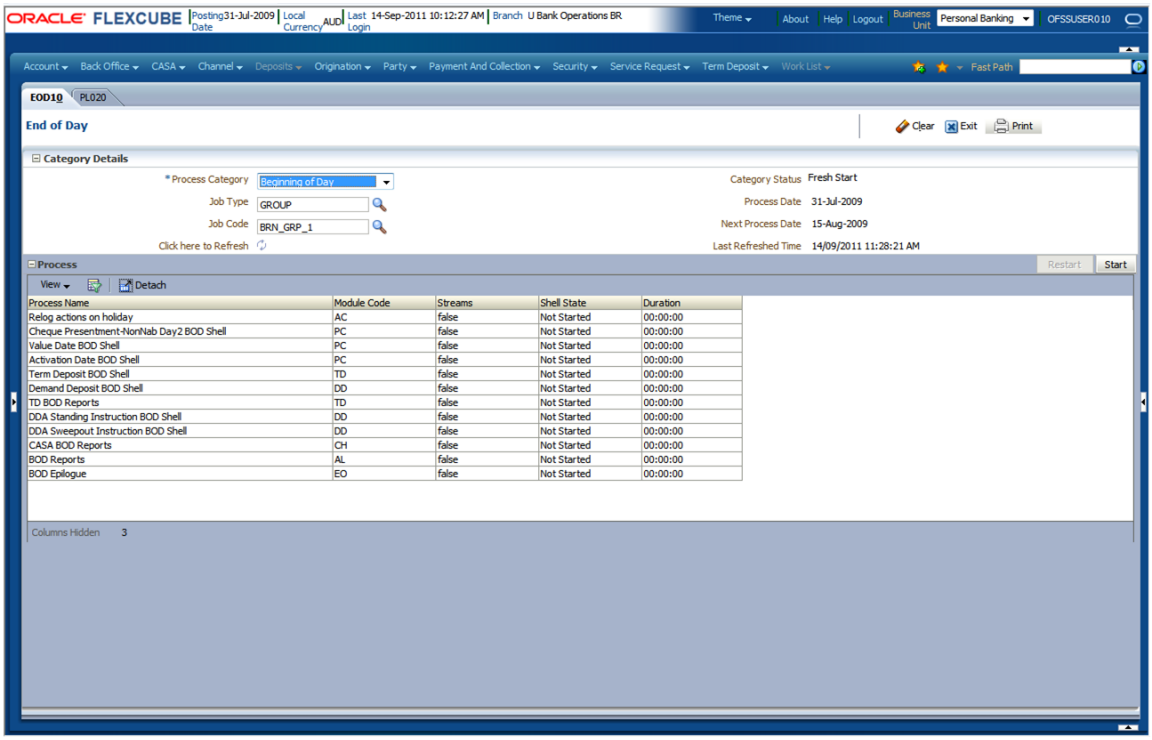
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started.*

   The **Category Status** is *Fresh Start.*

3. Verify the **Process Date** and the **Next Process Date.**

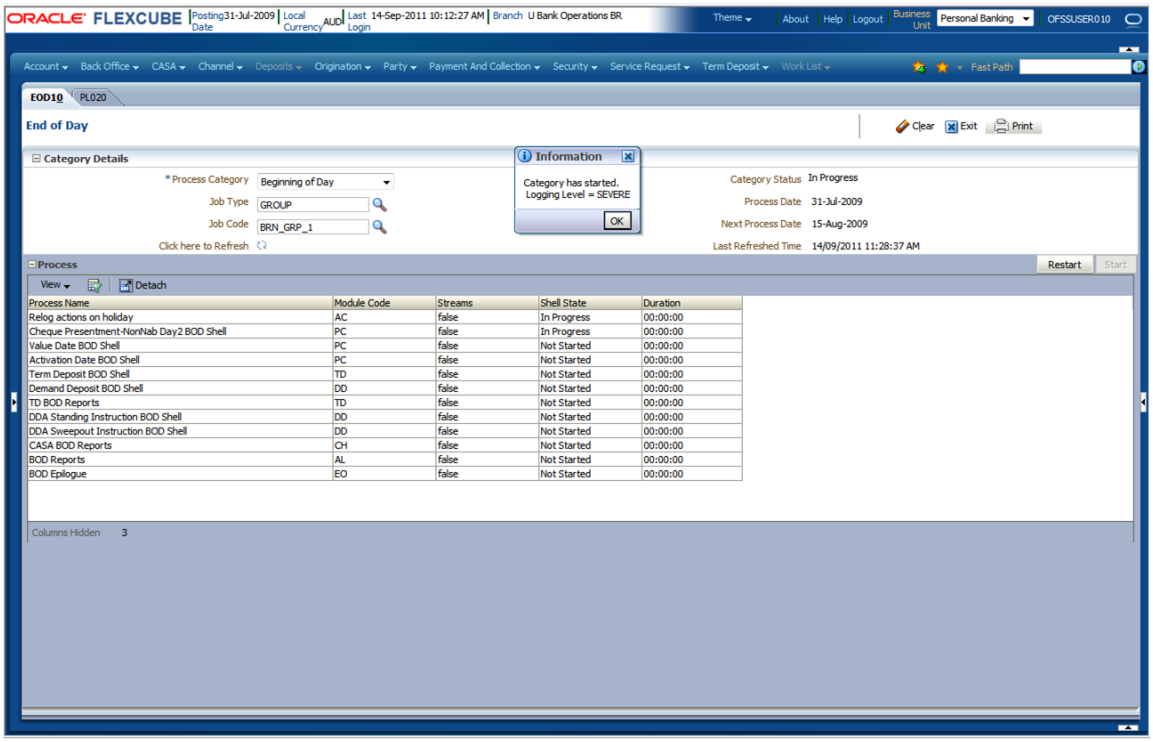
Figure 3–12 Beginning of Day



4. Click the **Start** button to begin the execution.

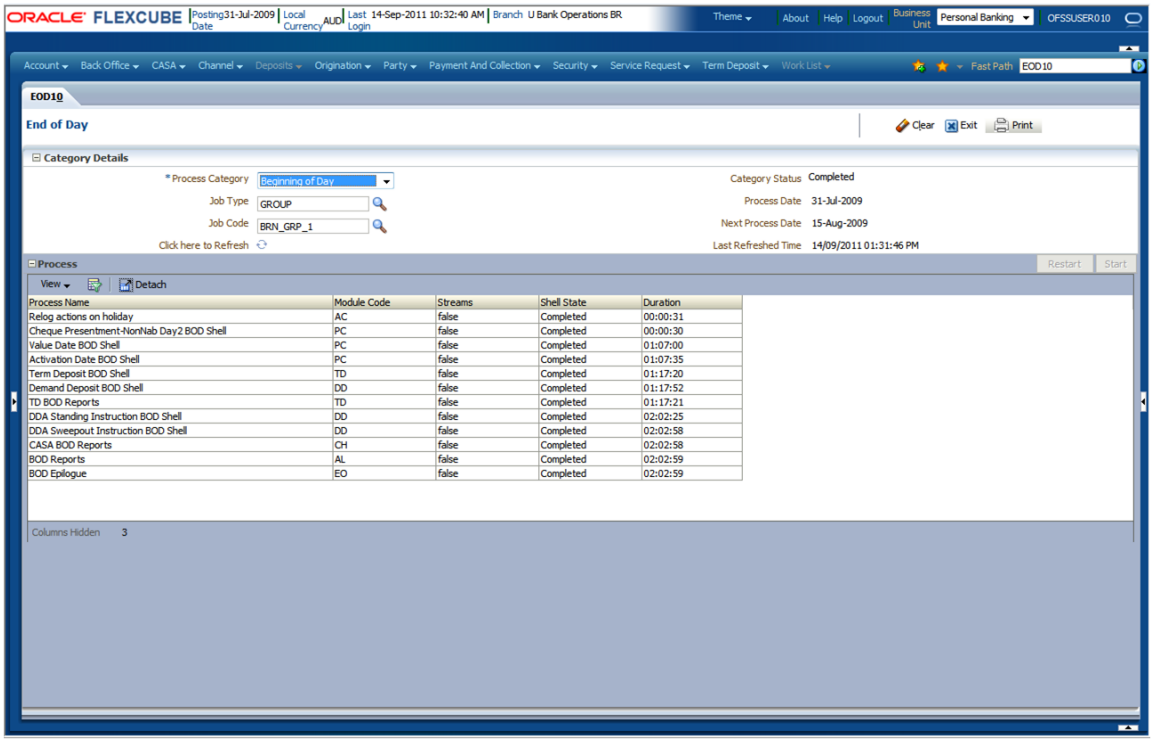
   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress.*

Figure 3–13 Beginning of Day in Progress



5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed*.

Figure 3–14 Beginning of Day Completed



3.1.8 Housekeeping Category Execution

To execute the Housekeeping category:

1. Select the relevant **Category Details** as shown in the table below:

Process Category	Housekeeping
Job Type	GROUP
Job Code	BRN_GRP_1

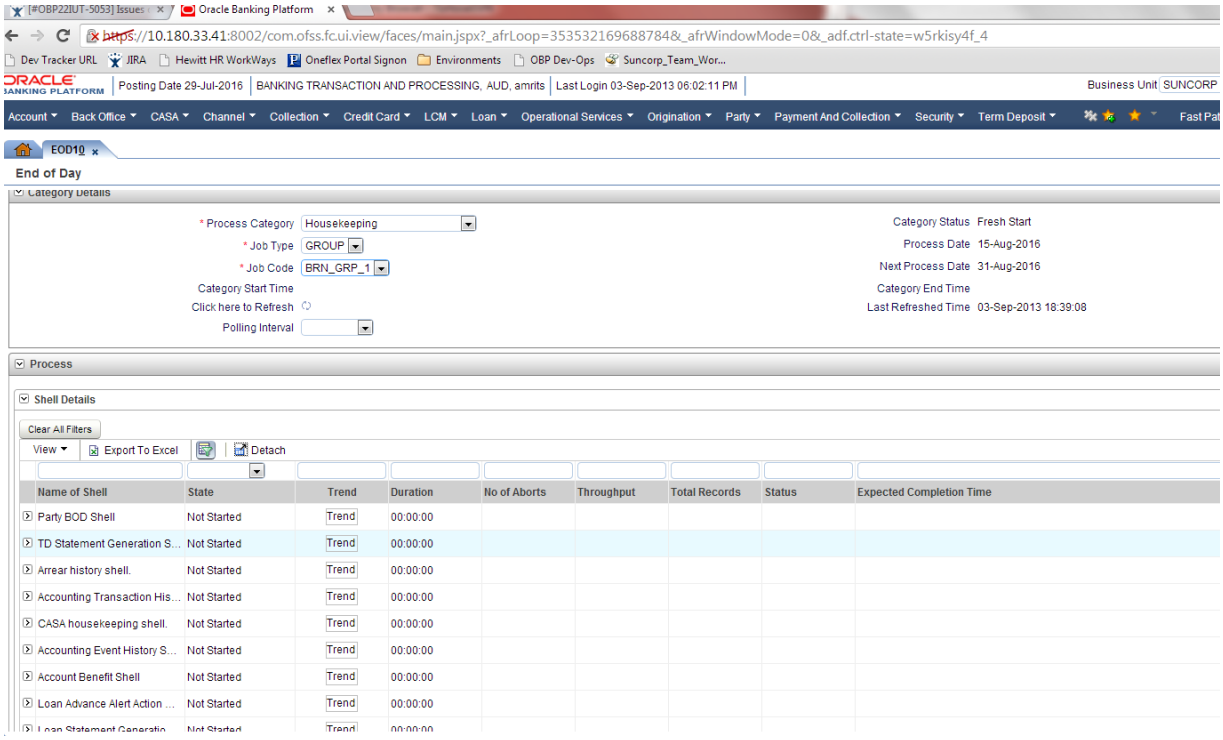
2. Click the **Refresh** button. The rest of the **Category Details** and the **Process Details** appear.

   Here, the **Shell State** is *Not Started.*

   The **Category Status** is *Fresh Start.*

3. Verify the **Process Date** and the **Next Process Date.**

Figure 3–15 Housekeeping

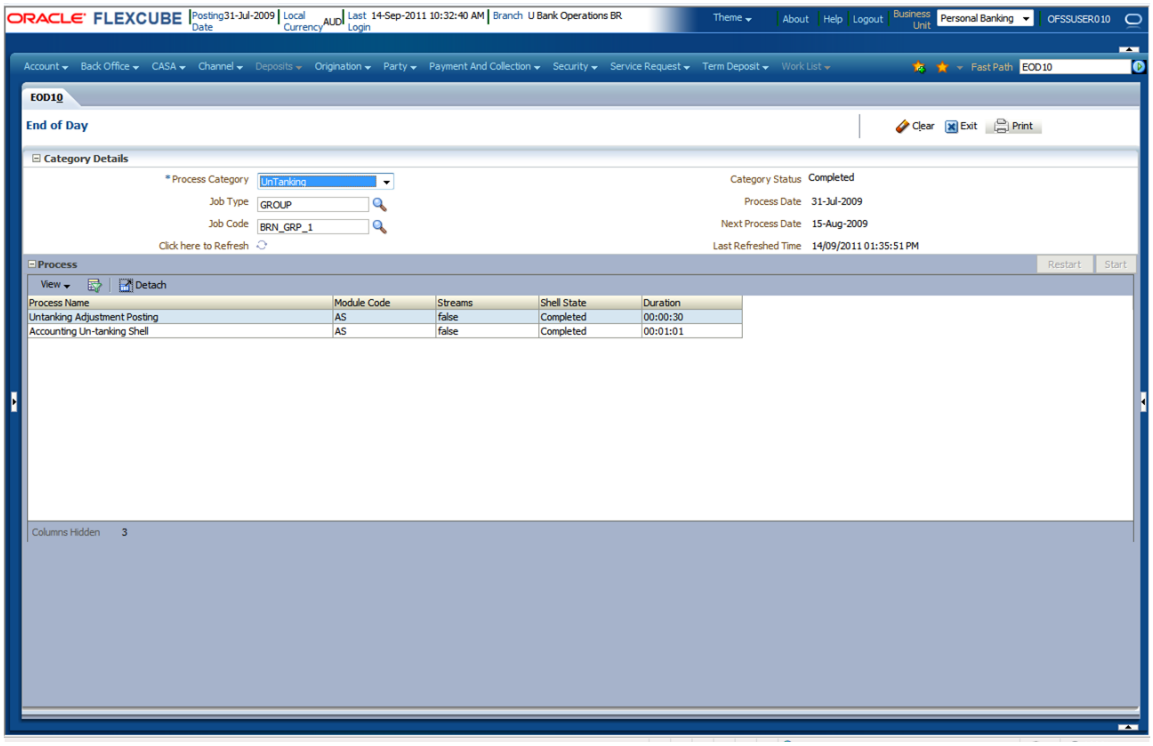


4. Click the **Start** button to begin the execution.

   Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress.*

5. On completion of the category, the **Category Status** and the **Shell State** of all the processes display *Completed.*

Figure 3–16 Housekeeping Completed



4

Setting Up The Bank And Branch

This chapter provides the process of setting up the bank and the branch commonly referred to as the Day 0 setups. It gives the details of the Administration User Interface (UI) offered by Oracle Banking Channels Bank User Experience, using which, the administrator can perform a bank and a branch setup.

This chapter includes the following topics:

- Common Services Day 0 Setup
- Product Manufacturing Unit Day 0 Setup
- Product Ledger Day 0 Setup

4.1 Common Services Day 0 Setup

The Common Services setup includes:

- Core Maintenances
- Currency Maintenances
- Calendar Maintenances

4.1.1 Core Maintenances

The Core Entity Services in Oracle Banking Platform, is a common service, which seeks to define the broad parameters within which the rest of the application functions. The service defines the bank, the various modules of the application that the bank may want to introduce, the languages and the time zones it operates in, the core parameters and structures of its various branches. The core entity services are also used by each of the different modules, and provide a variety of support functions to them.

The following Core Maintenances must be completed as a part of bank and branch setup:

- Bank Codes (Fast path: CS01)
- Bank Parameters (Fast path: CS03)
- Branch Parameters (Fast path: CS06)
- Country Codes (Fast path: CS09)
- Financial Cycle (Fast path: CS10)
- Payment Codes (Fast path: CS15)
- Reason Codes (Fast path: CS16)

- State Codes (Fast path: CS17)
- Verification Category (Fast path: CS21)
- Verification Type (Fast path: CS22)
- Verification Checklist Policy (Fast path: CS23)
- Purpose Codes (Fast path: CS24)
- Bank Policy (Fast path: CS26)
- Bank Policy Deviation Definition (Fast Path: CS39)

Note: To view the detailed procedure to be followed in the application page, see its context sensitive help in the application.

4.1.1.1 Head Office Setup

The Head Office branch creation is currently being done via seed data where the Branch Type is HO. Branch Type is a seed table with fixed values for all applicable branch types, that is uploaded to the application from the backend. After the creation of Head Office branch through seed data, you can proceed to create other branches from the application where the Branch Type is shown as a LOV (excluding HO).

The process to set up a head office branch is as follows:

1. Create a new bank code in the application through the page **Bank Codes (Fast path: CS01).**
2. Set up the new bank parameters through the page **Bank Parameters (Fast path: CS03).**
3. Modify the seed data for Branch Type to include the new bank code as HO and run the seed. Currently the seed will be for Bank Code 08. The head office branch is created via this seed data.
4. Proceed to create the other branches through the application using the page **Branch Parameters (Fast Path: CS06)**, that includes all branch types other than HO.

Note: To view the detailed procedure to be followed in the application page, see its context-sensitive help in the application.

4.1.2 Currency Maintenances

The Currency Services are a part of the common services of Oracle Banking Platform and serve to record and retrieve the various currency related information.

The following Currency Maintenances must be completed as a part of bank and branch setup:

- Currency Codes (Fast path: CY01)
- Amount Text (Fast path: CY02)
- Currency Pairs (Fast path: CY03)
- Currency Branch Parameters (Fast path: CY04)
- Currency Denomination (Fast path: CY05)
- Currency Rate Types (Fast path: CY06)

- Exchange Rates (Fast path: CY07)

> **Note:** To view the detailed procedure to be followed in the application page, see its context-sensitive help in the application.

4.1.3 Calendar Maintenances

The calendar services are embedded in the common services of Oracle Banking Platform and serve to record and retrieve the various holidays of the bank in a calendar year.

The following Calendar Maintenances must be completed as a part of bank and branch setup:

- Adhoc Calendar Maintenance (Fast path: CAL01)
- Calendar Maintenance (Fast path: CAL02)

> **Note:** To view the detailed procedure to be followed in the application page, see its context-sensitive help in the application.

4.2 Product Manufacturing Unit Day 0 Setup

The Product Manufacturing Unit (PMU) is a flexible and competent framework for defining products. It has list of tasks to be performed for defining a new product. Each task should be completed parallel or sequentially by respective departments in a bank. It can be mandatory, conditional or non mandatory. The product manufacturing is complete if the respective mandatory tasks are complete.

Prerequisites

Following are the prerequisites for PMU Day 0 Maintenances:

- Interest Rule Definition
- Asset Classification Plan ID
- Dormancy Rule Configuration
- Installment Rule Details
- Loan Schedule Type
- Product Ledger
- Common Services – Purpose Code, Currency Code, Calendar Maintenance, Bank Policy
- Accounting Template
- Facility Category
- Collateral Category
- DMS maintenance
- GAAP Code

Day 0 Maintenances

The following PMU Maintenances must be completed as a part of bank and branch set up:

- Define Party Bank Policy (Fast path: PM001)
- Define CASA Bank Policy (Fast path: PM002)
- Define TD Bank Policy (Fast path: PM003)
- Define Loan Bank Policy (Fast path: PM004)
- Define Bundle Bank Policy (Fast path: PM005)
- Define Hardship Relief Policy (Fast path: PM006)
- Define Statement Policy (Fast Path: PM007)
- Define Dormancy Policy (Fast path: PM008)
- Define Credit Policy (Fast path: PM009)
- Define Product Group (Fast path: PM010)
- Define Interest Rule (Fast path: PM011)
- Define TD Interest Payout Plan (Fast path: PM012)
- Define CASA Product (Fast path: PM013)
- Define Loan Product (Fast path: PM014)
- Define TD Product (Fast path: PM015)
- Define Credit Card Product (Fast path: PM016)
- Define Product Interest Linkage (Fast path: PM017)
- Define Product Branch Restriction (Fast path: PM018)
- Define Bonus Interest Parameters (Fast path: PM019)
- Define CASA Offer (Fast path: PM020)
- Define TD Offer (Fast path: PM021)
- Define Loan Offer (Fast path: PM022)
- Define Credit Card Offer (Fast path: PM023)
- Define Offer Eligibility (Fast path: PM024)
- Link Offers for Principal Offset Facility (Fast path: PM025)
- Define Offer Bundle (Fast path: PM026)
- Define Transaction Restriction (Fast path: PM027)
- Define Domain Category Settlement Mode (Fast path: PM030)
- Define Product Settlement Mode (Fast path: PM031)
- Define Product Group Accounting Entry Template (Fast path: PM032)
- Define Product Accounting Template (Fast path: PM033)
- Define Product Group Role Mapping (Fast path: PM034)
- Define Product Account Role Mapping (Fast path: PM035)
- Define Accounting Class (Fast path: PM036)
- Define Domain Category Accounting Template (Fast path: PM037)
- Define Domain Role Mapping (Fast path: PM038)

Note: To view the detailed procedure to be followed in the application page, see its context-sensitive help in the application.

4.3 Product Ledger Day 0 Setup

The Product Ledger module is supported by Module accounting, Domain Accounting, and Accounting services.

Module accounting handles transaction initiation, raises accounting event, and updates the customer account balances and Overdraft limits, and invokes account services.

Domain Accounting provides the services such as input, authorize, delete, and reverses to the modules to enable the module to initiate appropriate action on the transactions. Domain accounting also validates data and lookup accounting template, builds domain entries, and performs currency conversions.

Accounting services pick up the entries formed by the domain accounting and perform GAAP accounting, netting, currency position, Inter Branch entries, tanking of unauthorized transactions, suspense posting, generation of P&L entries for year end, and hand off data to product ledger.

The following Product Ledger Maintenances must be completed as a part of bank and branch setup:

- System Defined Elements Maintenance (Fast path: PL005)
- Accounting Configuration (Fast path: PL019)
- GAAP Summary (Fast path: PLS001)
- Bank Parameter Summary (Fast path: PLS002)
- Branch Parameter Summary (Fast path: PLS003)
- SDE Number Range Maintenance (Fast path: PLS006)
- SDE Class Maintenance (Fast path: PLS007)
- Product Ledger Summary (Fast path: PLS008)
- Ledger Additional Detail Summary (Fast path: PLS009)
- Product Ledger Group Summary (Fast path: PLS010)
- Inter Branch Parameter Summary (Fast path: PLS014)

Note: To view the detailed procedure to be followed in the application page, see its context-sensitive help in the application.

5

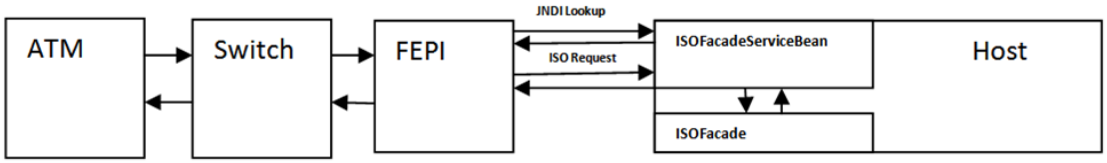
Setting Up Channels

This chapter describes the process of setting up the channels.

5.1 Channel Setup

This section explains the procedure to set up Channels for Oracle Banking Platform. The Figure 5–1 gives an overview of the transaction message flows.

Figure 5–1 Transaction Message Flows



The Channels setup involves three main steps as follows:

Step 1 Front End Processing Interface (FEPI) Application Installation

Install the FEPI application on the host server as a separate application, which will listen to any incoming request from the switch. There is separate FEPI application for ATM and POS.

Step 2 ATM or POS Bank Parameter Maintenance (Fast Path: ATM001)

Log in to Oracle Banking Platform and define the Bank Level parameters for ATM transactions using the **ATM or POS Bank Parameter Maintenance page (Fast Path: ATM001)**.

Following are some useful definitions of fields used in this page:

- **Originating Branch:** The originating branch code refers to the branch through which the ATM transactions of Remote On-us would be routed to respective branches.
- **ATM Offline Limit:** The offline limit is in turn maintained operationally in the Switch. When the ATM is offline, the offline limit is compared with the balance in the PBF file. The amount that is lower on comparison that is, either the PBF balance or the Switch limit is permitted for withdrawal. This functionality of comparison will be only for On-us transactions.

- **Institution ID:** Each code uniquely identifies the bank in the network. Institution ID is allotted by the central bank of the country, or the body that governs payment systems in the country.
- **Institution Name:** The name of the institution.
- **Holds Applicable:**
  - **ATM Debit Hold, POS Debit Hold:** It is applicable for marking hold on debit instructions. Rather than direct debiting the account, hold will be marked and then later after some time amount is debited from account.
  - **ATM Credit Hold:** It is applicable for marking hold on credit instructions.Rather than direct crediting the account, hold will be marked and then later after some time amount is credited to account.
  - **ATM Offline Hold:** It is applicable once ATM becomes online. Rather than direct debiting the account, hold will be marked and then later after some time amount is debited from account.
- **Forwarder Institution ID:** It identifies the network or Institution forwarding the request or advice message in an interchange system. Forwarder Institution ID is allotted by the central bank or the governing body for non-branch channels to Master Card, VISA, cash net. In case of tie-up, other bank's Acquirer Institution ID will be Forwarder Institution ID.
- **Forwarder Cash GL:** This GL account will be credited or debited for remote-on us transactions and off-us transactions done through the Bank's ATM.
- **Forwarder Fee GL:** This GL account will be credited for the transactions performed by other bank's customers through ATM, owned and managed by our bank.

Note: To view the detailed procedure to be followed in the application page **ATM or POS Bank Parameter Maintenance page (Fast Path: ATM001)**, see its context-sensitive help in the application.

Step 3 ATM or POS Terminal Setup (Fast Path: ATM002)

Set up individual ATM or POS terminal for the bank for transactions using the ATM or POS Terminal Setup (Fast Path: ATM002) page.

Following are some useful definitions of fields used in this page:

- **Card Acceptor Terminal ID:** User inputs the code which uniquely identifies a terminal at the card acceptor location. It is maintained for both ATM and POS machines.
- **Acquirer Institution ID:** Identifies the acquirer institution ID associated with the ATM/ POS terminal.
- **Forwarder Institution Supported:** Identifies the forwarder Institution ID to be supported for this ATM/POS terminal.
- **Branch Code:** The originating branch code refers to the branch through which the ATM transactions would be routed to respective branches. Originating Branch handles Cash Funding (To/ From ATM) and Cheque Deposited at ATM.
- **POS Terminal GL:** This GL account will be credited or debited for cash transactions done through the Bank's POS terminal.

- **ATM Cash GL:** This GL account will be credited or debited for cash transactions done through the Bank's ATM terminal by the bank customers.

> **Note:** To view the detailed procedure to be followed in the application page **ATM or POS Terminal Setup (Fast Path: ATM002)**, see its context-sensitive help in the application.

5.2 Front End Processing Interface (FEPI)

This section contains information related to FEPI.

5.2.1 Installation

FEPI is installed in the same location as Oracle Banking Platform (/scratch/app/product/fmw/obpinstall/obp). Following are the steps for FEPI installation:

1. Create a directory structure for FEPI similar to the OBP environment.
2. Change the paths in FEPI start scripts as per the environment, **start_fepi_atm.sh** and **start_fepi_pos.sh** at /fepi/scripts.
3. Change the following property values in **channels_atm.properties** and **channels_pos.properties** as per the environment:
   - **BANK_CODE:** Indicates the bank code

     For example, BANK_CODE=335
   - **LISTENER_PORT:** The port number on which FEPI server accepts incoming ISO message requests.

     For example, LISTENER_PORT=9999
   - **COMMAND_PORT:** The port number on which FEPI server accepts command message. (Note: Need to specify an available valid port number, so that FEPI starts; it is a feature of native code and currently no messages are sent)

     For example, COMMAND_PORT=9998
   - **FNDI.FJ.java.naming.provider.url:** The IP address and port number on which WebLogic accepts requests

     For example, FNDI.FJ.java.naming.provider.url=t3://10.180.9.108:7001
4. Start ATM and POS FEPI; On prompt enter WebLogic Log in credentials.

   For example, **$ sh start_fepi_atm.sh**
5. After running the **start_fepi_*.sh**, it will prompt for password. This password is the same as the admin password on Weblogic server.

5.2.2 ATM And POS Trace Logs

The Trace logs are available in logs folder, for example, /fepi/logs.

Additionally **fepi-console** and **fepi-ofss** logs are also stored at the above location for ATM and POS FEPI server.

5.2.3 module.channel or cz.module.channel enabling of logs

This is not related to FEPI, and these logs (host logs) are controlled by logging.xml of the WebLogic server.

To understand the logging mechanism, see the OEM Diagnosibility details at http://docs.oracle.com/cd/E25054_01/doc.1111/e24473/diagnosability_adminuser.htm.

5.2.4 Multiple Instances

Currently there are two instances of FEPI, they are ATM and POS. Each instance has a set of individual files along with the common shared files.

For example, for the ATM FEPI server:

File Name	Description
channels_atm.properties	Configuration file
fepi_atm.logging.xml	Logging configuration file
start_fepi_atm.sh	Start script
stop_fepi_atm.sh	Stop script

6

Application Monitoring Using OBP Application

This chapter provides an overview on the various monitoring operations performed as an administrator using Oracle Banking Platform application.

6.1 Dynamic Monitoring Service (DMS)

The aim is to monitor different channels involved in performing transactions with OBP. The monitoring parameters consists of channels, services, trends (current behavior of execution), time metrices etc. The monitoring is performed by DMS (Dynamic Monitoring Service).

What is DMS?

The Oracle Dynamic Monitoring Service (DMS) provides a set of Java APIs that measure and report performance metrics, trace performance and provide a context correlation service for Fusion Middleware and other Oracle products. Along with the APIs, DMS provides interfaces to enable application developers, support analysts, system administrators, and others to measure application-specific performance information.

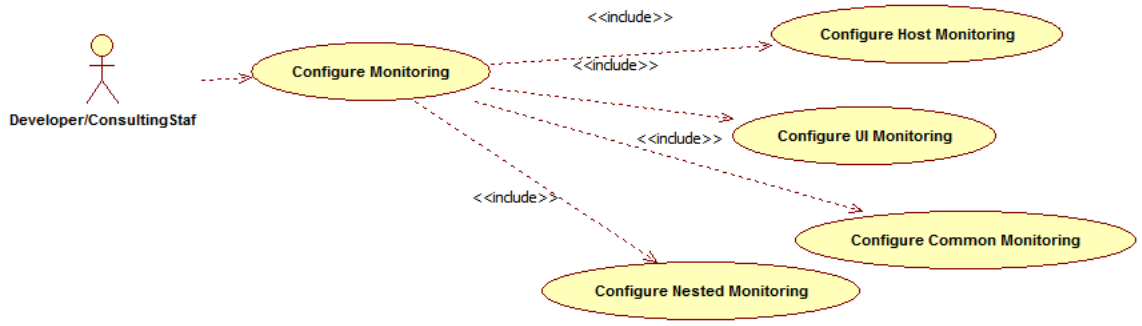
6.1.1 Usage

The usage of DMS is defined by the role of the user. Based on their roles, users can either take part in configuration of services for DMS or monitor the statistics collected via DMS.

Developers

These are the set of people who configure the monitoring services that are the part of OBP system. The configuration can be made either for available services or for new services.

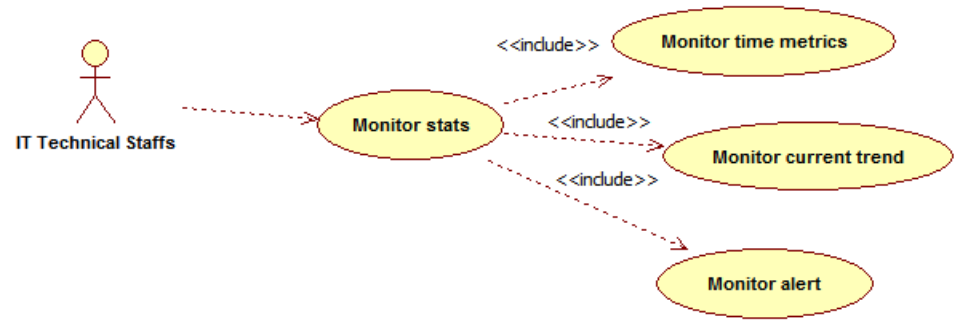
Figure 6–1 Developers



IT Technical Staff

This consists of set of people who monitor the DMS statistics generated for the service. With the help of various metrics generated they can analyze the behaviour of the target service. For example, 'time taken to execute' the service could indicate need of optimization of the service and so on.

Figure 6–2 IT Technical Staff



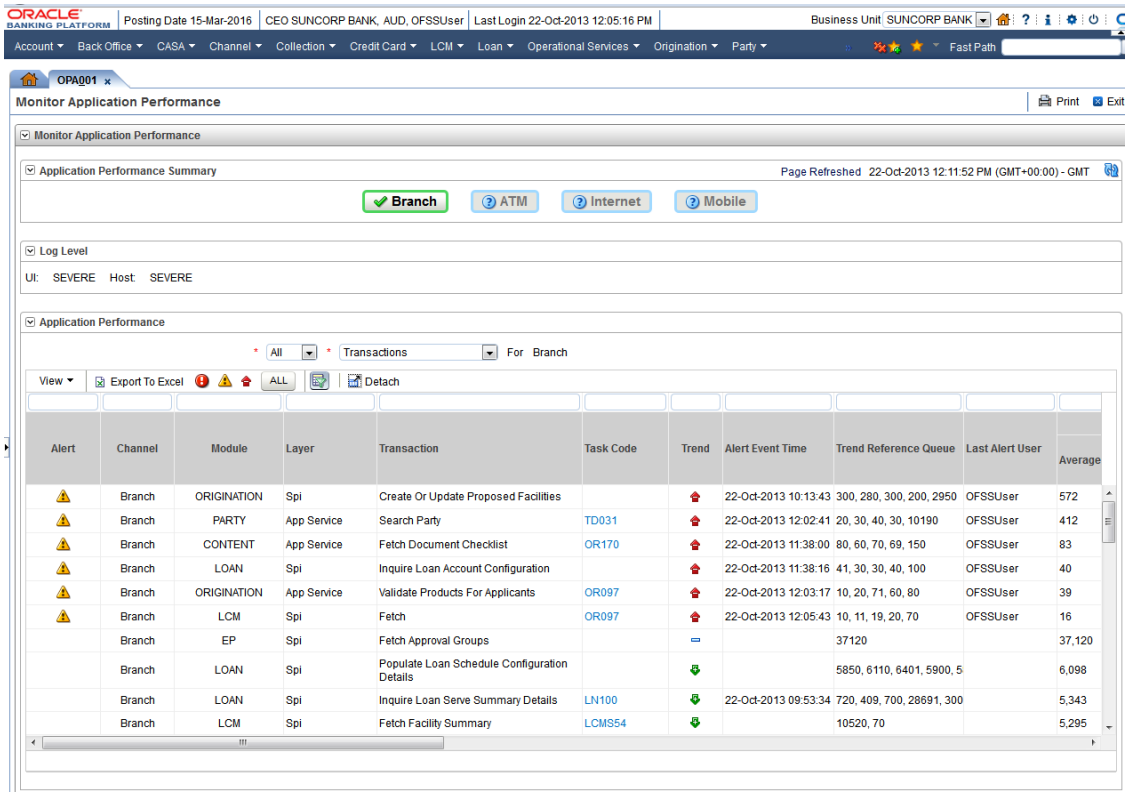
6.1.2 Monitoring Application using the OPA001 page

Once DMS statistics are captured for a particular Channel and transactions involving it, it requires a UI representation to understand the statistics in a readable form so that one can analyse the behaviour. The monitoring activities are mainly carried out by IT Technical staff.

6.1.2.1 Monitoring Application Performance (Fast path: OPA001)

This page gives the monitoring statistics of different channels and the transactions occurring through it. It gives the time metric of the transactions, trend of the current transactions, alert for the channel and so on.

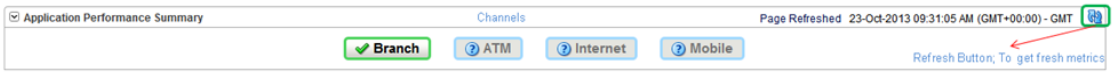
Figure 6–3 Monitoring Application Performance



The overall page can be subdivided in to 3 sub parts on the basis of information they provide:

6.1.2.1.1 Application Performance Summary This section gives the information about the different channels of OBP through which transactions are taking place. The information is about the health, active channels. The 'Refresh Button' on top of this section gets the latest (refreshed) metrics.

Figure 6–4 Application Performance Summary



Following are the few notification about the channels:

- Denotes transactions not present for the Channel
- Denotes normal status that is, the number of alerts are less than the specified limit
- Denotes warning status that is, the number of alerts are in the warning range
- Denotes critical status that is, number of alerts exceeds the limit

6.1.2.1.2 Log Level This section gives logger level information for the host and UI server.

Figure 6–5 Log Level

6.1.2.1.3 Application Performance This section gives the metrics for the transaction. Metrics include timing, alert, trending information. Certain Filters can be applied over the metric table. Initially only 100 (Initial page size which is configurable) transactions are displayed. To display all the transactions, 'ALL' button is to be clicked.

Trend

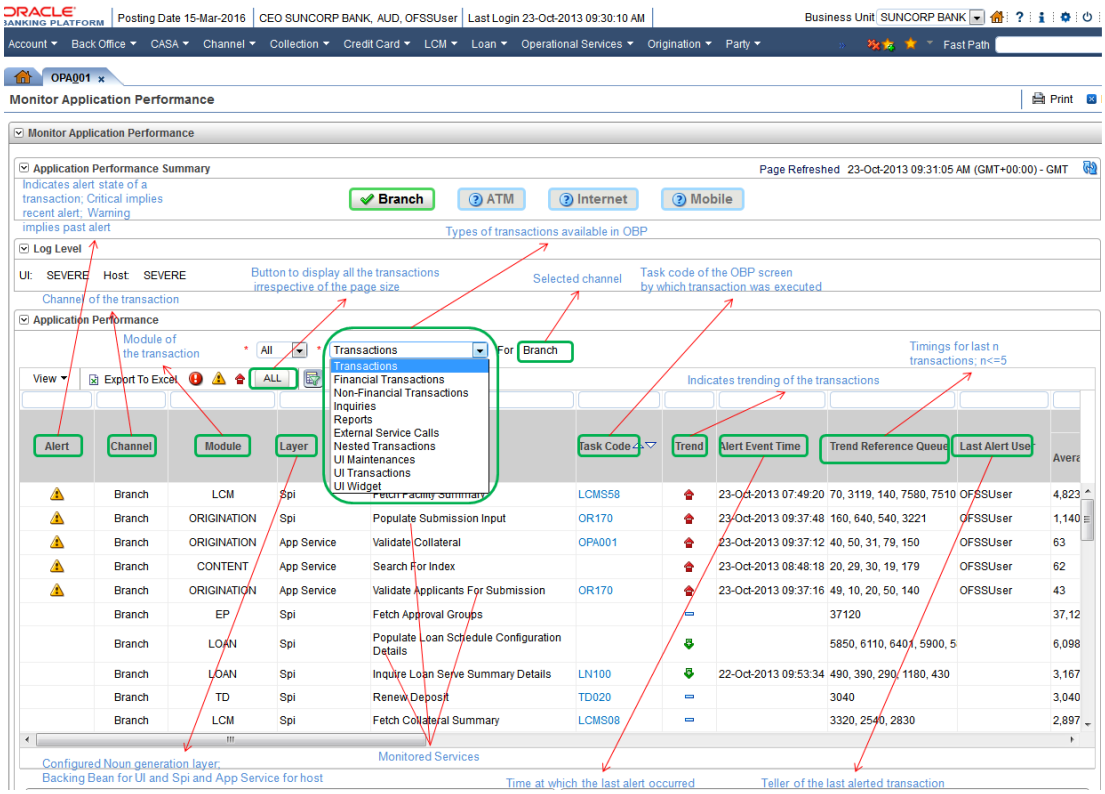
Indicates trending of execution timings of transaction. It is calculated by algorithm namely, Exponential Moving Average where if the execution time goes above the specified limit which is calculated by adding average execution time of the transaction and allowed limit (varies logarithmically to execution time); the transaction is considered as trending upwards and vice-versa for downwards trend.

However, if the execution time is with the range, trend is considered as neutral.

Alert

Indicates alerting state of the transaction. A transaction is given weight based on its properties namely, transaction type, timing category and OBP module. The weight gives the offset allowed for transaction execution time. If the current transaction time is greater than average transaction time + offset, it is marked as alert. Initially it is marked as 'Critical' and after sometime the state is marked as 'Warning'.

Figure 6–6 Alert State



The table below explains each column of the table present in the given snapshot.

Table 6–1 Alert State

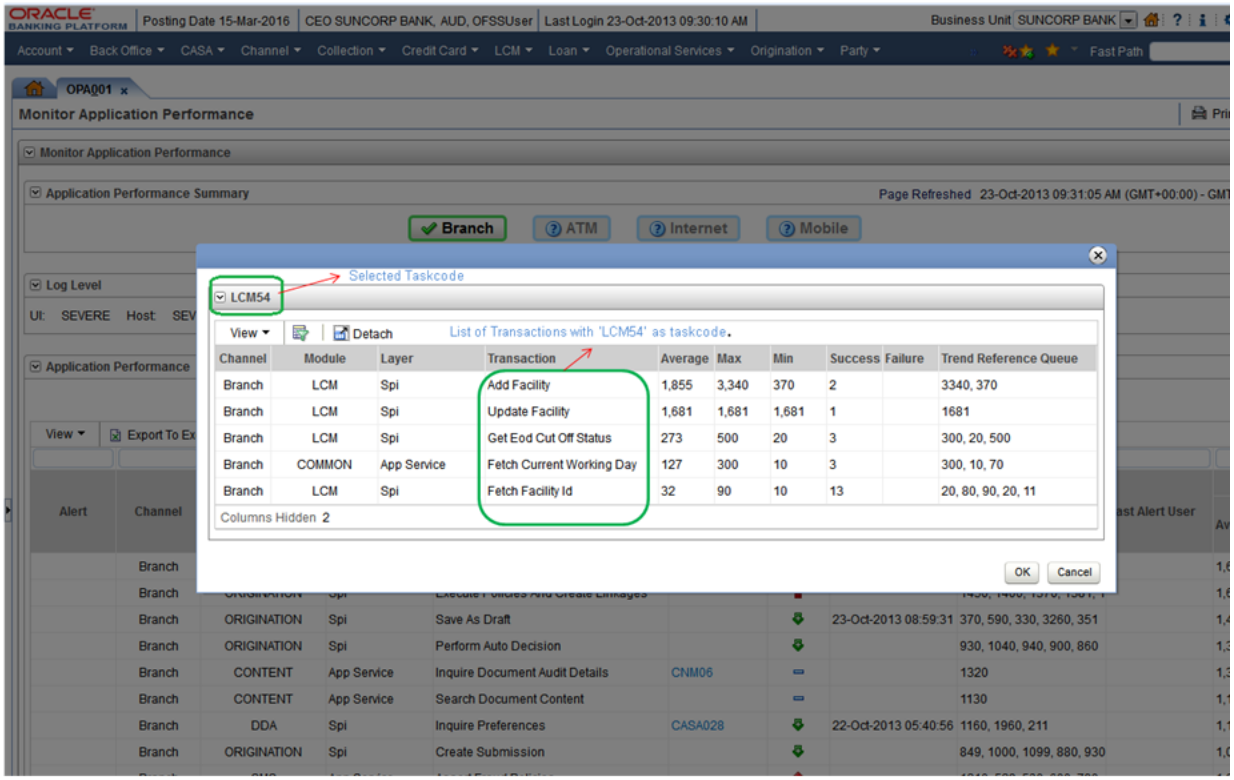
Sr. no.	Column Name	Description
1	Alert	Indicates alert sate of the transaction; Valid Values: BLANK: No alert, Warning: Alert in past (default 5 minutes), Critical: Alerted Transaction
2	Channel	Indicates channel through which the transaction occurred; Valid Values: Branch, ATM, POS and so on
3	Module	Indicates the OBP module of which transaction is a part.
4	Layer	Configured Noun generation layer. Backing Bean for UI and Spi and App Service for Host.
5	Transaction	Indicates the name of the transaction
6	Task Code	Indicates the task code of the OBP page by which the transaction was triggered.
7	Trend	Indicates the trending of transaction. Valid Values: Upwards, Downwards, Neutral
8	Alert Event Time	Time at which last alert occurred for the transaction
9	Trend Reference Queue	Execution time of last n transactions(n=5)
10	last Alert User	Teller who performed the last alerted transaction
11	Average Time	Average execution time
12	Max Time	Maximum time of execution of the transaction
13	Min Time	Minimum time of execution of the transaction
14	Total Time	Total time of execution
15	Success Count	Number of times transaction executed successfully
16	Failure Count	Number of times transaction failed.
17	Debit Amount	Amount debited after transaction
18	Credit Amount	Amount credited after transaction
19	Trend Reference	Execution time of last transaction
20	Alert ECID	ECID of the last alerted transaction
21	Service	Service name of the transaction
22	Completed Operations	Number of completed transactions
23	Max Active Threads	Maximum active threads
24	Host	Host name
25	Server	Server name
26	Process	Process name
27	App Root Type	Root type of noun
28	Failure Security Event	Failure due to security error
29	2FA Event	Authentication failure

Table 6–1 (Cont.) Alert State

Sr. no.	Column Name	Description
30	Failure Database Event	Failure due to database error
31	Failure Technical Event	Failure due to technical error
32	Failure Outbound Event	Failure due to outbound call (call outside OBP)

One can select any of the task code which opens a popup with information about that taskCode only.

Figure 6–7 Select Task Code



Detailed Transaction View

This section gives the detailed view of a selected transaction. The desired transaction can be selected from the table (metric table). Click on any row to display a detailed view of the transaction.

Figure 6–8 Selection of Desired Transaction

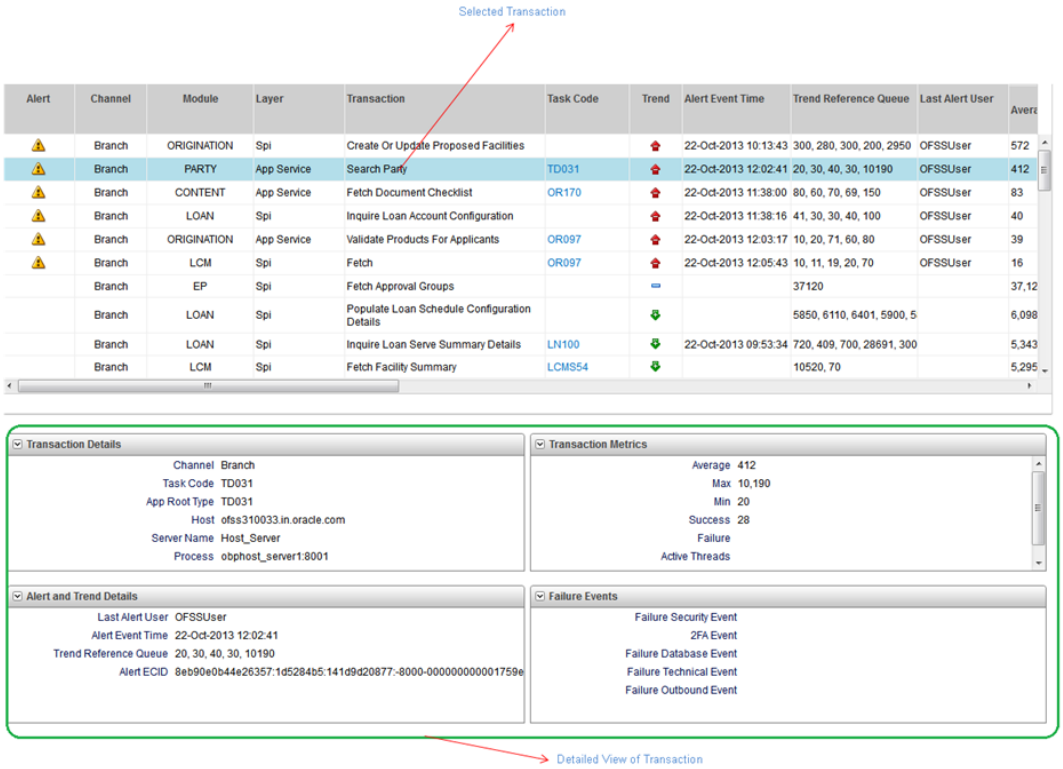


Figure 6–9 Transaction Details

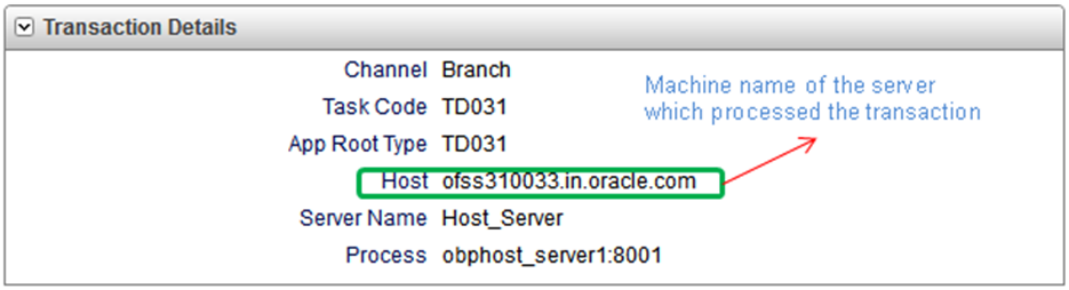


Figure 6–10 Transaction Metrics

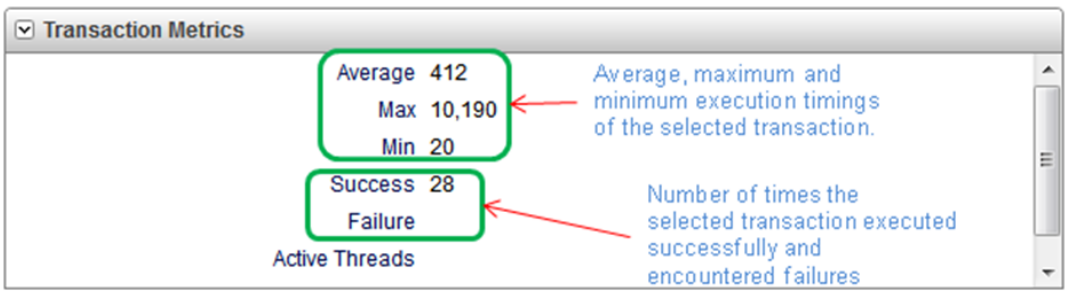
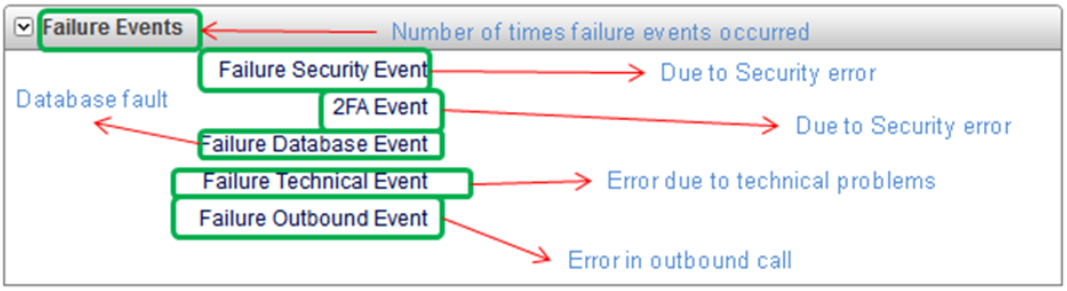


Figure 6–11 Alert and Trend Details

Figure 6–12 Failure Events



Configurations

The below mentioned configurations can be made in DMSConfig.properties:

- **Channel Status:** Number of alerts for which the channel shows 'Critical and 'Warning' status can be configured
- **Alert Status:** The time after which a 'Critical' alert changes to 'Warning' is configurable
- **Initial Page Size:** Every time host data is fetched only rows equal to page size are displayed. The page size is configurable

These configurations can be made in DMSConfig.properties.

6.2 Batch Performance Monitoring

Most of the enterprise applications would require bulk processing of records to perform business operations in real time environments. These business operations include complex processing of large volumes of information that is most efficiently processed with minimal or no user interaction. Such operations would typically include time based events (for example, month-end calculations, notices or correspondence), periodic application of complex business rules processed repetitively across very large data sets (for example, rate adjustments). Batch monitoring includes monitoring of all such batch processes. These batch processes generate huge statistics, which needs to be monitored in order to understand and improve its performance. OPA003 page is used to monitor these processes in detail along various metrices like duration, throughput, aborts, and so on.

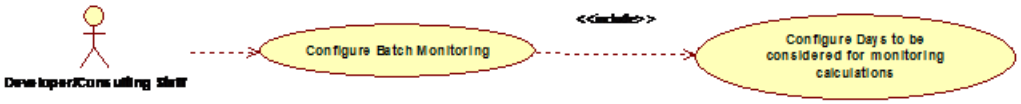
6.2.1 Use Cases

The overall use cases for the whole Batch monitoring operation are divided into two units on the basis of actor that works over batch monitoring operations. The different actors along with their use cases are as below:

Developers

These are the set of people who configure the monitoring services that are the part of OBP system. The configuration can be made in the properties file BatchStatistics.properties. Configuration include the number of previous batch runs to be considered for calculation for monitored metrics.

Figure 6–13 Developers



IT Technical Staff

This consists of set of people who monitor the Batch statistics generated during the batch run.

Figure 6–14 IT Technical Staff

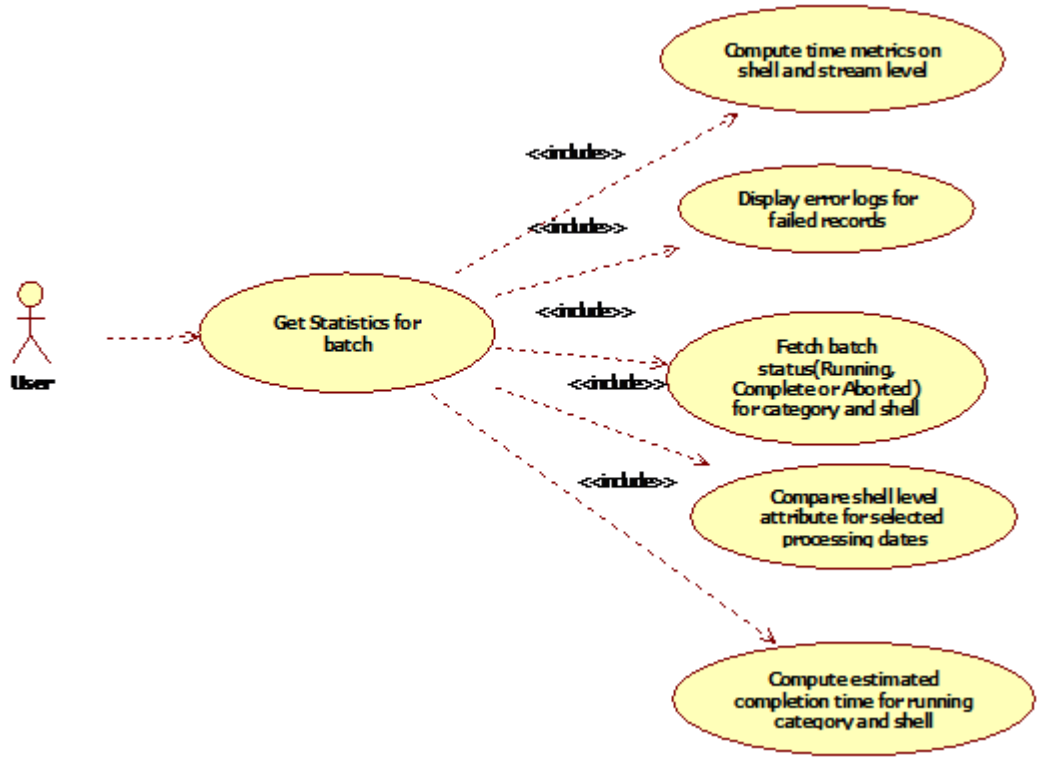


Figure 6–15 IT Technical Staff - Monitor Batch Stats

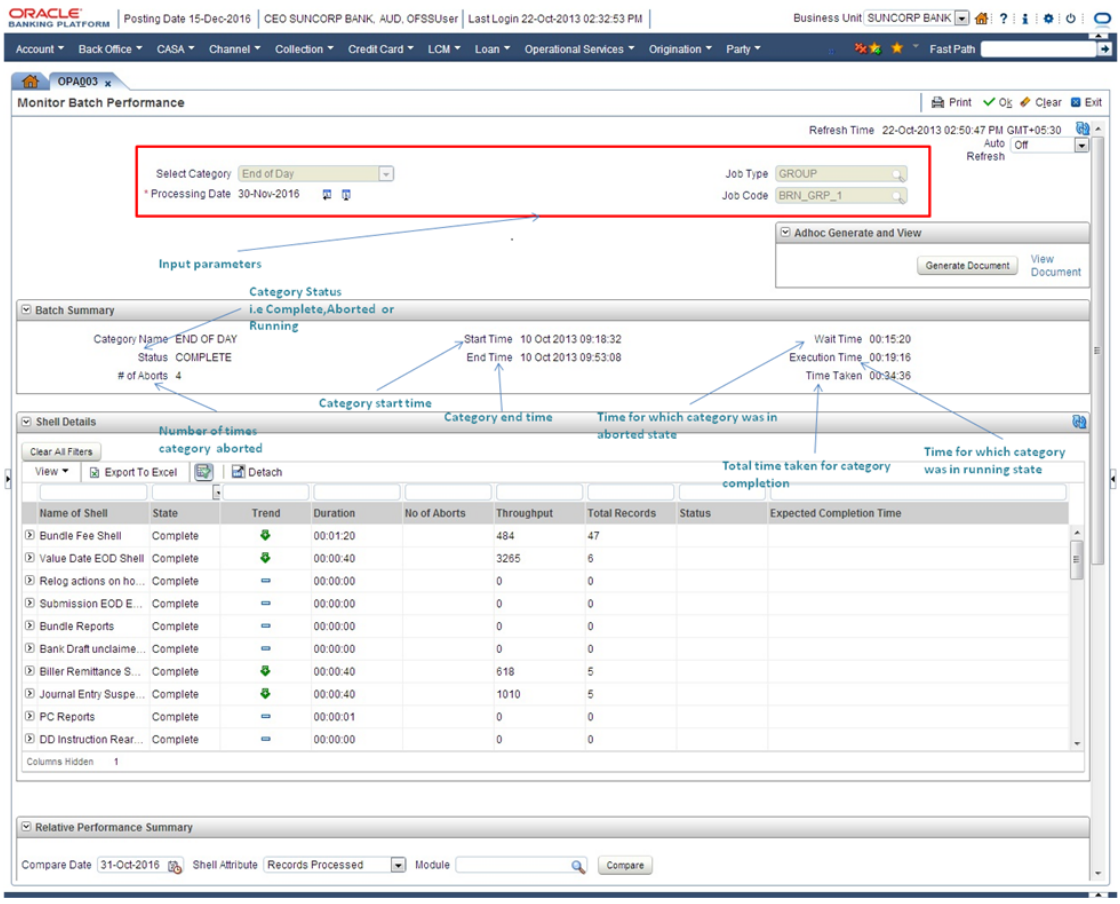
6.2.2 Monitoring Batch Performance Using OPA003 page

Once batch starts it needs UI representation to easily understand and interpret the batch stats. By monitoring these stats, one can understand the bottle necks of the batch process and hence can work in a way to improve batch performance.

6.2.2.1 Monitor Batch Performance (Fast path: OPA003)

This page takes category, job code, job type, and processing date as input and provides monitoring stats for shells running for selected category.

Figure 6–16 Batch Performance Monitoring



The batch statistics are monitored at different levels as follows:

- Category Level
- Shell Level
- Record level

Category Details

The parameter monitored at category level as given as below:

Table 6–2 Category Details

Attribute Name	Description
Category Name	Name of category for example, EOD (End Of Day).
Status	Valid Values: COMPLETE, RUNNING, ABORTED. Indicates status of selected category.
StartTime	Indicates start time of category. The time is represented in DD-MM-YYYY hh:mm:ss format.
EndTime	Indicates end time of category. The time is represented in DD-MM-YYYY hh:mm:ss format.
WaitTime	It is the time for which category is in Aborted state. Wait time for category denotes the time for which batch was halted.
ExecutionTime	It the time for which category is in Running state.
NumberOfAborts	Indicates number of times category was aborted.
Total time	It is the total time taken by category to complete. Time taken for category is summation of wait and execution time.
EstimatedCompletionTime	It is the predicted time for category completion. This time is calculated based on number of incomplete and pending shells for the current running category. The averaged value of previous run duration is considered for calculating estimated time. Similar calculation is done for estimating completion time for shell. This attribute is displayed only during batch run. It is not displayed once batch is complete
TimeStatus	The status of category (that is, delayed or early) denotes whether category is running slow or fast. This value is calculated based on average of historical data. This attribute is displayed only during batch run. It is not displayed once batch is complete

Shell Details

This level displays statistics of all shells corresponding to selected category. The parameters monitored at shell level are given below:

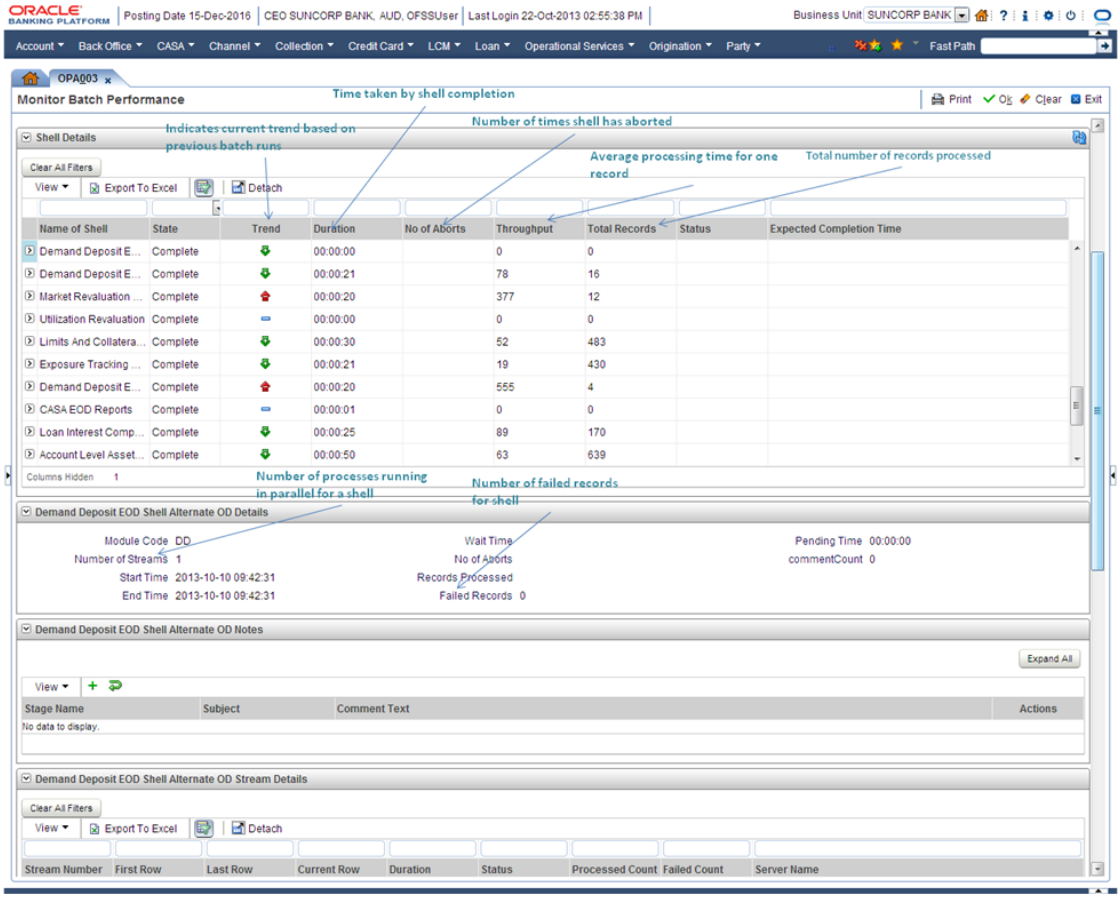
Table 6–3 Shell Details

Attribute Name	Description
Name of shell	Represents name of shell
Trend	Valid Values: UPWARD, DOWNWARD, NEUTRAL. It denotes the trend based on historical data for time required by shells to complete.
Status	Valid Values: Complete, Running, Aborted, Not Started. Indicates status of shell
Duration	It is the time required by shell to complete.
StartTime	Indicates start time of shell. The time is represented in DD-MM-YYYY hh:mm:ss format.
EndTime	Indicates the time at which shell is completed. The time is represented in DD-MM-YYYY hh:mm:ss format.
WaitTime	It is the time for which shell is in aborted state.
ExpectedCompletionTime	Indicates the estimated time for a shell to complete.
FailedRecords	Number of failed records for a shell

Table 6–3 (Cont.) Shell Details

Attribute Name	Description
RecordsProcessed	Number of records processed in a shell
Number of Streams	Number of streams denote number of processes running in parallel for a shell. On proper analysis of historical data of stream count, number of records and duration for particular shell one can optimize throughput for it.
Throughput	It is the average processing time for one record. Throughput is denoted in millisecs.

Figure 6–17 Shell Details

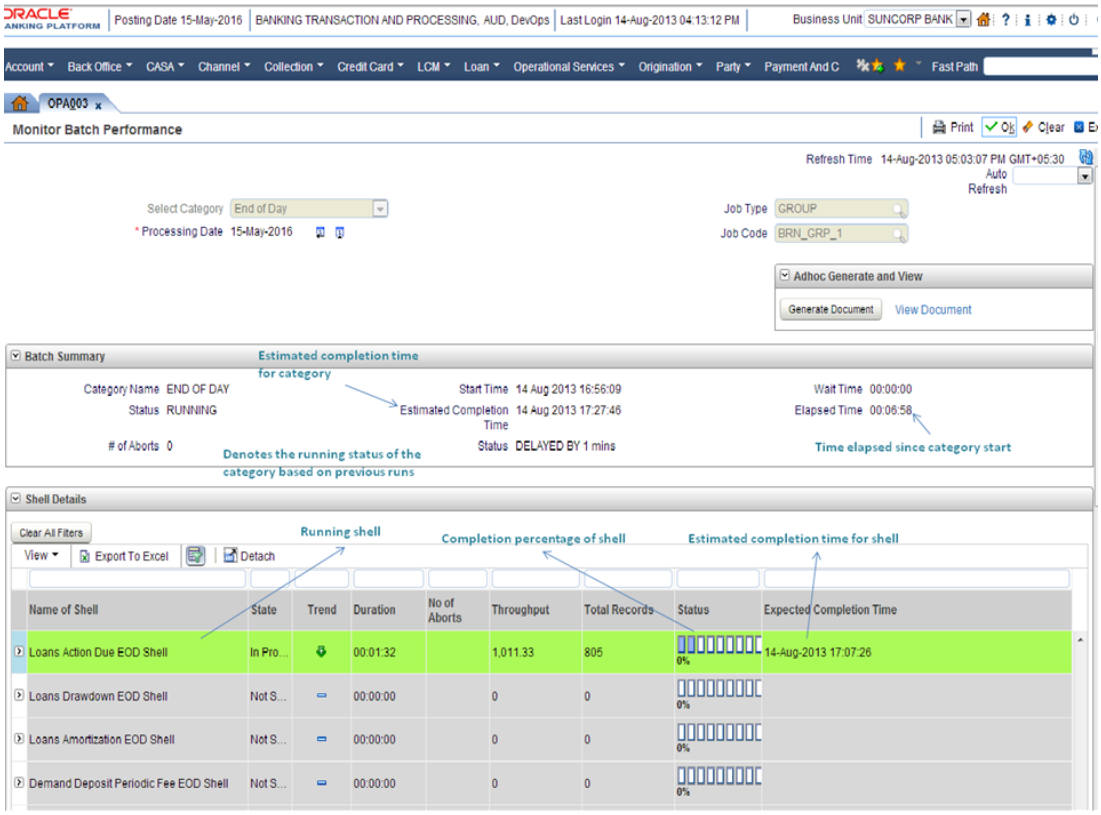


Note: Note the following:

- Trend for a particular shell is decided based on comparison of time statistics (that is, current run time and historical data for previous batch runs). Number of previous batch run to be considered is configurable. It is configured in the property file that is, (BatchStatistics.properties). The trend and other estimated time seems more realistic if number of previous run days configured in property file are more.
- Trend gives an idea whether a particular shell is running fast or slow compared to previous runs though it is important to consider number of records being processed in that shell.

Figure 6–18 shows the view displayed during batch run. Few extra parameters like estimated completion time for shell and category are monitored during batch run.

Figure 6–18 View of Batch Run



Comments Table

Comments Table is rendered based on row click of shell details table.

- In case of batch abort, it is important to know the reason behind abort and how that is fixed. Comments table serves this purpose as one can log the details regarding fix and reason behind shell abort. Multiple comments can be captured for particular shell.
- Also one can query historical data for comments. The historical data of comments can be used to analyse the reason behind failure of particular shell.

Stream Details Table

Stream Details table is rendered based on row click of shell details table.

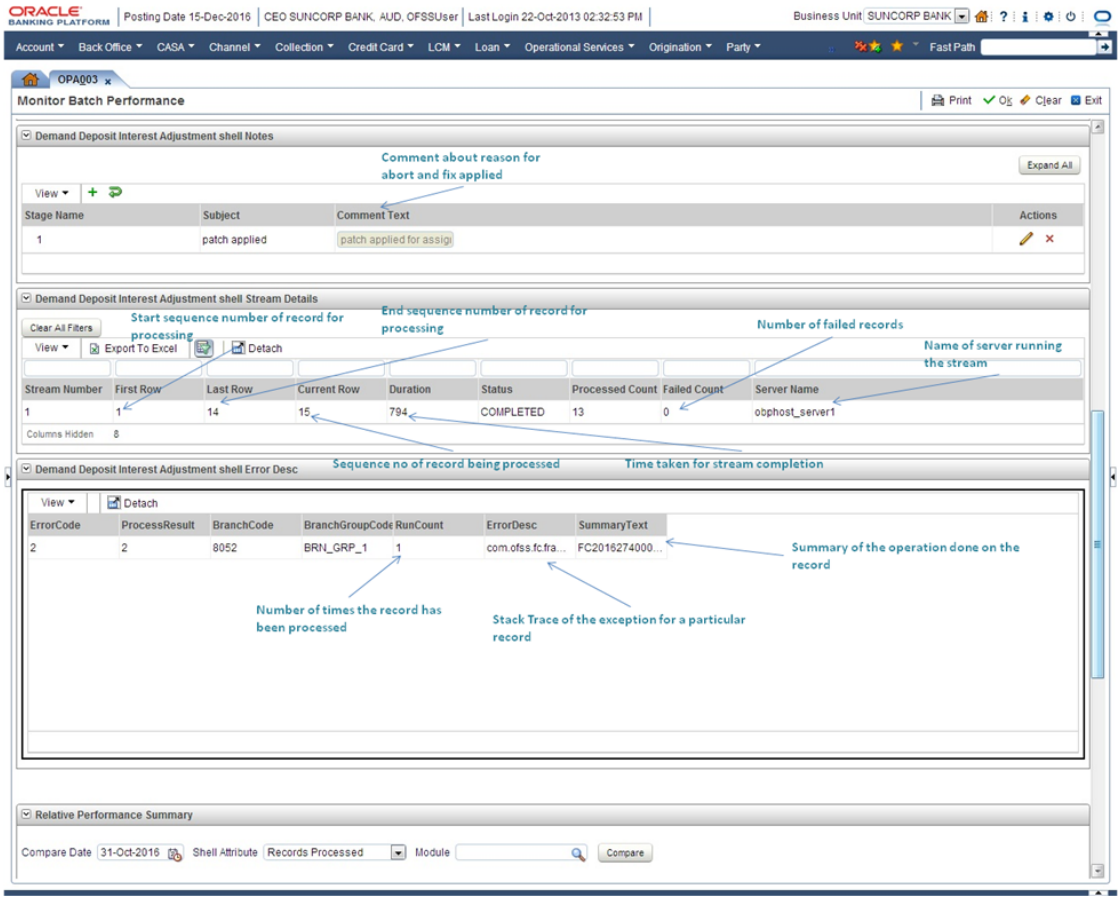
Table 6–4 Stream Details

Attribute Name	Description
Stream Number	Indicates the number of a stream in which the record is being processed
FirstRow	Indicates the start sequence number of a record, processing in a particular stream.
LastRow	Indicates the end sequence number of a record, processing in a particular stream.
Duration	It is the time required for stream to complete.

Table 6–4 (Cont.) Stream Details

Attribute Name	Description
Status	Valid Values: COMPLETED, RUNNING. It indicates the status of selected stream
ProcessedCount	Number of records processed in a stream
ServerName	Name of a server running the stream

Figure 6–19 Stream Based Shells

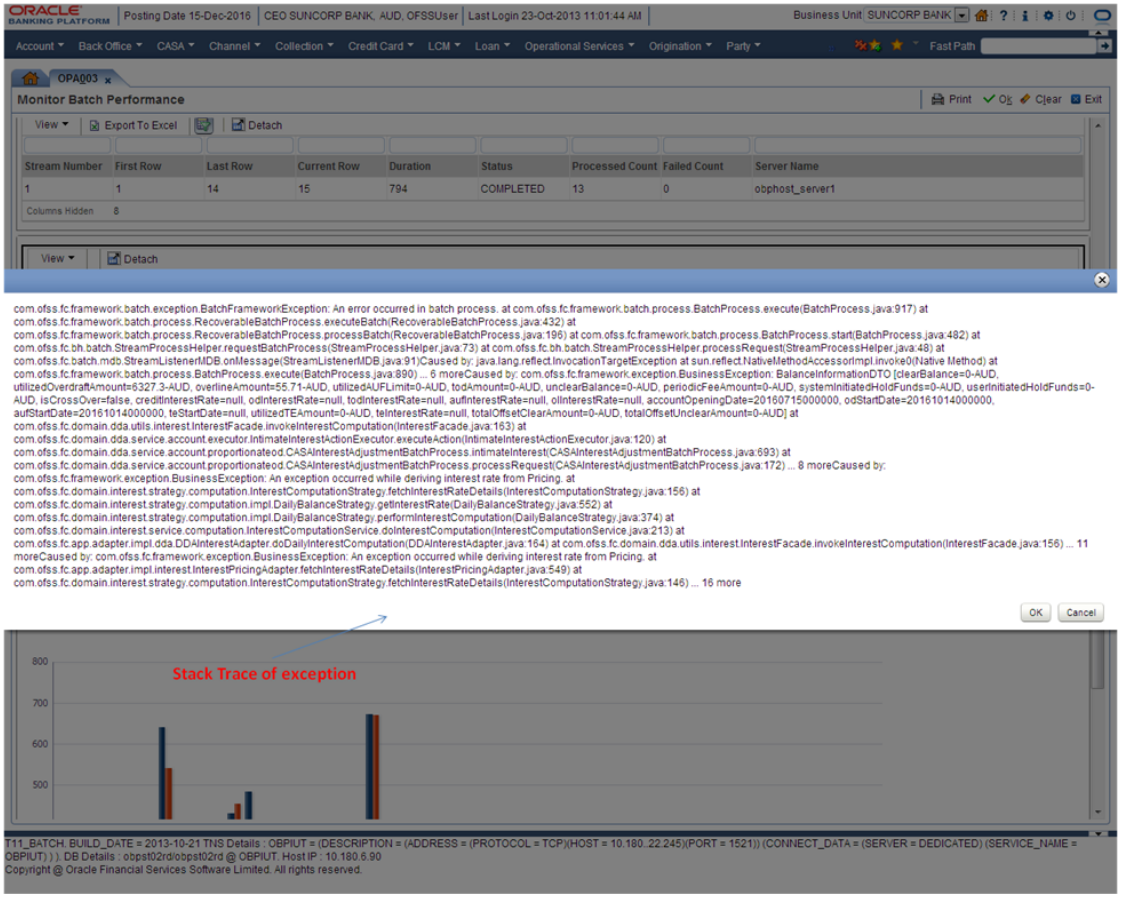


Note: Shells are categorized into two types that is, Stream based shells and Report based shells. Figure 6–19 displays view for stream based shells.

Exception Log

On row click of the driver level details, it pops up a window showing the stack trace of failed records if present. One can analyse and know the reason behind the failure of that particular record.

Figure 6–20 Exception Log



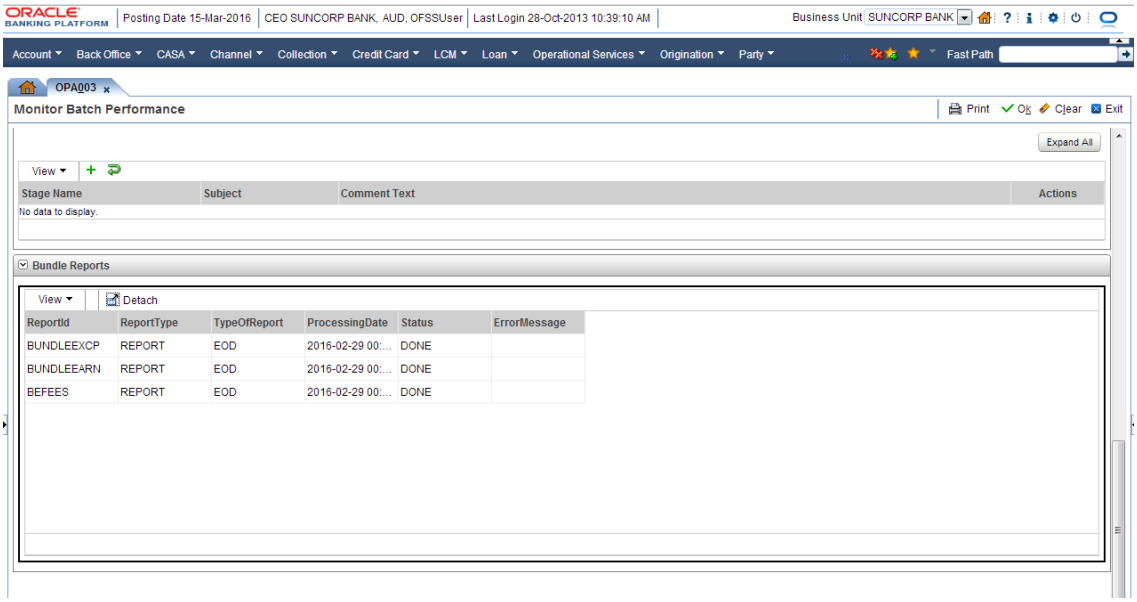
Reports Table

For Report based shells different parameters related to report processing are monitored. The monitored parameters are given below:

Table 6–5 Reports Table

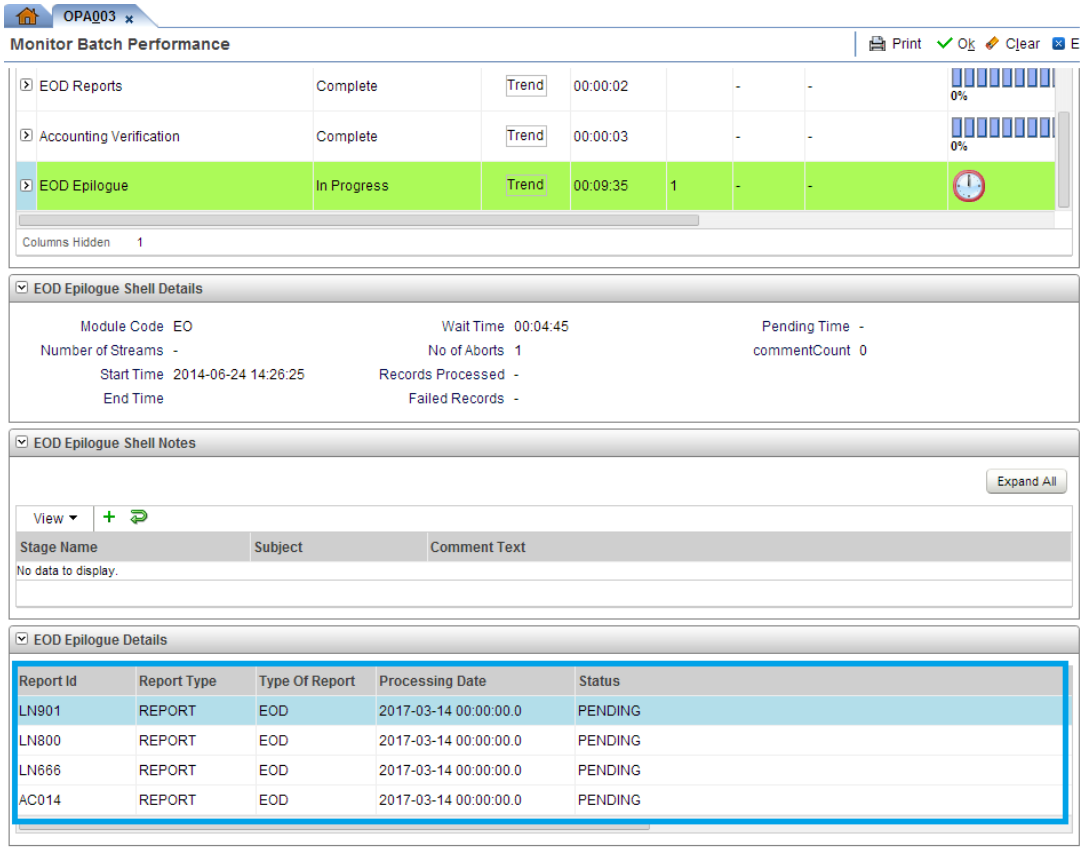
Attribute Name	Description
ReportId	ID to uniquely identify report
TypeOfReport	Indicates type of reports. Reports are classified based on category.
ProcessingDate	Indicates processing date of report.
Status	Valid Values: DONE, PENDING, RUNNING, ABORTED. It indicates the status of the report.
ErrorMessage	Errormessage represents the reason for report failure. No message is diplayed in case of successful run

Figure 6–21 Report Based Shells



The status of report based shell during batch run is shown in Figure 6–22:

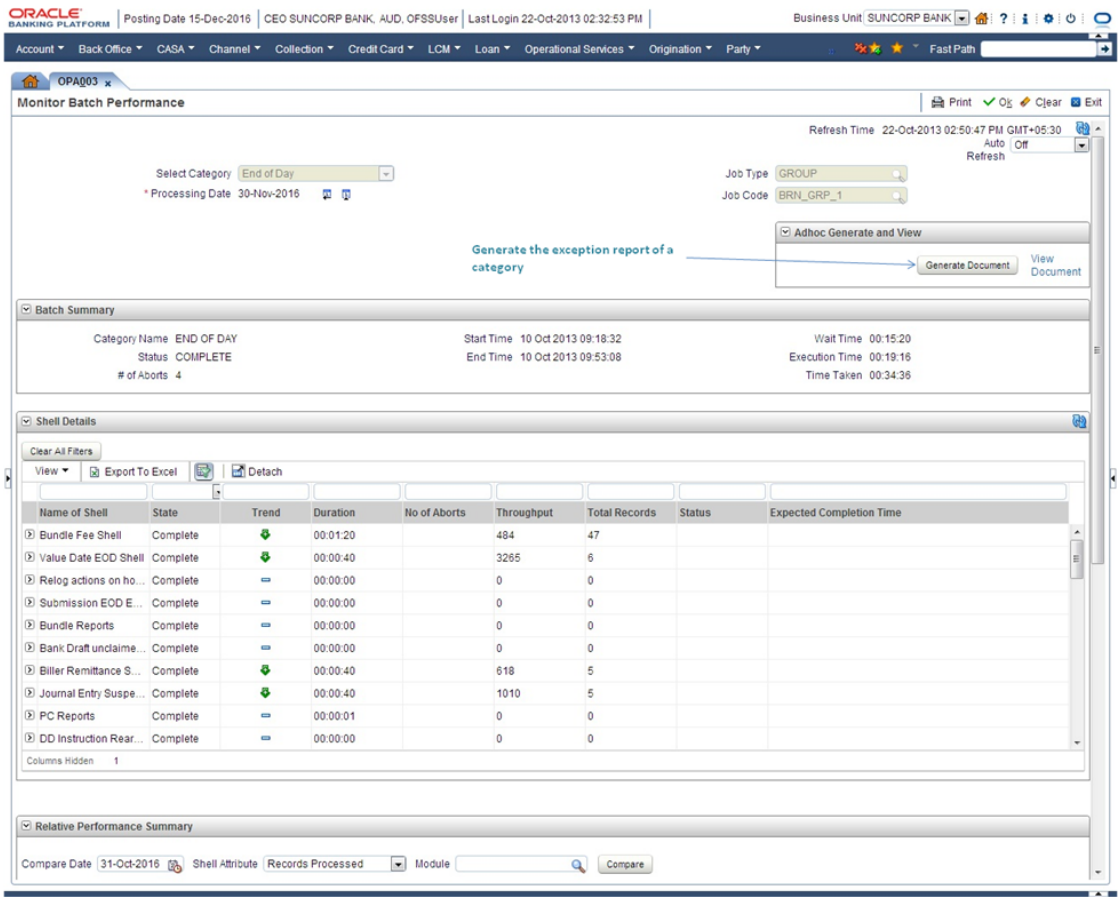
Figure 6–22 Status of Report Based Shell



Exception Report

On click of Generate Document, it generates a report for aborted shells with information like Abort count, exception log and so on.

Figure 6–23 Exception Report



The different parameter monitored at shell level and exception logs for all aborted shells are part of exception report. Figure 6–24 displays sample report for a particular shell.

Figure 6–24 Sample Report

Bundle Fee Shell

Start Time :	11-Sep-2013 6:58 AM	**End Time :**	11-Sep-2013 7:23 AM
Records Processed :	6	**Duration :**	00:24:26
Number of Aborts :	2	**Wait Time :**	00:23:23

ABORT STATISTICS

ABORT_TIME	RESTART_TIME	ABORT DURATION
11-Sep-2013 7:15 AM	11-Sep-2013 7:22 AM	00:07:38
11-Sep-2013 6:59 AM	11-Sep-2013 7:14 AM	00:15:45

Exception Log Table

The figure below provides the details of the exception log.

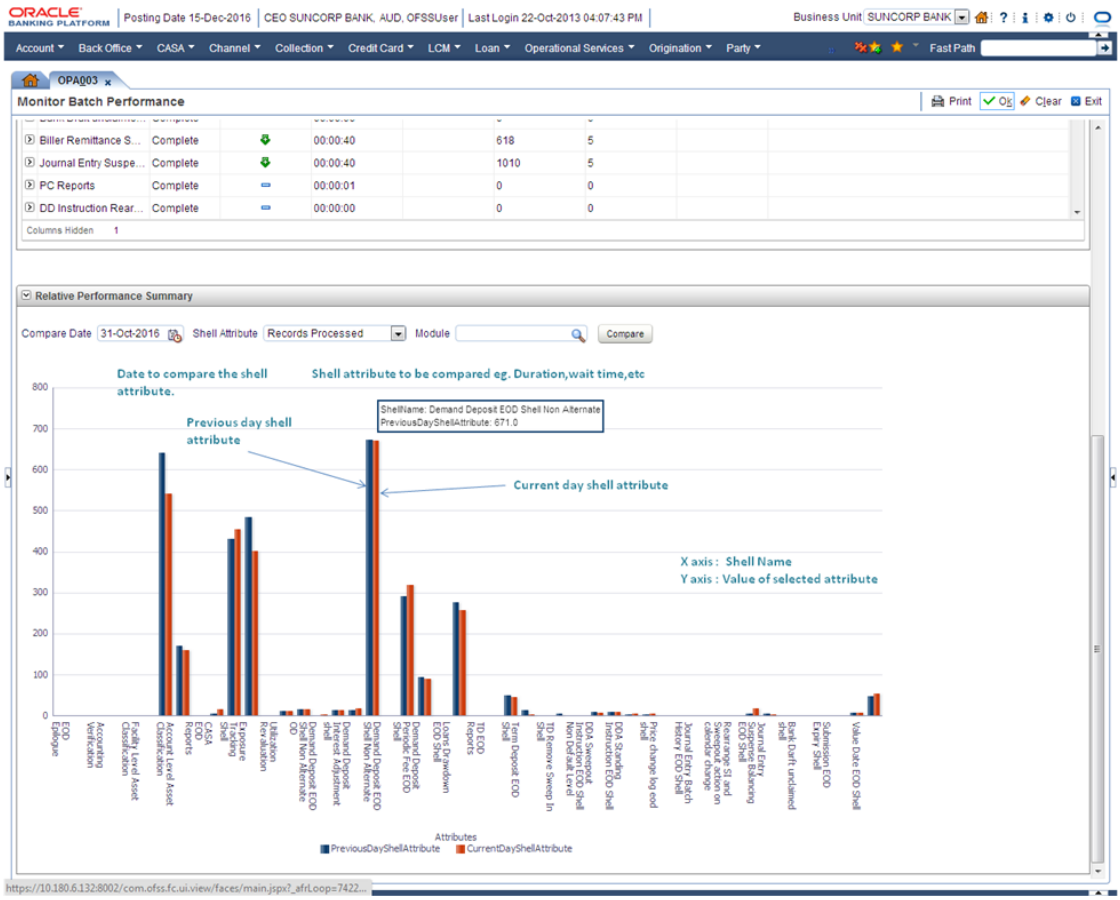
Figure 6–25 Exception Log Table

ERROR CODE	ERROR DESCRIPTION	SUMMARY
kException: An error occurred in batch process.862-8203-0444-6360		
Time Of Last Update : 2013-09-11 11:28:51.438 Error reference Number :862-8182-2552-7227Error msg :An error occurred in batch process.Error cause :com.ofss.fc.framework.batch.exception.BatchFrameworkException: An error occurred in batch process.862-8182-2552-7227	com.ofss.fc.framework.batch.exception.BatchFrameworkException: An error occurred in batch process. at com.ofss.fc.framework.batch.process.BatchProcess.execute(BatchProcess.java:910) at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.executeBatch(RecoverableBatchProcess.java:432) at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.processBatch(RecoverableBatchProcess.java:	
Time Of Last Update : 2013-09-11 11:28:53.868 Error reference Number :862-8206-5767-2044Error msg :An error occurred in batch process.Error cause :com.ofss.fc.framework.batch.exception.BatchFrameworkException: An error occurred in batch process.862-8206-5767-2044	com.ofss.fc.framework.batch.exception.BatchFrameworkException: An error occurred in batch process. at com.ofss.fc.framework.batch.process.BatchProcess.execute(BatchProcess.java:910) at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.executeBatch(RecoverableBatchProcess.java:432) at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.processBatch(RecoverableBatchProcess.java:	

6.2.3 Histogram of Shell Attribute Comparison

This section provides graphical view for comparing shell attributes for any two selected dates. The processing date, shell attribute and module name are taken as input to this table. The output is displayed as bar graph in which X axis represents the name of the shell and Y axis denotes value of shell attribute. For each shell two records are displayed, these records corresponds to the dates for which the data is being compared.

Figure 6–26 Graphs

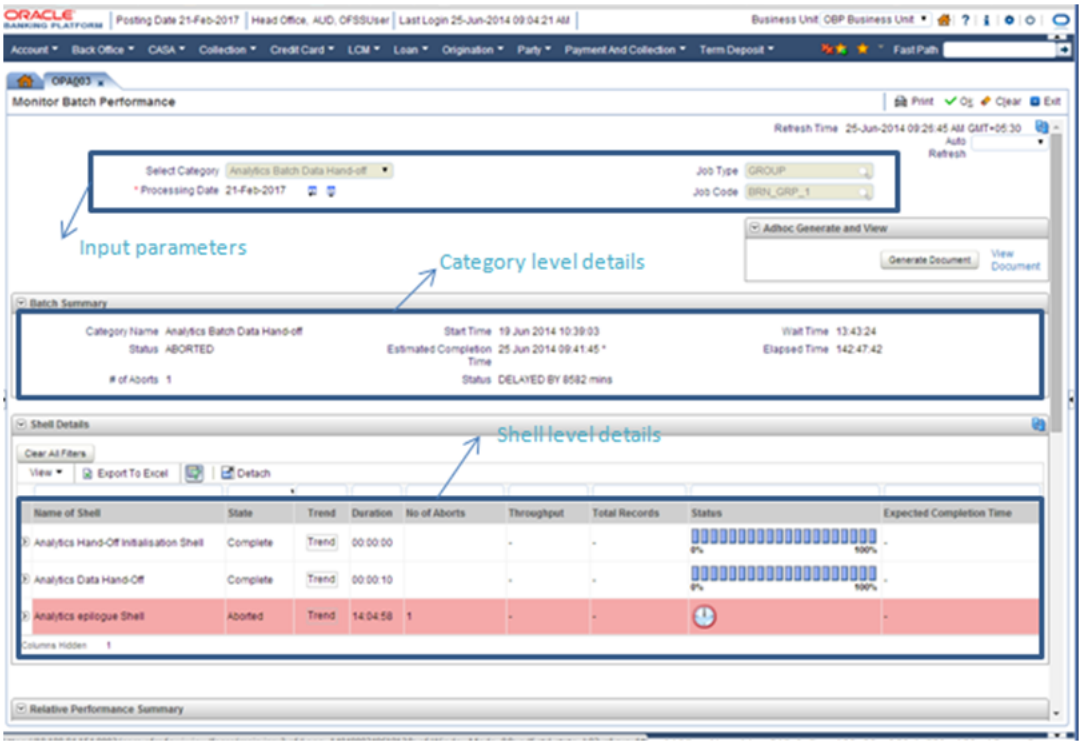


6.3 ODI Batch Handoff Monitoring

This section provides, top package level execution details for monitoring of ODI handoff. The input for these details are category ID, branch group code, branch group type, and processing date.

Figure 6–27 describes the input parameters for the batch handoff monitoring and the shell details like name of shell, start time, end time, number of aborts.

Figure 6–27 Input Parameters for Batch Handoff



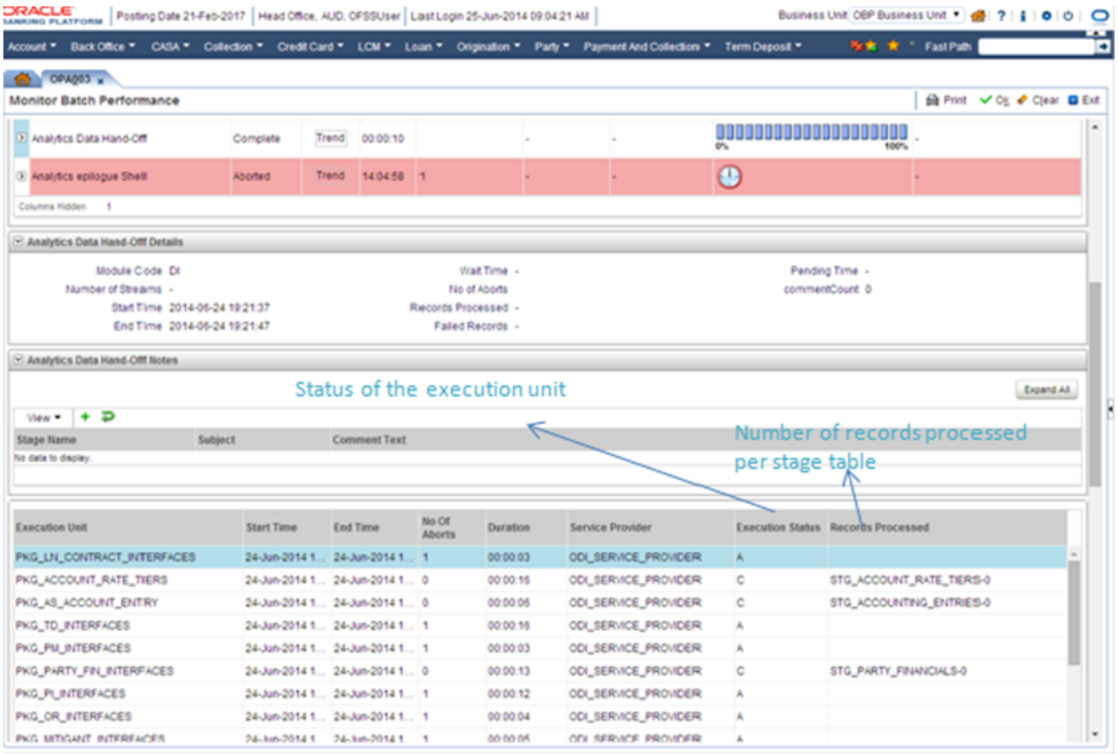
Execution Unit

On click of the analytics data Handoff shell, the below table is shown with the execution unit (top level package) level details:

This table contains the following attributes:

- Execution unit name
- Start time of execution of the execution unit
- End time of execution of the execution unit
- Number of aborts of the execution unit
- Duration of execution of the execution unit
- Service provider for ETL process (ODI)
- Execution status of the execution unit, that is, complete, running, and aborted
- Records processed at the stage level in ETL process

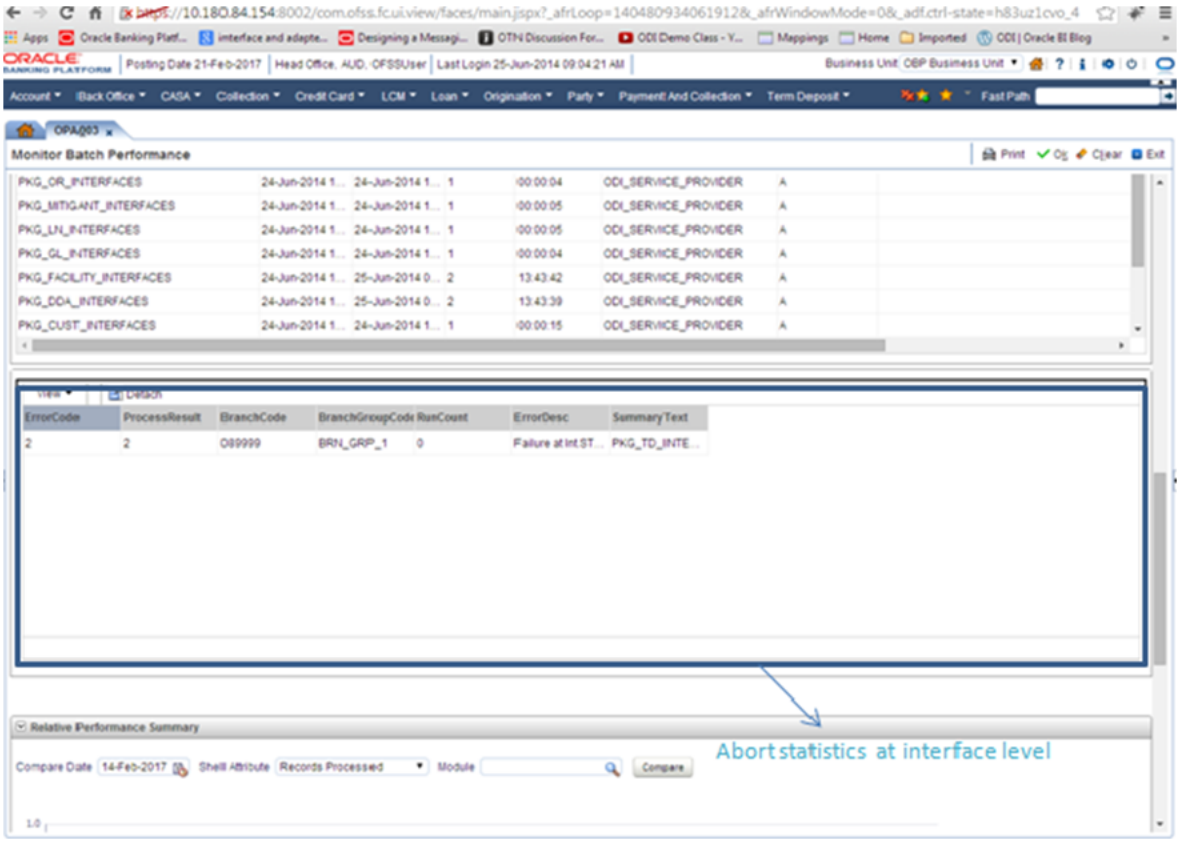
Figure 6–28 Execution Unit



Abort Statistics

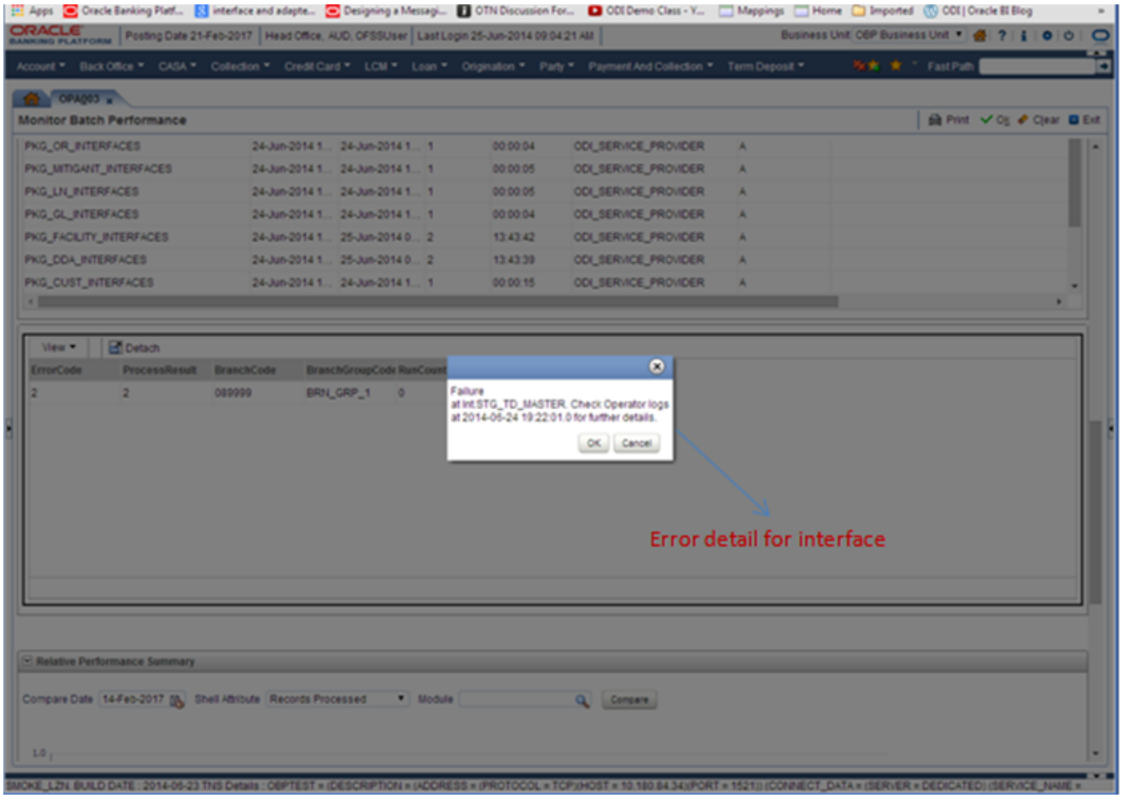
On click of aborted execution unit, the below table is shown with the abort details like run count, the actual error description, and summary of the exception containing the interface name for which the exception occurred.

Figure 6–29 Abort Statistics



On click of the error description table, the error description message appears as shown in Figure 6–30:

Figure 6–30 Failure Error Description



7

Application Monitoring Using OBP EM Plugin

This chapter provides an overview on the various monitoring operations performed as an administrator, using Oracle Banking Platform - Enterprise Manger (EM) Plugin.

7.1 Monitoring Application Using OBP EM Plugin

Once DMS statistics are captured for a particular Channel and transactions involving in it, it requires a UI representation to understand the stats in a readable form so that one can analyse the behaviour. The monitoring activities are mainly carried out by IT Technical staff.

7.1.1 Oracle Enterprise Manager (EM)

Oracle Enterprise Manager is the application where all the OBP monitoring data exists. It includes server and machines status and performance and also OBP monitoring statistics.

All the OBP servers are monitored by EM including Host, UI, SOA, and so on.

We have a view corresponding to every environment containing all the components of OBP which include outbound components.

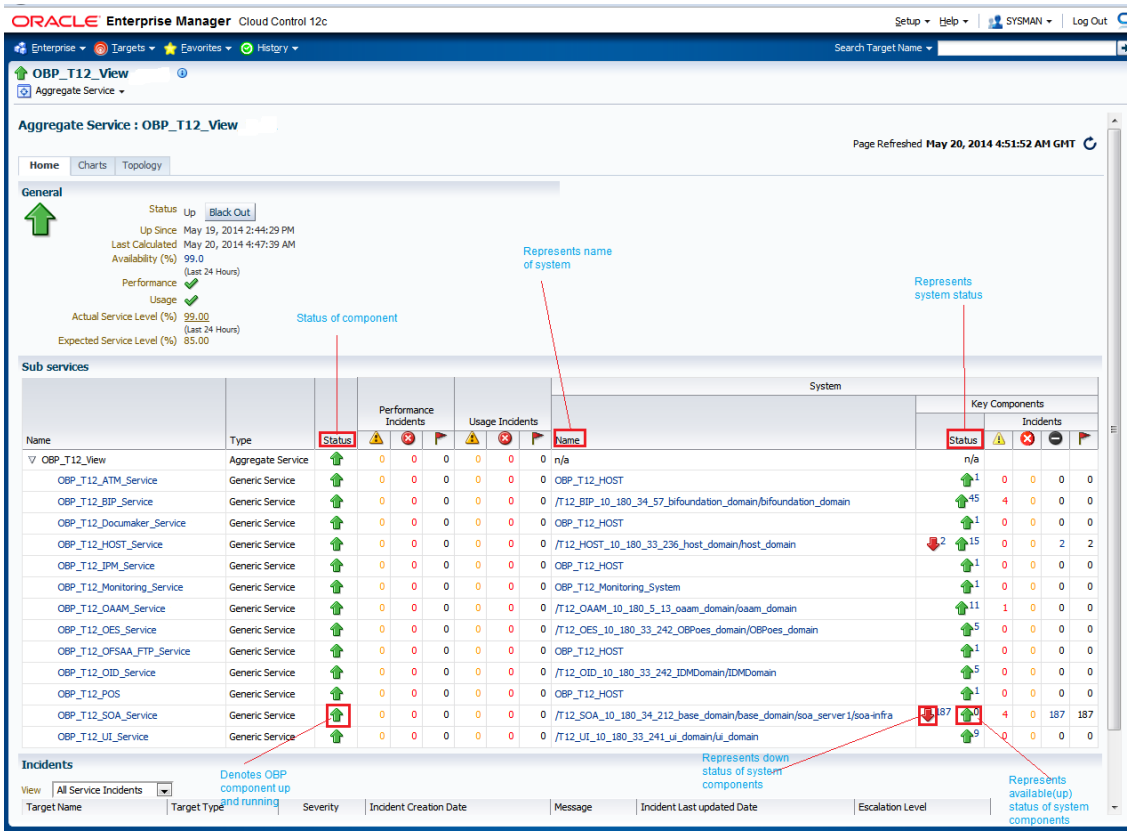
Some notations in EM are provided below:

Table 7–1 Notations in EM

	Indicates component is down
	Indicates component is up and running
	Indicates Alerts
	Indicates Warnings
	Indicates metric collection error
	Indicates healthy status

Figure 7–1 shows the view of T12 environment:

Figure 7–1 View of T12 environment



The views in Figure 7–1 include OBP UI, Host and SOA servers.

Security Stacks components such as OAAM, OID, OES, outbound components such as BIP, IPM, Documaker, ATM and POS channels are also part of the environment view.

Each component can be further explored for details by clicking on the links provided for them.

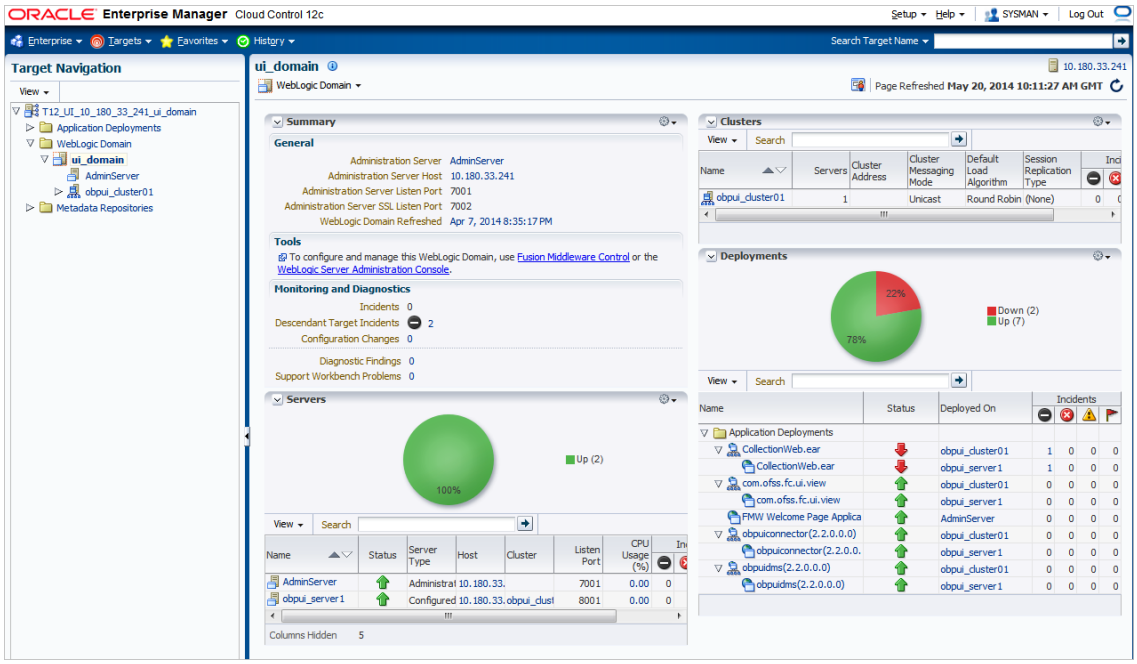
7.1.2 OBP UI

OBP UI is hosted on WebLogic domain, so the EM target of UI machine is WebLogic domain. EM gives the following information for UI:

- Server Performance Statistics
- Up/Down Status
- List of deployed applications
- Incidents or Alerts; if any

Figure 7–2 displays the WebLogic domain for UI.

Figure 7–2 WebLogic Domain for UI



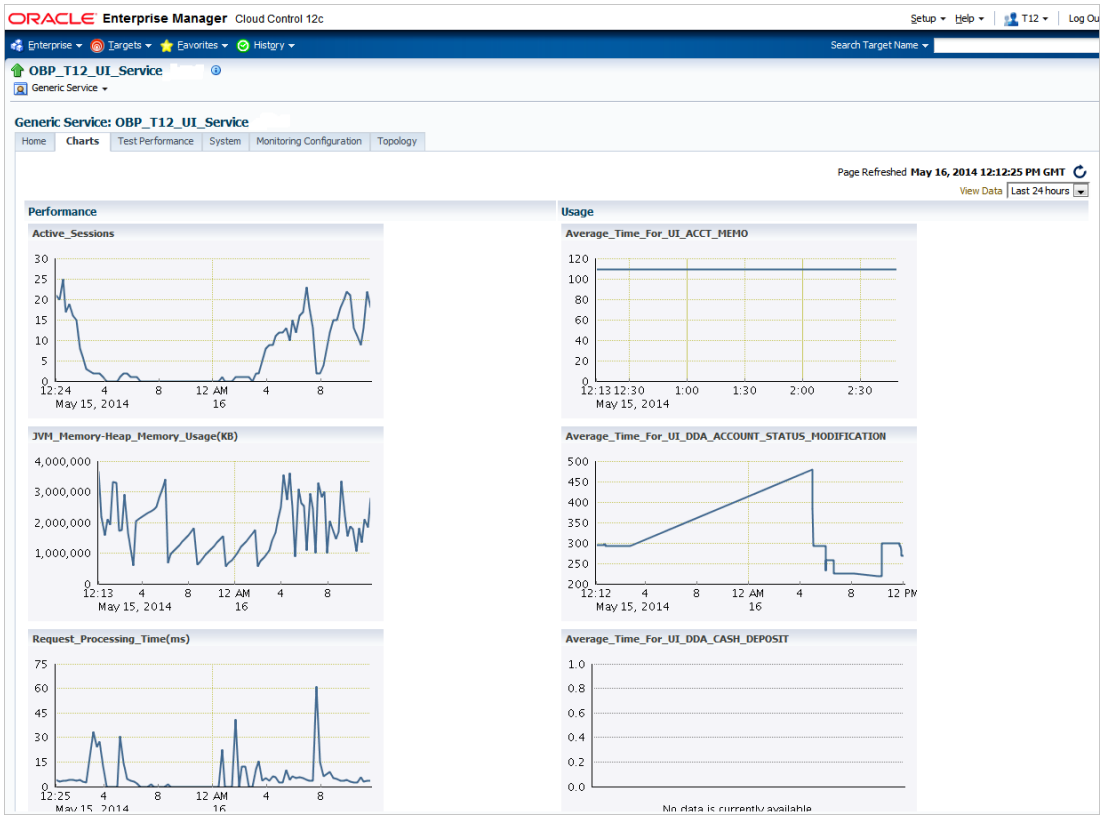
UI contains OBP DMS metric along with other default metrics which can be shown in EM in the form of charts.

The performance metric includes metrics like CPU Utilization, Memory Utilization, Active Sessions and are default metrics provided by EM.

The usage metrics are user defined and in this case it is DMS metrics for UI like Average time for UI DDA Cash deposit and so on.

Figure 7–3 displays the Usage Metrics:

Figure 7–3 Usage Metrics



7.1.3 OBP Host

Similar to UI, OBP host is also deployed on WebLogic domain and has similiar metrics like UI.

Figure 7–4 displays the OBP host target in EM.

Figure 7–4 OBP Host Target in EM

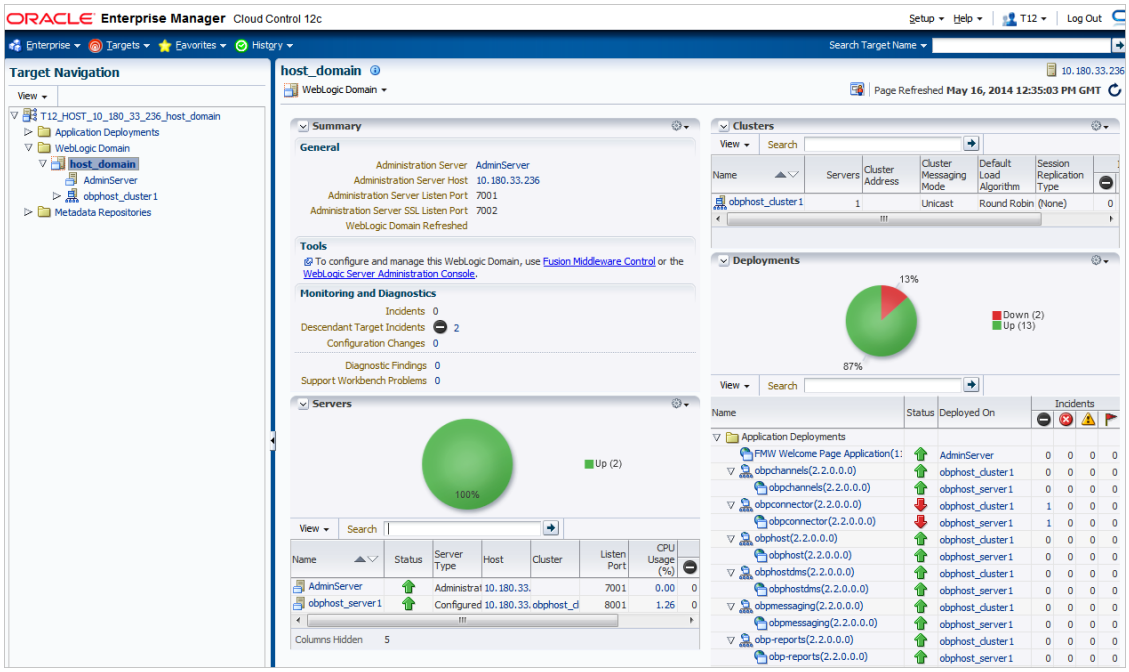
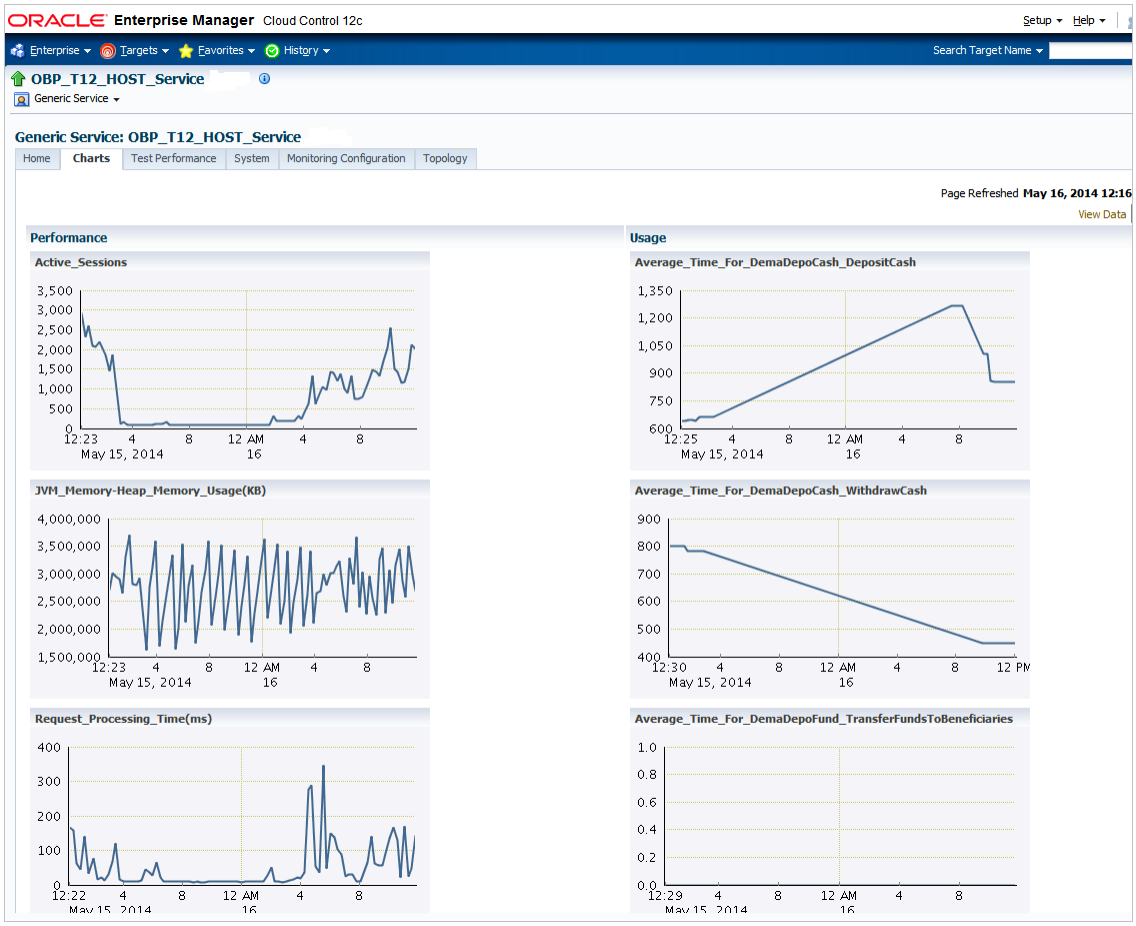


Figure 7–5 displays the metric charts:

Figure 7–5 Metrics Chart

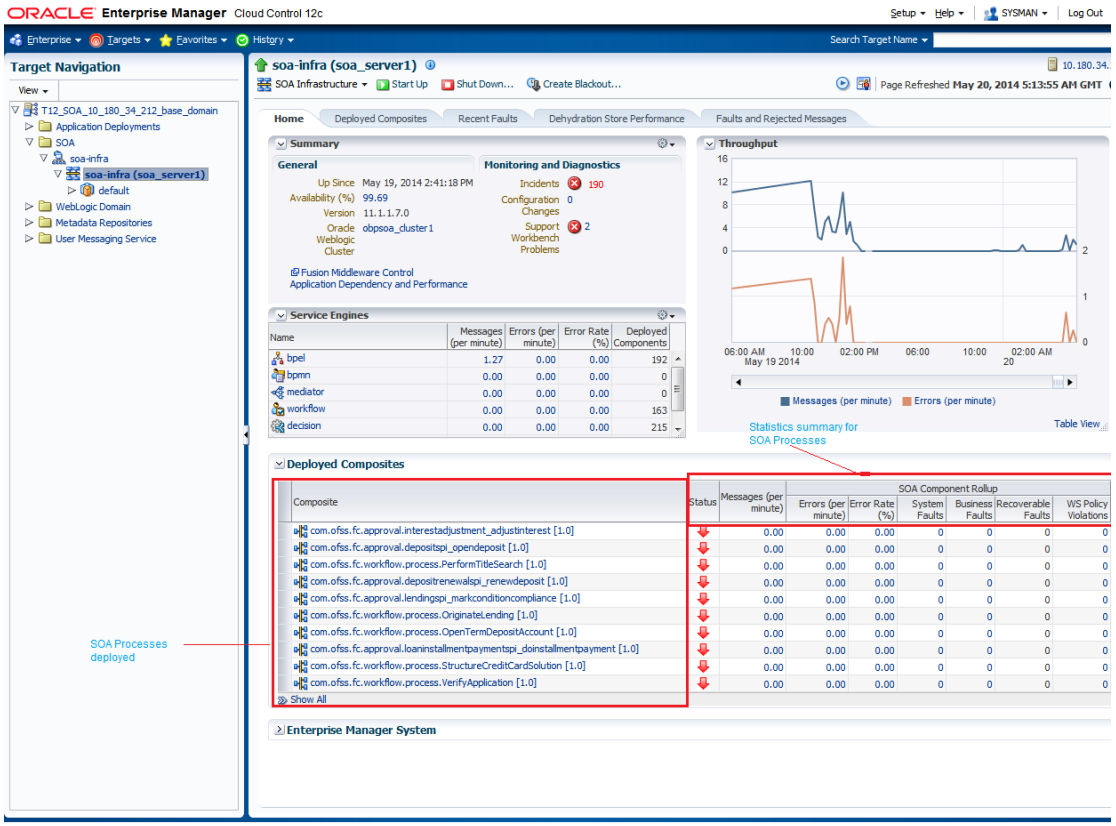


7.1.4 OBP SOA

In OBP, SOA server is deployed on WebLogic domain where the SOA processes are deployed.

The process list can be seen in the list of deployed applications. The other metrics remain same as for WebLogic domain in EM. Figure 7–6 displays the process list.

Figure 7–6 Viewing Process List



7.1.5 OBP Security Stack (OID, OES, OAAM)

OID, OES and OAAM are also deployed as WebLogic domain.

Figure 7–7 OID WebLogic Domain

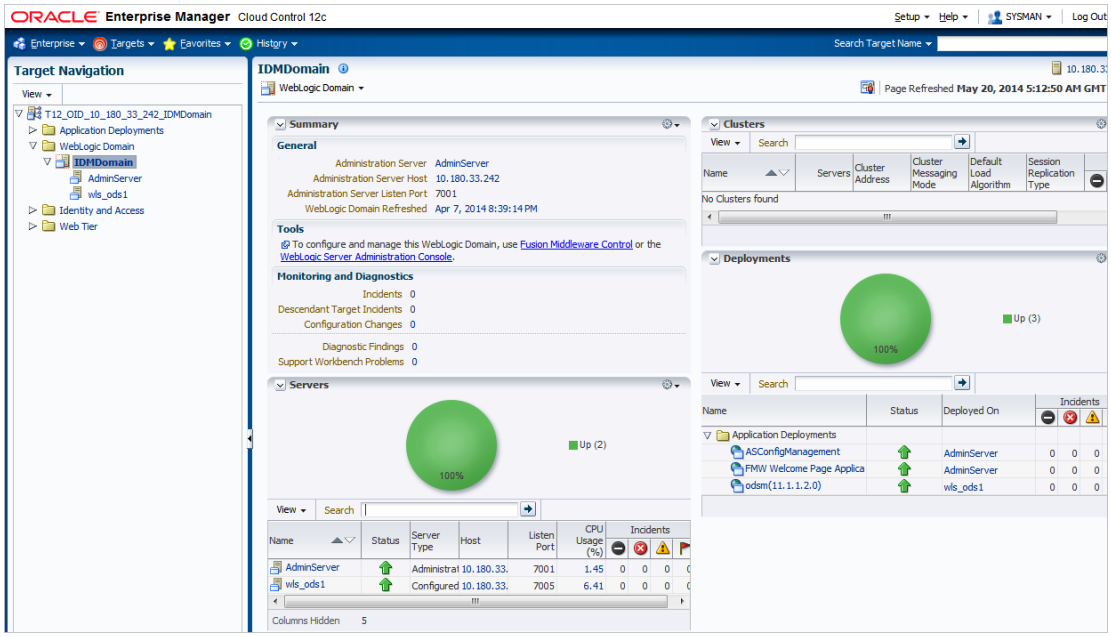


Figure 7–8 OES WebLogic Domain

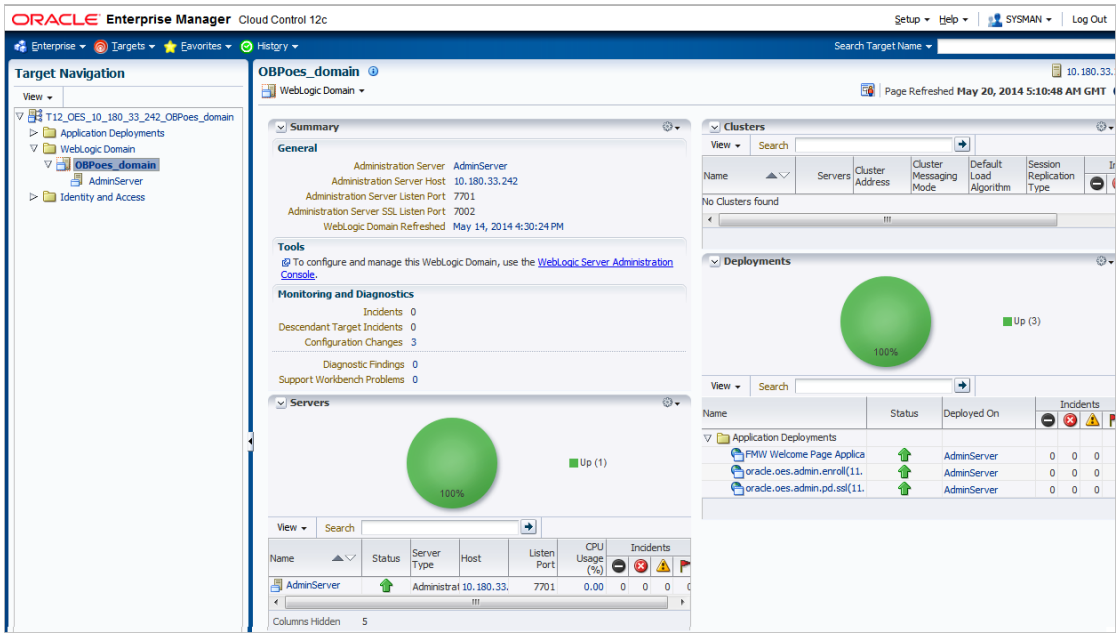
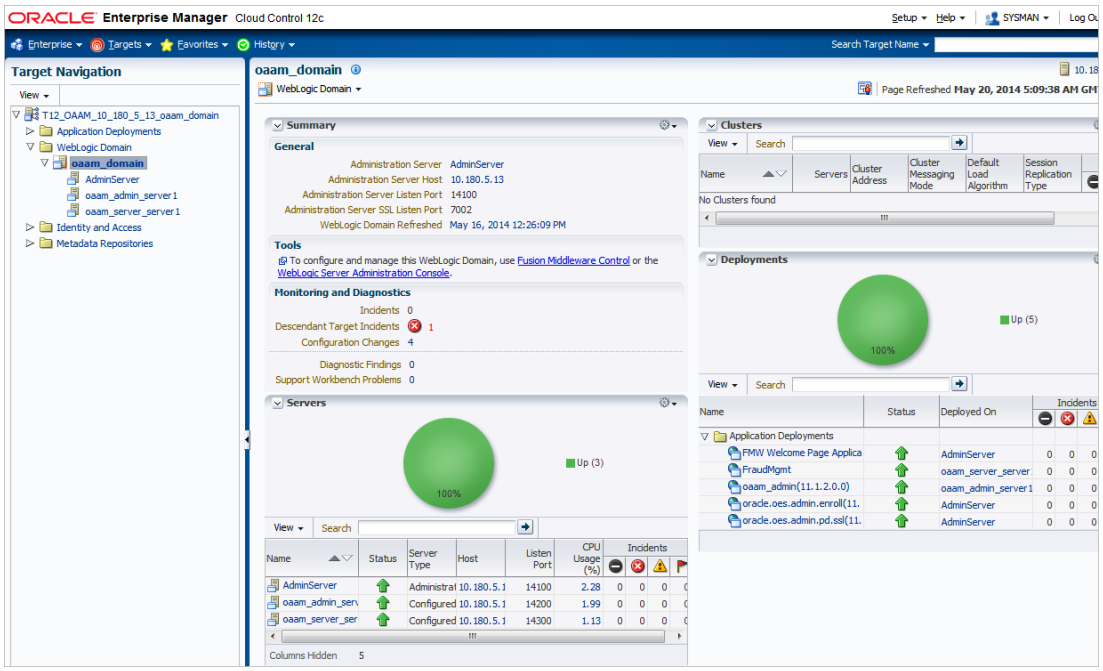


Figure 7–9 OAAM WebLogic Domain



7.1.6 Document Generation Outbound Components (Documaker, BIP, IPM)

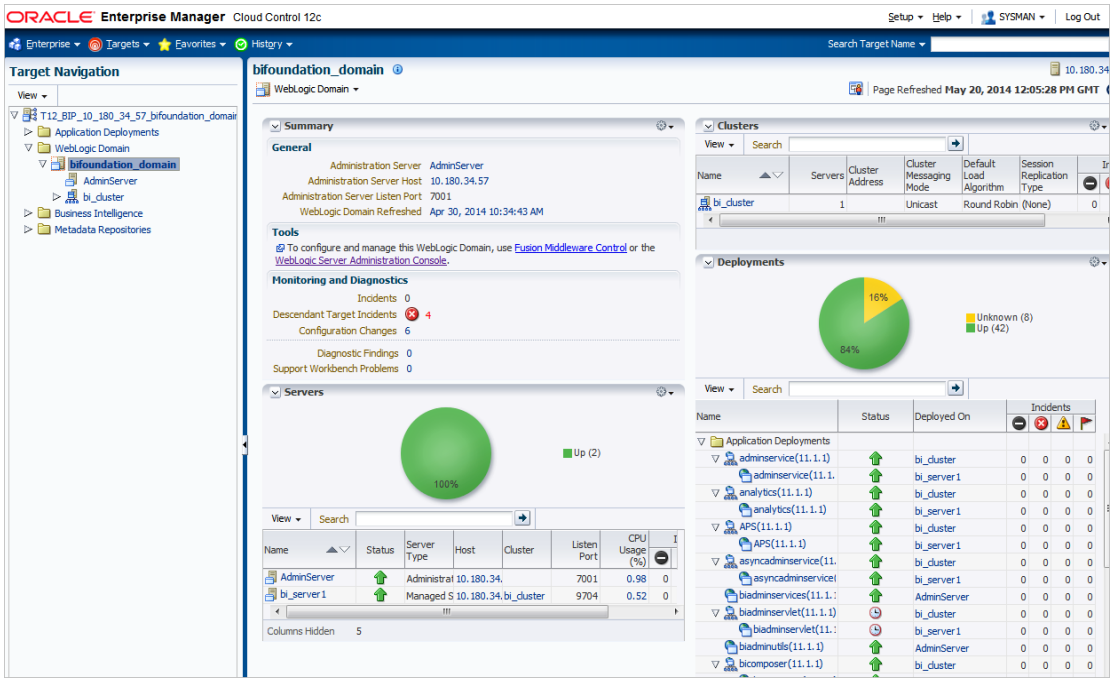
These are not part of OBP application, but we monitor these so as to detect the cause of failure in case the document generation fails at any point of time.

A webservice is invoked for generating the documents from OBP with BIP as well as documaker. From EM, we check whether that webservice is up and running or not, which gives the status of these components. Figure 7–10 displays the status from EM.

Figure 7–10 Document Generation Status

BIP is also deployed on WebLogic domain.

Figure 7–11 BIP Deployment



7.1.7 ATM and POS (Point Of Sales) Channels

ATM and POS work on socket listener mechanism.

So, for them to be up and running the port on which they listen should be up. In EM, to monitor these channels we check if the port is listening.

Figure 7–12 displays the status from EM.

Figure 7–12 EM Monitoring

7.1.8 Outbound OFSAA call

OBP calls OFSAA for calculation of economic cost. This is done through a webservice.

To monitor this, we check if the webservice is up and running.

Figure 7–13 Web Monitoring

7.1.9 OBP Monitoring Views

OBP monitoring views show the OBP batch and application performance statistics along with server performance history. It consists of Batch Monitoring and Application Monitoring tabs which show detailed view of batch performance and application performance statistics along with the server performance statistics on which they are running.

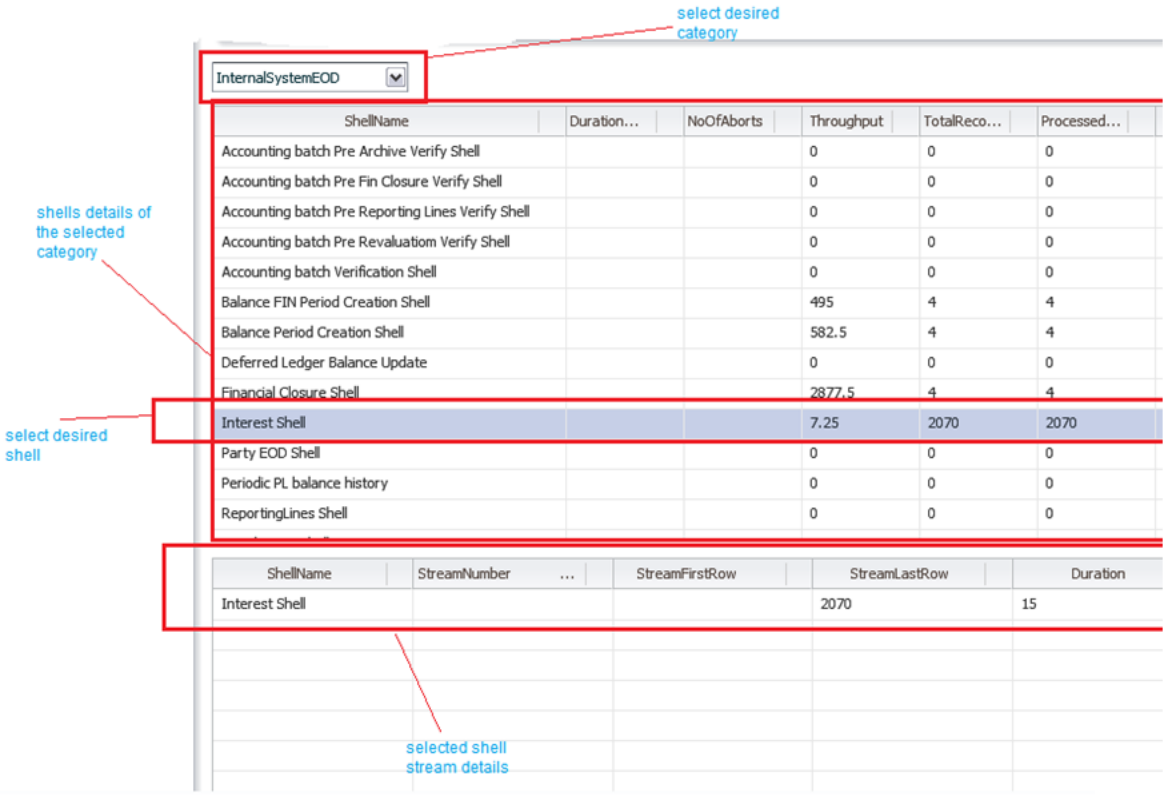
Batch Monitoring

This shows detailed view for host and database server performance charts along with batch performance statistics of OBP.

The batch performance statistics are the details of the categories run in OBP. The date for which category details are shown is the last run date. The categories include EOD, CutOff, Internal System EOD and BOD.

To get the details of a particular category, select it from the combo box. This will display the list of shells in the category in the table below. From the table, select the desired shell, the shell details provides the stream details of the selected shell.

Figure 7–14 Database Server Info



The streams can run in different servers. To get the details of the performance of the server in which the stream is executed, select the stream. The charts below gives the performance summary of the server in which the stream is executed and the database performance.

Figure 7–15 displays the status from EM:

Figure 7–15 Batch Monitoring Status

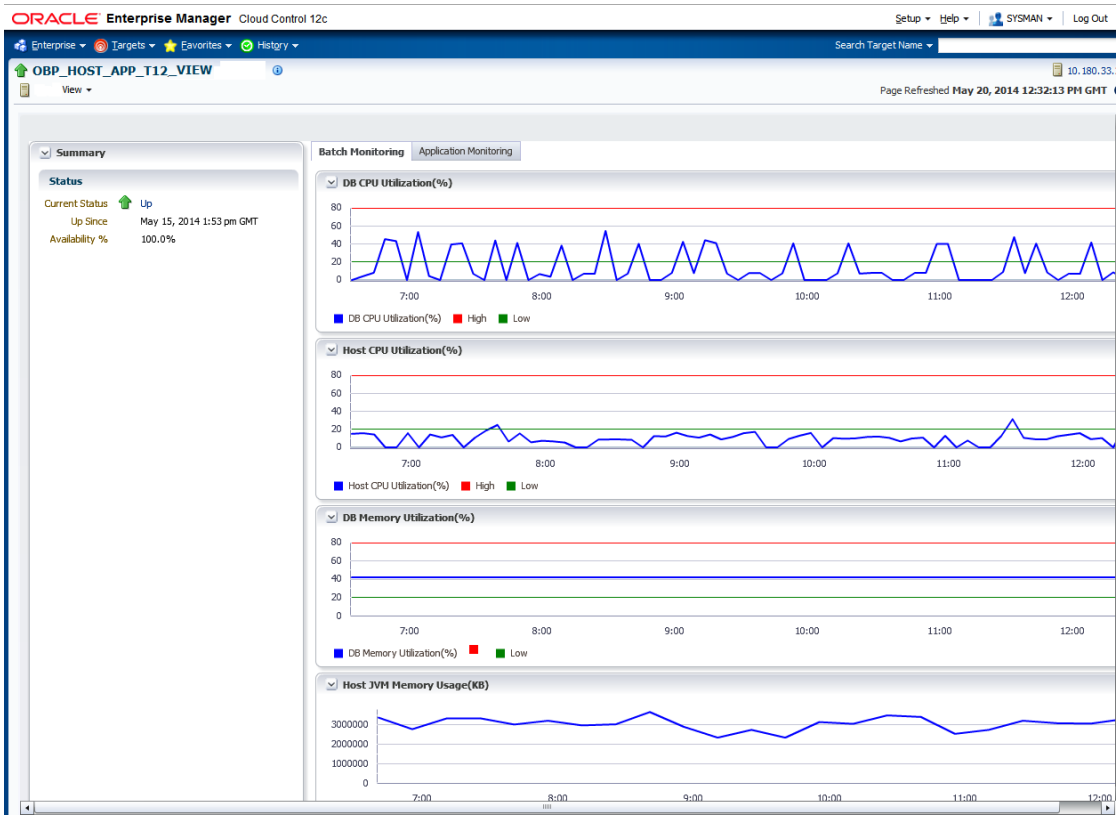


Figure 7–16 Batch Configuration

Figure 7–17 WebLogic Service Info

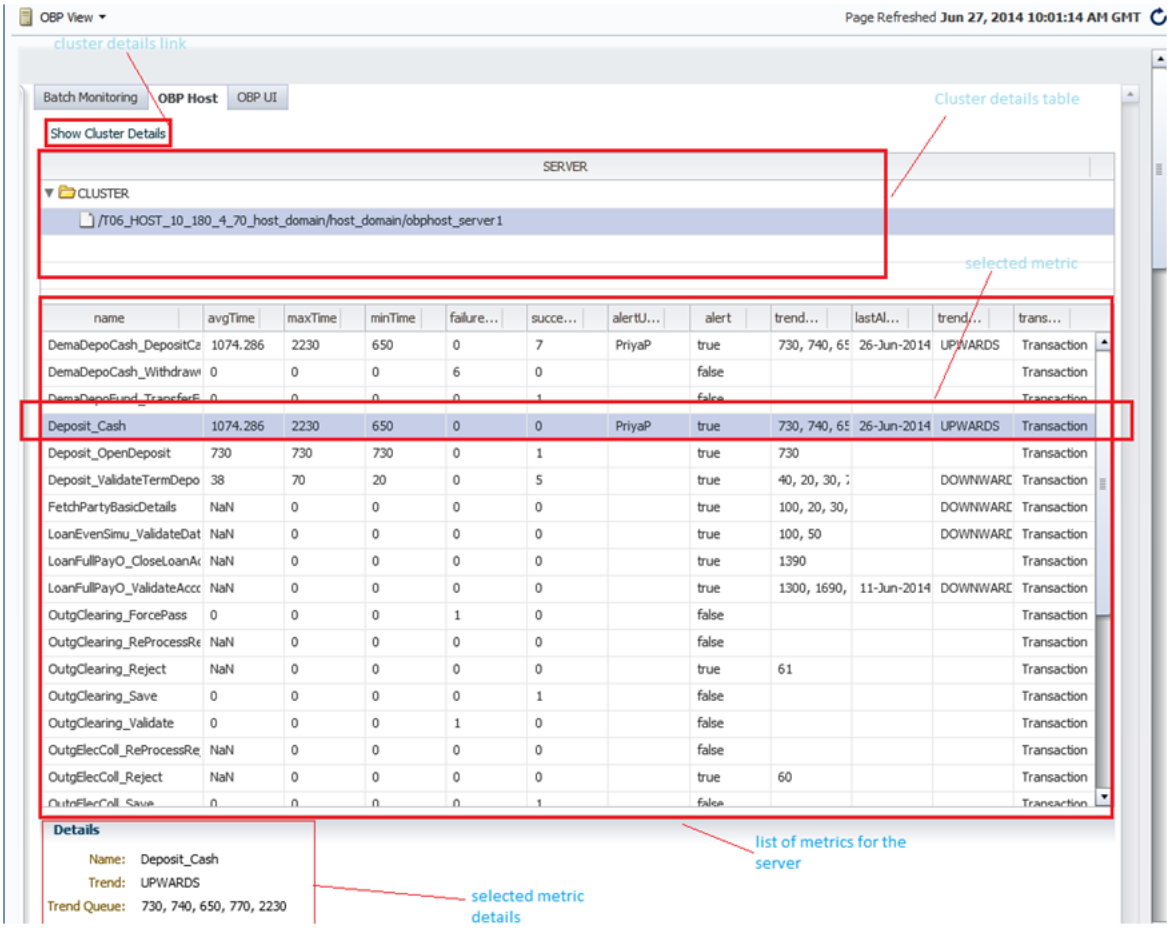
Application Monitoring

Application Monitoring shows detailed view of UI and host clusters and servers.

There are two separate tabs for UI and host. Same details are shown for both the servers. To get details of the cluster, click the link 'Show cluster details', which gives the cluster information in the data grid.

Expand it to get the list of all the servers. Each server can be further selected to get the details at the server level. On clicking the server, the application metrics are displayed in the table below. As shown in the figure.

Figure 7–18 Application Metrics



Also, details of each metric along with the metric historical chart format can be seen on selecting any metric from the table.

Performance history graphs also gets displayed on selection of server from the cluster details table as shown below:

Figure 7–19 Performance History Graphs

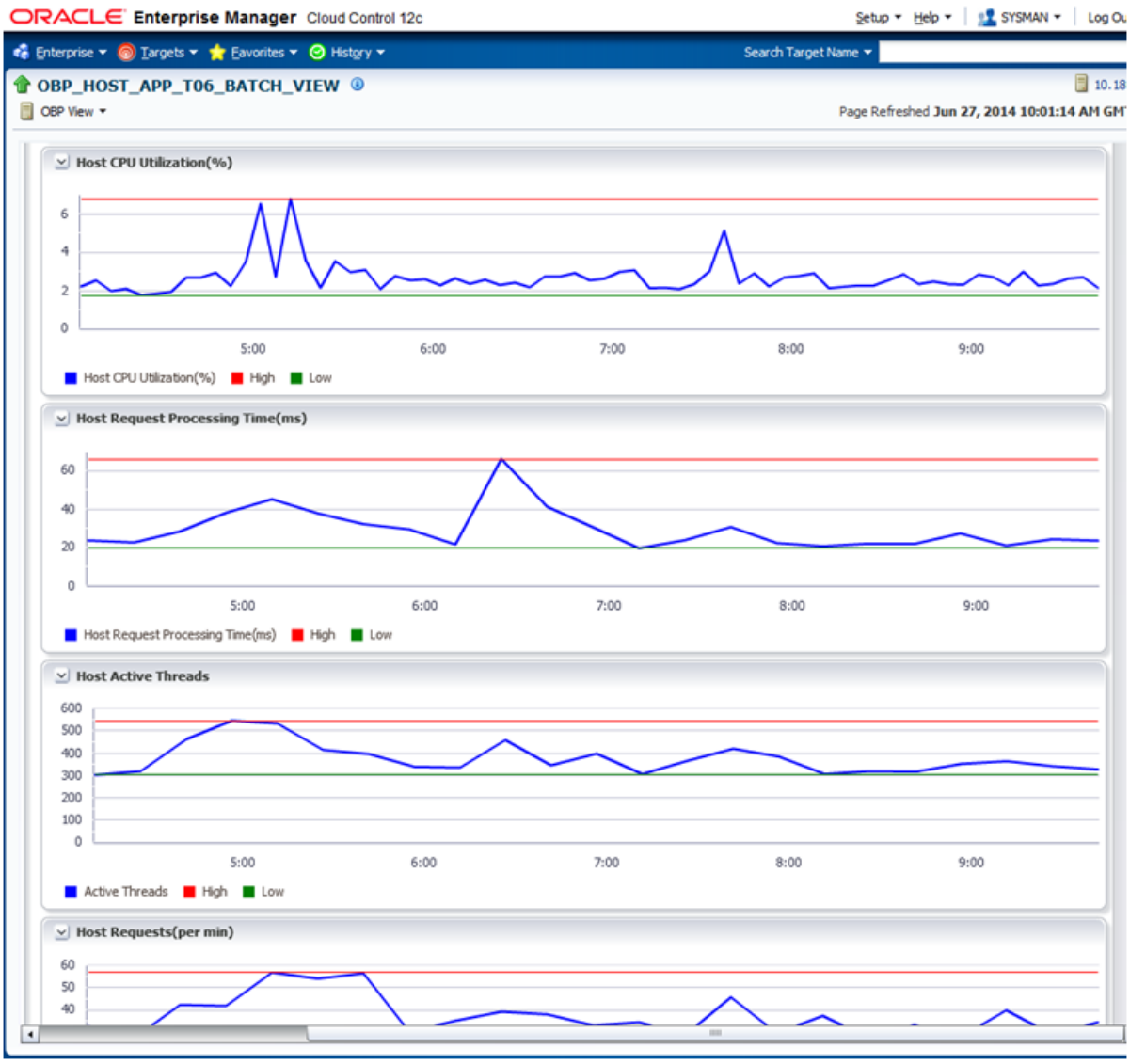
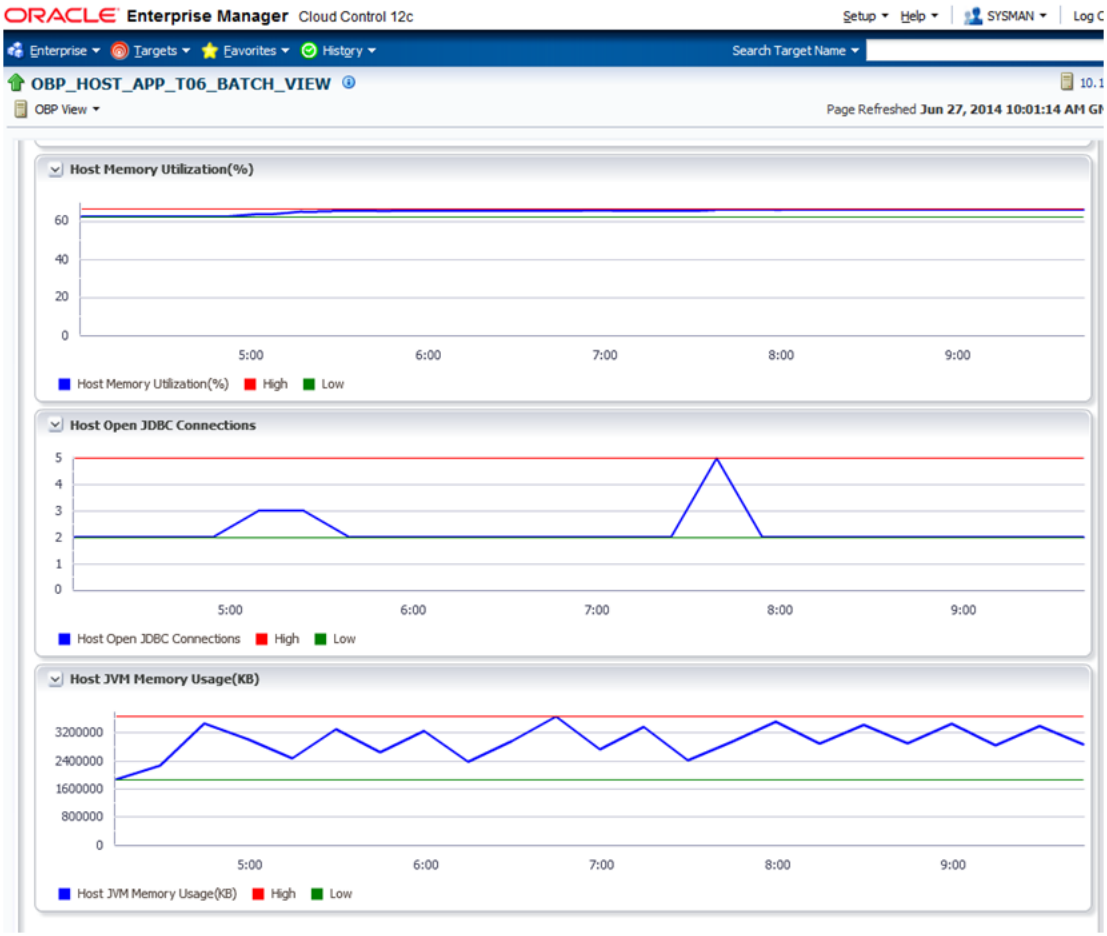


Figure 7–20 Performance History Graphs - Continuation



8

Configuration Export-Import Operations

This chapter gives an insight to the Configuration Export-Import operations.

8.1 Objective

Config operations include exporting business configurations, from one environment, to DB or file and importing these configurations in another environment, thus replicating the entire data with the golden copy.

Compare Business Configurations (OPA005) page provides the UI to compare the entities present in two environments on the basis of the data attributes.

Suppose R1 is an environment where the teller has maintained an entity, say currency AUD and in R2 environment the teller wants the exact copy of R1. The 'Import Export' operations allows the user to export a single entity or all entities of a 'taskCode' and can replicate the working environment with the exported version of data very effectively.

The overall Config operations are divided into five parts, each part representing an operation with its specific functionality. The user has the option to invoke any of the operation to get the required work done.

8.2 Export

This operation aims at exporting a business configuration of a 'taskCode' to the configured location. It stores the serializable response of the entity. When export operation is invoked, data gets exported to the database or file as per configuration.

This operation can be carried out as a webservice call for the Export operation of the specific taskcode whose page level configuration has not been done.

8.3 Import

This operation aims at replicating the entity of target environment with exported data from a source environment. It retrieves the serializable response of the entity from database or file as per configuration and de-serializes the response to replicate the entity in target environment. When import operation is performed, it fetches the response from the source environment database and inserts/updates in the target environment.

8.4 Export All

This operation aims at exporting all the entities of a given taskcode. So that the same can be replicated in other environment. It is carried out through a web service call, by invoking the fetchAllAndExport method of ExportImportApplicationService. The request parameters are sessionContext, taskCode.

Export Request

Export request xml is provided below:

```
- <soapenv:Envelope xmlns:soapenv="http:// schemas.xmlsoap.org/soap/envelope/"
xmlns:exp="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:con="http:// context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com">
  <soapenv:Header />
- <soapenv:Body>
- <exp:fetchAllAndExportExportImport>
- <exp:sessionContext>
  <con:bankCode>48</con:bankCode>
  <con:businessUnit>SUNCORP</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>SUNCORP</con:targetUnit>
  <con:transactionBranch>8542</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </exp:sessionContext>
  <exp:taskCode>PM031</exp:taskCode>
  </exp:fetchAllAndExportExportImport>
  </soapenv:Body>
  </soapenv:Envelope>
```

Export Response

Once this service is invoked with the above request, it fetches the configVersionNo of the exported data in response which is the version number with which Export All was performed.

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Body>
- <ns11:fetchAllAndExportExportImportResponse
xmlns:ns11="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationServic
e" xmlns:ns10="http://fact.enumeration.fc.ofss.com"
xmlns:ns9="http://enumeration.fc.ofss.com"
xmlns:exceptioninfra="http://exception.infra.fc.ofss.com"
xmlns:datatype="http://datatype.fc.ofss.com"
xmlns:contextapp="http://context.app.fc.ofss.com"
xmlns:dtocoreseedopsapp="http://dto.core.seed.ops.app.fc.ofss.com"
xmlns:dtocommondomainframework="http://dto.common.domain.framework.fc.ofss.com"
xmlns:errorvalidationinfra="http://error.validation.infra.fc.ofss.com"
xmlns:opsapp="http://ops.app.fc.ofss.com"
xmlns:responseservice="http://response.service.fc.ofss.com">
- <ns11:return>
  <responseservice:configVersionId>104</responseservice:configVersionId>
- <responseservice:status>
  <responseservice:errorCode>0</responseservice:errorCode>
  <responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016075018875027</responseservice:interna
lReferenceNumber>
```

```
  <responseservice:isOverriden>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>false</responseservice:isServiceChargeAppl
ied>
- <responseservice:postingDate>
  <datatype:dateString>20130228000000</datatype:dateString>
  <datatype:month>2</datatype:month>
  <datatype:monthDate>228</datatype:monthDate>
  <datatype:monthDateTime>228000000</datatype:monthDateTime>
  <datatype:timestamp>2013-02-28T00:00:00+05:30</datatype:timestamp>
  <datatype:year>2013</datatype:year>
  </responseservice:postingDate>
  <responseservice:replyCode>0</responseservice:replyCode>
  <responseservice:replyText>Operation completed
successfully.</responseservice:replyText>
  <responseservice:spReturnValue>0</responseservice:spReturnValue>
  </responseservice:status>
  </ns11:return>
  </ns11:fetchAllAndExportExportImportResponse>
  </S:Body>
  </S:Envelope>
```

The user can import the required data based on this version number.

8.5 Import All

This operation aims at importing the record for the given taskCode and configVersionNo in the target environment. This method fetches the exported record based on versionNo and taskCode and tries to update if the records exist, else create the new records.

This is carried out by making a web service call to importAll method of ExportImportApplicationService with taskCode, versionNo as input. The request and response xml are as attached.

Import Request

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:exp="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:con="http://context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com">
  <soapenv:Header />
- <soapenv:Body>
- <exp:importAllExportImport>
- <exp:sessionContext>
  <con:bankCode>48</con:bankCode>
  <con:businessUnit>SUNCORP</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>SUNCORP</con:targetUnit>
  <con:transactionBranch>8542</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </exp:sessionContext>
  <exp:taskCode>PM031</exp:taskCode>
  <exp:versionNo>104</exp:versionNo>
  </exp:importAllExportImport>
  </soapenv:Body>
  </soapenv:Envelope>
```

Import Respose

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Body>
- <ns5:importAllExportImportResponse xmlns="http://enumeration.fc.ofss.com"
xmlns:ns2="http://fact.enumeration.fc.ofss.com"
xmlns:ns3="http://context.app.fc.ofss.com"
xmlns:ns4="http://exception.infra.fc.ofss.com"
xmlns:ns5="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:ns6="http://dto.common.domain.framework.fc.ofss.com"
xmlns:ns7="http://datatype.fc.ofss.com" xmlns:ns8="http://ops.app.fc.ofss.com"
xmlns:ns9="http://response.service.fc.ofss.com"
xmlns:ns10="http://error.validation.infra.fc.ofss.com"
xmlns:ns11="http://dto.core.config.app.fc.ofss.com">
- <ns5:return>
  <ns9:errorCode>0</ns9:errorCode>
  <ns9:extendedReply />
  <ns9:internalReferenceNumber>2012132010145535</ns9:internalReferenceNumber>
  <ns9:isOverriden>false</ns9:isOverriden>
  <ns9:isServiceChargeApplied>false</ns9:isServiceChargeApplied>
- <ns9:postingDate>
  <ns7:month>4</ns7:month>
  <ns7:monthDate>425</ns7:monthDate>
  <ns7:monthDateTime>425000000</ns7:monthDateTime>
  <ns7:timestamp>2012-04-25T00:00:00+05:30</ns7:timestamp>
  <ns7:year>2012</ns7:year>
  </ns9:postingDate>
  <ns9:replyCode>0</ns9:replyCode>
  <ns9:replyText>Operation completed successfully.</ns9:replyText>
  <ns9:spReturnValue>0</ns9:spReturnValue>
  </ns5:return>
  </ns5:importAllExportImportResponse>
  </S:Body>
  </S:Envelope>
```

8.6 Config Compare

This operation is used to compare Domain Objects, with same key, for a given taskCode. It aims at comparing the entities from two databases which are termed as TO and FROM database. The comparison is such as it contains following information:

- Present only in TO database (presently working environment)
- Present only in FROM database (configurable DB environment)
- Present in both, but data is different

In the whole set of operations, Export and Import can be performed either by screen or by webservice. For performing import using DB datastore, the reference DataSource needs to be configured in the target environment (the reference datasource is initially configured at the time of installation), which points to the data base where export has been performed. For ExportAll and ImportAll there is a common service ExportImportApplicationService which have the operation to perform the duties.

8.7 Data Store Configuration

The Data Store for Config operations can be configured to either Database or File. The user has the option to choose any one of the two data store configurations. The exported response will be stored in database or file as per this configuration.

8.7.1 DB Data Store

This configuration stores the exported data to database. For using this configuration, following changes have to be made:

1. In **FLX_FW_CONFIG_ALL_B** table, maintain **DataSourceType=File**

   select ***from flx_fw_config_all_b** where **category_id='DataSourceDestination'** and **prop_id='DataSourceType'**

> **Note:** No separate configuration is required for export and import in case of DB Data Store.

8.7.2 File Data Store

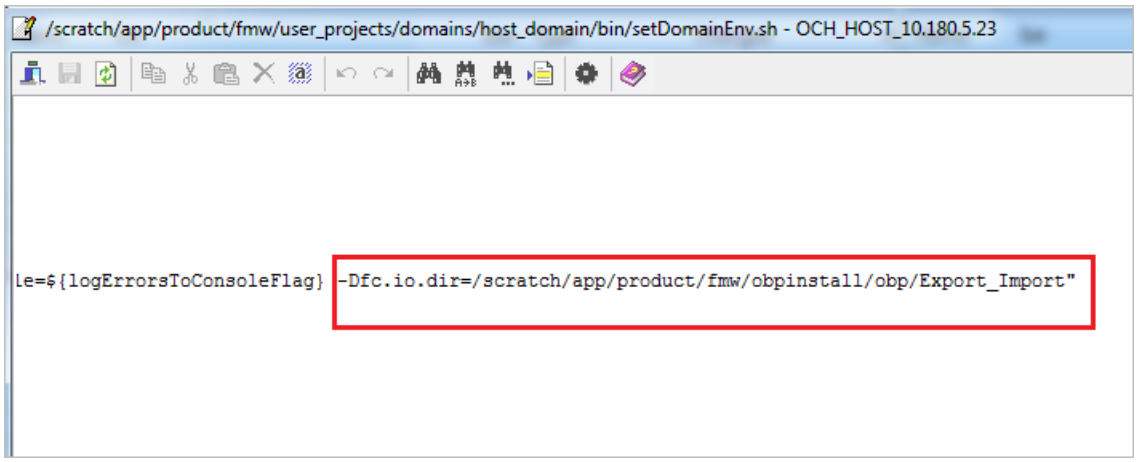
This configuration stores the exported data to file. For using this configuration, following changes have to be made:

1. In **FLX_FW_CONFIG_ALL_B table**, maintain **DataSourceType=FILE**.

   select * **from flx_fw_config_all_b** where **category_id='DataSourceDestination'** and **prop_id='DataSourceType'**

2. Set the location of system property **fc.io.dir** in **setDomainEnv.sh** to a valid directory.

Figure 8–1 File Data Store



8.7.2.1 Configuration for Export

The configuration for export requires the following changes:

1. Set **ExportLoggingPath** variable in **FLX_FW_CONFIG_ALL_B** to the directory where the exported files are needed to be stored. This is relative path with respect to **fc.io.dir/runarea/BusinessUnit**.

   select * **from flx_fw_config_all_b** where **category_id='DataSourceDestination'** and **prop_id='ExportLoggingPath'**

   If **fc.io.dir** is set to */scratch/app/product/fmw/obpinstall/obp/Export_Import*, **ExportLoggingPath** is set to **export** and **business unit** is **DEMO_BANK**, then the files will be stored at */scratch/app/product/fmw/obpinstall/obp/Export_Import/runarea/DEMO_BANK/export.*

8.7.2.2 Configuration for Import

The configuration for import requires the following changes:

1. Set **ImportLoggingPath** variable in **FLX_FW_CONFIG_ALL_B** to the directory from where the exported response has to imported.

   select * **from flx_fw_config_all_b** where **category_id='DataSourceDestination'** and **prop_id='ImportLoggingPath'**

   /scratch/app/product/fmw/obpinstall/obp/Export_Import/runarea/DEMO_BANK/export will be set as **ImportLoggingPath** in our case.

8.8 How to Export Records

Export Operations can be performed using screen or webservice. This operation can be used to export either a single record or multiple records based on the requirement.

In case of DB Data Store, exported data is stored in **flx_ops_config_data_item** and for File Data Store, exported files are generated at the path specified for export configuration. For more information, see Chapter 8.7, "Data Store Configuration".

8.8.1 Exporting Single Record

This operation is used to export single record of an entity.

Let us assume the configuration is done for **DATABASE**, so the data gets stored in **flx_ops_config_data_item** table of the source database. The entity inquiry response gets stored as a serialized byte into the database.

Export operation can be carried out as a webservice call for the export operation of the specific taskcode. A single record of business configuration can be exported using the service **<BusinessConfiguration>ApplicationService**, which provides a '**fetch<BusinessConfiguration>AndExport**' method.

The request parameters to this service are:

- SessionContext
- <BusinessConfiguration>DTO - Representing the key of the record to be exported.

  In response of the service call, it returns '**configVersionNo**'. This 'configVersionNo' will be used to import this record into the target environment.

Sample request and response are as follows:

Export Single Request

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:act="http://action.service.ep.app.fc.ofss.com/ActivityEventActionApplication
Service" xmlns:con="http://context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com"
xmlns:dto="http://dto.common.domain.framework.fc.ofss.com"
xmlns:dat="http://datatype.fc.ofss.com"
xmlns:act1="http://action.dto.ep.app.fc.ofss.com"
xmlns:rule="http://rule.action.dto.ep.app.fc.ofss.com"
xmlns:sub="http://subscriber.action.dto.ep.app.fc.ofss.com">
  <soapenv:Header />
- <soapenv:Body>
- <act:fetchActivityEventActionAndExportActivityEventAction>
- <!-- Optional:
  -->
- <act:sessionContext>
```

```
  <con:bankCode>08</con:bankCode>
  <con:businessUnit>OBP_BU</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>OBP_BU</con:targetUnit>
  <con:transactionBranch>089999</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </act:sessionContext>
- <!-- Optional:
  -->
- <act:activityEventActionDTO>
- <act1:keyDTO>
- <!-- Optional:
  -->
  <act1:actionId>A</act1:actionId>
- <!-- Optional:
  -->

<act1:activityId>com.ofss.fc.domain.lcm.batch.service.BatchCovenantService.process
NotificationForCompliance</act1:activityId>
- <!-- Optional:
  -->
  <act1:eventId>LM_STATUS_COV</act1:eventId>
  </act1:keyDTO>
  </act:activityEventActionDTO>
  </act:fetchActivityEventActionAndExportActivityEventAction>
  </soapenv:Body>
  </soapenv:Envelope>
```

Export Single Response

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Header>
  <work:WorkContext
xmlns:work="http://oracle.com/weblogic/soap/workarea/">rO0ABXdWABx3ZWJsb2dpYy5hcHA
ub2JwLXdlYnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrYXJlYS5TdHJpbmdXb3JrQ29udGV4dAAJM
i4yLjAuMC4wAAA=</work:WorkContext>
  </S:Header>
- <S:Body>
- <ns14:fetchActivityEventActionAndExportActivityEventActionResponse
xmlns:ns14="http://action.service.ep.app.fc.ofss.com/ActivityEventActionApplicatio
nService" xmlns:ns13="http://ep.enumeration.fc.ofss.com"
xmlns:ns12="http://enumeration.fc.ofss.com"
xmlns:exceptioninfra="http://exception.infra.fc.ofss.com"
xmlns:validationdtoapp="http://validation.dto.app.fc.ofss.com"
xmlns:ruleactiondtoepapp="http://rule.action.dto.ep.app.fc.ofss.com"
xmlns:datatype="http://datatype.fc.ofss.com"
xmlns:contextapp="http://context.app.fc.ofss.com"
xmlns:dtocommondomainframework="http://dto.common.domain.framework.fc.ofss.com"
xmlns:errorvalidationinfra="http://error.validation.infra.fc.ofss.com"
xmlns:actiondtoepapp="http://action.dto.ep.app.fc.ofss.com"
xmlns:responseservice="http://response.service.fc.ofss.com"
xmlns:userdtosmsapp="http://user.dto.sms.app.fc.ofss.com"
xmlns:subscriberactiondtoepapp="http://subscriber.action.dto.ep.app.fc.ofss.com">
- <ns14:return>
  <responseservice:configVersionId>187</responseservice:configVersionId>
- <responseservice:status>
  <responseservice:errorCode>0</responseservice:errorCode>
  <responseservice:extendedReply />
```

```
<responseservice:internalReferenceNumber>2016305031622005</responseservice:interna
lReferenceNumber>
  <responseservice:isOverriden>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>false</responseservice:isServiceChargeAppl
ied>
- <responseservice:postingDate>
  <datatype:dateString>20130228000000</datatype:dateString>
  </responseservice:postingDate>
  <responseservice:replyCode>0</responseservice:replyCode>
  <responseservice:replyText>Operation completed
successfully.</responseservice:replyText>
  <responseservice:spReturnValue>0</responseservice:spReturnValue>
  </responseservice:status>
- <actiondtoepapp:activityEventActionDTO>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:createdBy>ArvindKu</dtocommondomainframework:createdBy>
- <dtocommondomainframework:creationDate>
  <datatype:dateString>20130809000000</datatype:dateString>
  </dtocommondomainframework:creationDate>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>

<dtocommondomainframework:lastUpdatedBy>OFSSUser</dtocommondomainframework:lastUpd
atedBy>
- <dtocommondomainframework:lastUpdatedDate>
  <datatype:dateString>20140721162124</datatype:dateString>
  </dtocommondomainframework:lastUpdatedDate>
  <dtocommondomainframework:version>2</dtocommondomainframework:version>
  <actiondtoepapp:alertName>Covenant Status Complied</actiondtoepapp:alertName>
- <actiondtoepapp:alertTemplate>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
- <actiondtoepapp:keyDTO>
  <actiondtoepapp:id>1</actiondtoepapp:id>
  </actiondtoepapp:keyDTO>
  <actiondtoepapp:importance>CRITICAL</actiondtoepapp:importance>
  <actiondtoepapp:language>ENG</actiondtoepapp:language>
  <actiondtoepapp:name>Email Template</actiondtoepapp:name>
  <actiondtoepapp:urgency>HIGH</actiondtoepapp:urgency>
  </actiondtoepapp:alertTemplate>
  <actiondtoepapp:alertType>MANDATORY</actiondtoepapp:alertType>
- <actiondtoepapp:decisionAgent>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
- <ruleactiondtoepapp:keyDTO>
  <ruleactiondtoepapp:id>0</ruleactiondtoepapp:id>
  </ruleactiondtoepapp:keyDTO>
- <ruleactiondtoepapp:rule>
```

```
<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
  <ruleactiondtoepapp:description>Invokes the default
rule</ruleactiondtoepapp:description>
  <ruleactiondtoepapp:keyDTO />
  <ruleactiondtoepapp:name>defaultRule</ruleactiondtoepapp:name>

<ruleactiondtoepapp:ruleClass>com.ofss.fc.domain.ep.service.action.rule.DefaultRul
eHandler</ruleactiondtoepapp:ruleClass>
  <ruleactiondtoepapp:ruleEngine>INTERNAL</ruleactiondtoepapp:ruleEngine>
  </ruleactiondtoepapp:rule>
  </actiondtoepapp:decisionAgent>
- <actiondtoepapp:expiryDate>
  <datatype:dateString>20991231000000</datatype:dateString>
  </actiondtoepapp:expiryDate>
  <actiondtoepapp:isConditional>false</actiondtoepapp:isConditional>
  <actiondtoepapp:isRetryAllowed>true</actiondtoepapp:isRetryAllowed>
  <actiondtoepapp:isTransactional>false</actiondtoepapp:isTransactional>
- <actiondtoepapp:keyDTO>
  <actiondtoepapp:actionId>A</actiondtoepapp:actionId>

<actiondtoepapp:activityId>com.ofss.fc.domain.lcm.batch.service.BatchCovenantServi
ce.processNotificationForCompliance</actiondtoepapp:activityId>
  <actiondtoepapp:eventId>LM_STATUS_COV</actiondtoepapp:eventId>
  </actiondtoepapp:keyDTO>
  <actiondtoepapp:maxRetryCount>2</actiondtoepapp:maxRetryCount>
- <actiondtoepapp:recipientMessageTemplates>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
  <subscriberactiondtoepapp:amount>0</subscriberactiondtoepapp:amount>
  <subscriberactiondtoepapp:bankerType>NA</subscriberactiondtoepapp:bankerType>

<subscriberactiondtoepapp:conditional>false</subscriberactiondtoepapp:conditional>
- <subscriberactiondtoepapp:decisionAgent>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
- <ruleactiondtoepapp:keyDTO>
  <ruleactiondtoepapp:id>0</ruleactiondtoepapp:id>
  </ruleactiondtoepapp:keyDTO>
- <ruleactiondtoepapp:rule>

<dtocommondomainframework:auditSequence>1</dtocommondomainframework:auditSequence>

<dtocommondomainframework:generatedPackageId>false</dtocommondomainframework:gener
atedPackageId>
  <dtocommondomainframework:version>1</dtocommondomainframework:version>
  <ruleactiondtoepapp:description>Invokes the default
rule</ruleactiondtoepapp:description>
```

```
  <ruleactiondtoepapp:keyDTO />
  <ruleactiondtoepapp:name>defaultRule</ruleactiondtoepapp:name>

<ruleactiondtoepapp:ruleClass>com.ofss.fc.domain.ep.service.action.rule.DefaultRul
eHandler</ruleactiondtoepapp:ruleClass>
  <ruleactiondtoepapp:ruleEngine>INTERNAL</ruleactiondtoepapp:ruleEngine>
  </ruleactiondtoepapp:rule>
  </subscriberactiondtoepapp:decisionAgent>
- <subscriberactiondtoepapp:keyDTO>
  <subscriberactiondtoepapp:actionId>A</subscriberactiondtoepapp:actionId>

<subscriberactiondtoepapp:activityId>com.ofss.fc.domain.lcm.batch.service.BatchCov
enantService.processNotificationForCompliance</subscriberactiondtoepapp:activityId
>

<subscriberactiondtoepapp:destinationType>EMAIL</subscriberactiondtoepapp:destinat
ionType>
  <subscriberactiondtoepapp:eventId>LM_STATUS_
COV</subscriberactiondtoepapp:eventId>
  <subscriberactiondtoepapp:messageTemplateId>LCM_Covenant status is
Complied</subscriberactiondtoepapp:messageTemplateId>

<subscriberactiondtoepapp:subscriberType>PARTY</subscriberactiondtoepapp:subscribe
rType>

<subscriberactiondtoepapp:subscriberValue>CUSTOMER</subscriberactiondtoepapp:subsc
riberValue>
  </subscriberactiondtoepapp:keyDTO>

<subscriberactiondtoepapp:recipientType>INTERNAL</subscriberactiondtoepapp:recipie
ntType>
  </actiondtoepapp:recipientMessageTemplates>
  </actiondtoepapp:activityEventActionDTO>
  </ns14:return>
  </ns14:fetchActivityEventActionAndExportActivityEventActionResponse>
  </S:Body>
  </S:Envelope>
```

8.8.2 Exporting All Records

This operation is used to export all the entities of a given task code. The exported package can then be replicated into the target environment. All records of a Business configuration entity can be exported using the **FetchAllAndExport** method of **ExportImportApplicationService**.

The request parameters to this service are:

- **SessionContext**
- **TaskCode**

A '**configVersionNo**' is returned in the response. This '**configVersionNo**' will be used as an identifier to trigger an import into the target environment.

Sample request and response are as follows:

Export All Request

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:exp="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:con="http://context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com">
```

```
  <soapenv:Header />
- <soapenv:Body>
- <exp:fetchAllAndExportExportImport>
- <!-- Optional:
  -->
- <exp:sessionContext>
  <con:bankCode>08</con:bankCode>
  <con:businessUnit>OBP_BU</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>OBP_BU</con:targetUnit>
  <con:transactionBranch>089999</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </exp:sessionContext>
  <exp:taskCode>AL04</exp:taskCode>
  </exp:fetchAllAndExportExportImport>
  </soapenv:Body>
  </soapenv:Envelope>
```

Export All Response

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Header>
  <work:WorkContext
xmlns:work="http://oracle.com/weblogic/soap/workarea/">rO0ABXdWABx3ZWJsb2dpYy5hcHA
ub2JwLXdlYnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrYXJlYS5TdHJpbmdXb3JrQ29udGV4dAAJM
i4yLjAuMC4wAAA=</work:WorkContext>
  </S:Header>
- <S:Body>
- <ns13:fetchAllAndExportExportImportResponse
xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationServic
e" xmlns:ns12="http://ops.enumeration.fc.ofss.com"
xmlns:ns11="http://fact.enumeration.fc.ofss.com"
xmlns:ns10="http://enumeration.fc.ofss.com"
xmlns:exceptioninfra="http://exception.infra.fc.ofss.com"
xmlns:validationdtoapp="http://validation.dto.app.fc.ofss.com"
xmlns:datatype="http://datatype.fc.ofss.com"
xmlns:contextapp="http://context.app.fc.ofss.com"
xmlns:dtocoreseedopsapp="http://dto.core.seed.ops.app.fc.ofss.com"
xmlns:dtocommondomainframework="http://dto.common.domain.framework.fc.ofss.com"
xmlns:errorvalidationinfra="http://error.validation.infra.fc.ofss.com"
xmlns:opsapp="http://ops.app.fc.ofss.com"
xmlns:responseservice="http://response.service.fc.ofss.com">
- <ns13:return>
  <responseservice:configVersionId>186</responseservice:configVersionId>
- <responseservice:status>
  <responseservice:errorCode>0</responseservice:errorCode>
  <responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016305031622003</responseservice:interna
lReferenceNumber>
  <responseservice:isOverriden>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>false</responseservice:isServiceChargeAppl
ied>
- <responseservice:postingDate>
  <datatype:dateString>20130228000000</datatype:dateString>
  </responseservice:postingDate>
  <responseservice:replyCode>0</responseservice:replyCode>
  <responseservice:replyText>Operation completed
```

```
successfully.</responseservice:replyText>
  <responseservice:spReturnValue>0</responseservice:spReturnValue>
  </responseservice:status>
  </ns13:return>
  </ns13:fetchAllAndExportExportImportResponse>
  </S:Body>
  </S:Envelope>
```

In case of DB Data Store, exported data is stored in **flx_ops_config_data_item** and for File Data Store, exported files are generated at the path specified for export configuration. For more information, see Chapter 8.7, "Data Store Configuration".

Figure 8–2 Exported Data

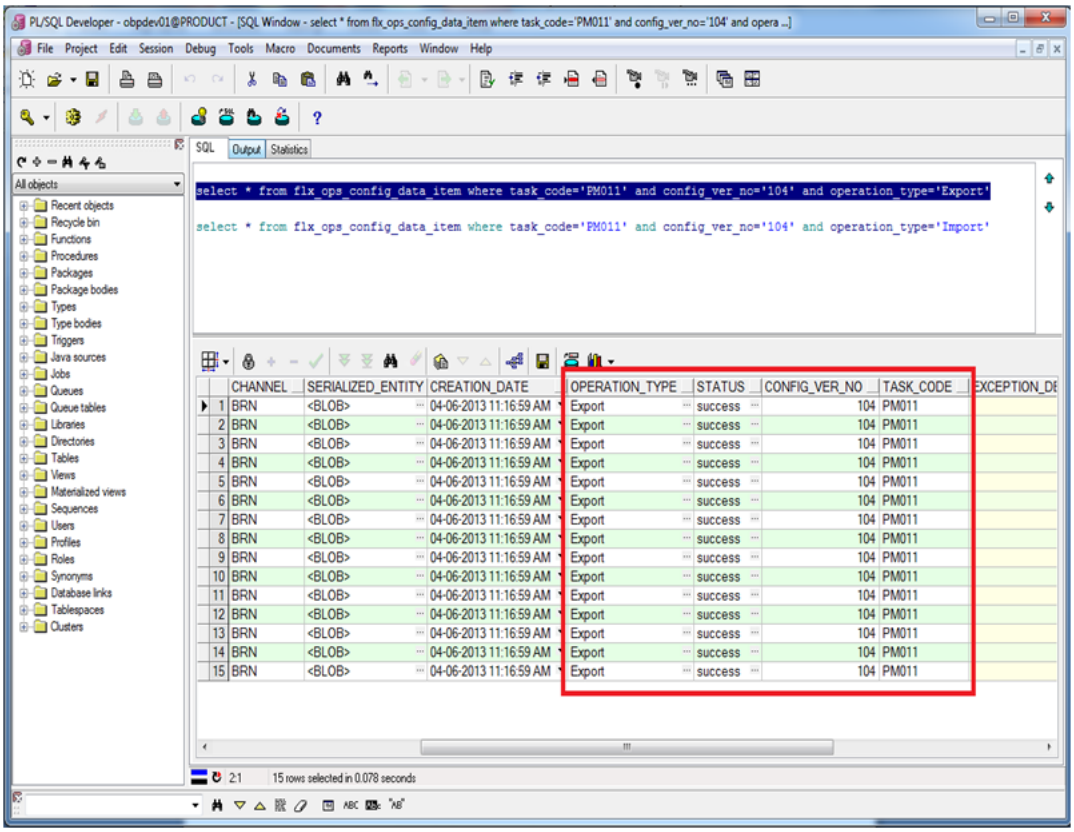
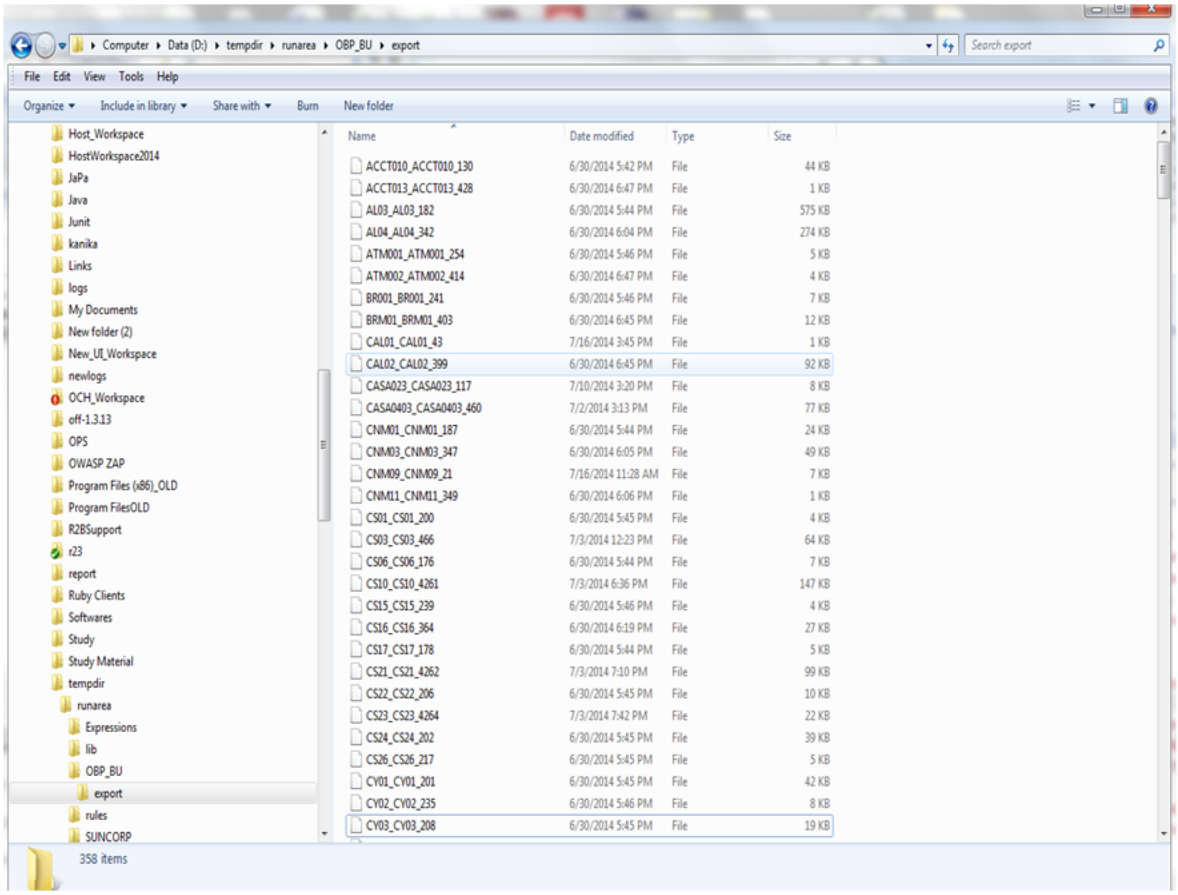


Figure 8–3 Exported Files



8.9 How to Import Records

Import Operations can be performed using webservice clients. This operation can be used to import either a single record or multiple records based on the requirement.

8.9.1 Importing Single Record

This operation is used to import single record of an configuration.

8.9.1.1 Using API Client

A single record of a business configuration entity can be imported using the **ExportImportApplicationService**, which provides an '**importAll**' method.

The request parameters to this service are:

- SessionContext
- TaskCode
- configVersionNo (from Export Single Record response)

The steps to import single record using API client are same as importing all records. These are mentioned in the further section.

Sample request and response are as below:

Import All Request

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:exp="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:con="http://context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com">
  <soapenv:Header />
- <soapenv:Body>
- <exp:importAllExportImport>
- <!-- Optional:
  -->
- <exp:sessionContext>
  <con:bankCode>08</con:bankCode>
  <con:businessUnit>OBP_BU</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>OBP_BU</con:targetUnit>
  <con:transactionBranch>089999</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </exp:sessionContext>
  <exp:taskCode>AL04</exp:taskCode>
  <exp:versionNo>186</exp:versionNo>
  </exp:importAllExportImport>
  </soapenv:Body>
  </soapenv:Envelope>
```

Import All Response

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Header>
  <work:WorkContext
xmlns:work="http://oracle.com/weblogic/soap/workarea/">rO0ABXdWABx3ZWJsb2dpYy5hcHA
ub2JwLXdlYnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrYXJlYS5TdHJpbmdXb3JrQ29udGV4dAAJM
i4yLjAuMC4wAAA=</work:WorkContext>
  </S:Header>
- <S:Body>
- <ns13:importAllExportImportResponse
xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationServic
e" xmlns:ns12="http://ops.enumeration.fc.ofss.com"
xmlns:ns11="http://fact.enumeration.fc.ofss.com"
xmlns:ns10="http://enumeration.fc.ofss.com"
xmlns:exceptioninfra="http://exception.infra.fc.ofss.com"
xmlns:validationdtoapp="http://validation.dto.app.fc.ofss.com"
xmlns:datatype="http://datatype.fc.ofss.com"
xmlns:contextapp="http://context.app.fc.ofss.com"
xmlns:dtocoreseedopsapp="http://dto.core.seed.ops.app.fc.ofss.com"
xmlns:dtocommondomainframework="http://dto.common.domain.framework.fc.ofss.com"
xmlns:errorvalidationinfra="http://error.validation.infra.fc.ofss.com"
xmlns:opsapp="http://ops.app.fc.ofss.com"
xmlns:responseservice="http://response.service.fc.ofss.com">
- <ns13:return>
  <responseservice:errorCode>0</responseservice:errorCode>
  <responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016305031622004</responseservice:interna
lReferenceNumber>
  <responseservice:isOverriden>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>false</responseservice:isServiceChargeAppl
ied>
```

```
<responseservice:replyCode>0</responseservice:replyCode>
<responseservice:spReturnValue>0</responseservice:spReturnValue>
</ns13:return>
</ns13:importAllExportImportResponse>
</S:Body>
</S:Envelope>
```

8.9.2 Importing All Records

This operation is used to import the records belonging to the given '**TaskCode**' and '**configVersionNo**' into the target environment. This method fetches the exported records based on the '**configVersionNo**' and '**TaskCode**', and upserts the same into the target environment. All records of a Business configuration entity can be imported using the **ImportAll** method of **ExportImportApplicationService**.

The request parameters to this service are:

- TaskCode
- ConfigVersionNo

For performing the import operation, the '**Config Data Source**' needs to be configured in the target environment, this datasource points to the database of the reference environments.

Sample request and response are as below:

Import All Request

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:exp="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationService
" xmlns:con="http://context.app.fc.ofss.com"
xmlns:exc="http://exception.infra.fc.ofss.com">
  <soapenv:Header />
- <soapenv:Body>
- <exp:importAllExportImport>
- <!-- Optional:
  -->
- <exp:sessionContext>
  <con:bankCode>08</con:bankCode>
  <con:businessUnit>OBP_BU</con:businessUnit>
  <con:channel>BRN</con:channel>
  <con:marketEntity>SUN01</con:marketEntity>
  <con:postingDateText>20130228000000</con:postingDateText>
  <con:targetUnit>OBP_BU</con:targetUnit>
  <con:transactionBranch>089999</con:transactionBranch>
  <con:userId>OFSSUser</con:userId>
  </exp:sessionContext>
  <exp:taskCode>AL04</exp:taskCode>
  <exp:versionNo>186</exp:versionNo>
  </exp:importAllExportImport>
  </soapenv:Body>
  </soapenv:Envelope>
```

Import All Response

```
- <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
- <S:Header>
  <work:WorkContext
xmlns:work="http://oracle.com/weblogic/soap/workarea/">rO0ABXdWABx3ZWJsb2dpYy5hcHA
ub2JwLXdlYnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrYXJlYS5TdHJpbmdXb3JrQ29udGV4dAAJM
i4yLjAuMC4wAAA=</work:WorkContext>
```

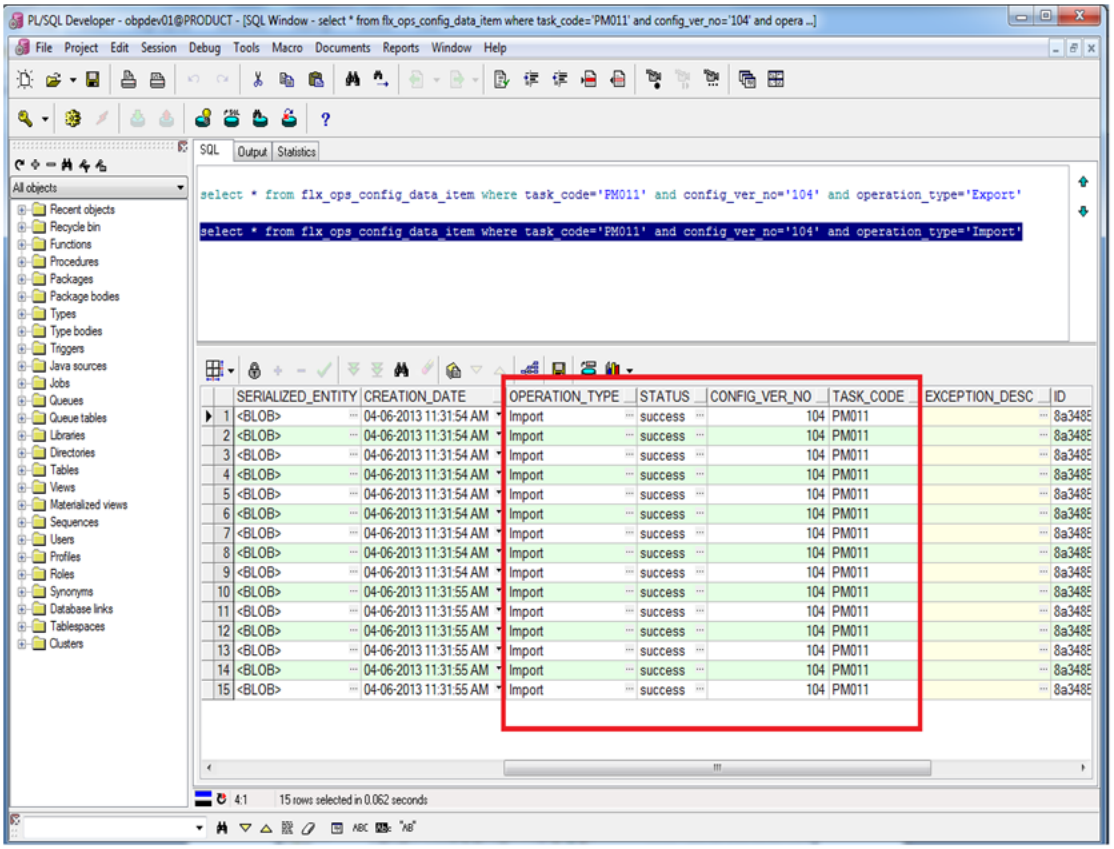
```
  </S:Header>
- <S:Body>
- <ns13:importAllExportImportResponse
xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportApplicationServic
e" xmlns:ns12="http://ops.enumeration.fc.ofss.com"
xmlns:ns11="http://fact.enumeration.fc.ofss.com"
xmlns:ns10="http://enumeration.fc.ofss.com"
xmlns:exceptioninfra="http://exception.infra.fc.ofss.com"
xmlns:validationdtoapp="http://validation.dto.app.fc.ofss.com"
xmlns:datatype="http://datatype.fc.ofss.com"
xmlns:contextapp="http://context.app.fc.ofss.com"
xmlns:dtocoreseedopsapp="http://dto.core.seed.ops.app.fc.ofss.com"
xmlns:dtocommondomainframework="http://dto.common.domain.framework.fc.ofss.com"
xmlns:errorvalidationinfra="http://error.validation.infra.fc.ofss.com"
xmlns:opsapp="http://ops.app.fc.ofss.com"
xmlns:responseservice="http://response.service.fc.ofss.com">
- <ns13:return>
  <responseservice:errorCode>0</responseservice:errorCode>
  <responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016305031622004</responseservice:interna
lReferenceNumber>
  <responseservice:isOverriden>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>false</responseservice:isServiceChargeAppl
ied>
  <responseservice:replyCode>0</responseservice:replyCode>
  <responseservice:spReturnValue>0</responseservice:spReturnValue>
  </ns13:return>
  </ns13:importAllExportImportResponse>
  </S:Body>
  </S:Envelope>
```

Database entry for imported response will be stored in **flx_ops_config_data_item**.

Figure 8–4 Importing Data Using SOAP UI - Storing Response



8.10 Configuration Comparison

This topic describes the details of configuration comparison.

8.10.1 Compare Business Configuration (Fast Path: OPA005)

This page is used to compare two entities on basis of its content.

It provides facility to compare Domain Objects, with same key, for a given task code. It aims at comparing the entities from two database which are termed as **TO** and **FROM** database. The comparison is such as it contains following information:

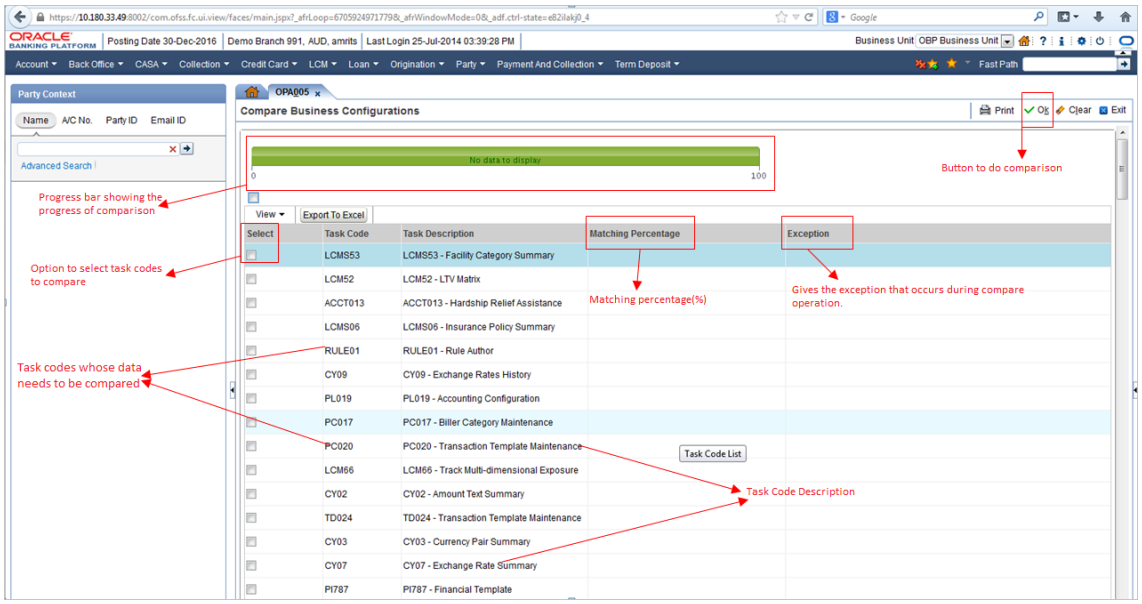
- Present only in TO database (presently working environment).
- Present only in FROM database (configurable DB environment).
- Present in both, but data is different.

8.10.2 Usage

The comparison results among entities can be generated by performing the following steps:

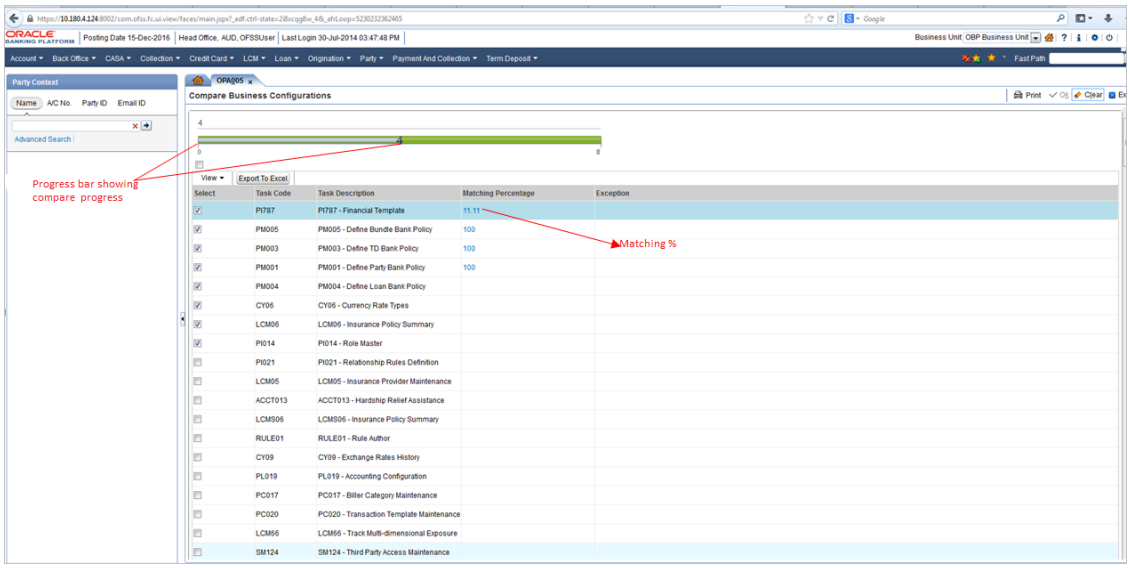
1. Open the OPA005 page, which loads all the entities configured in the table **flx_ops_task_defn**.

Figure 8–5 Entity Comparison



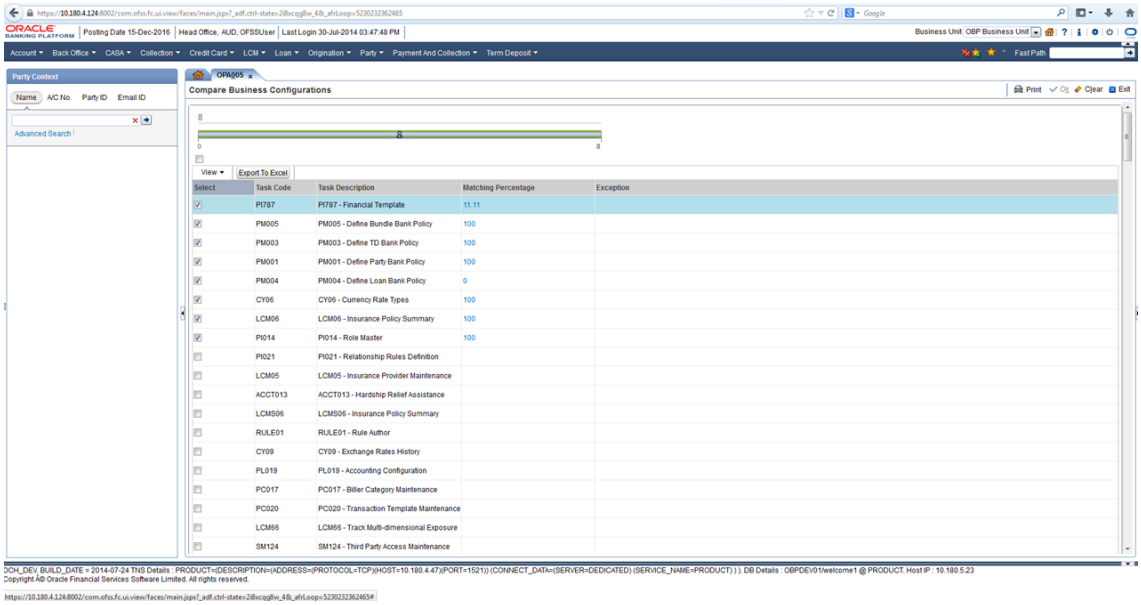
2. Select the option under **Select** column to do the comparison of configuration/configurations which shows the matching % of data in the two environment.

Figure 8–6 Entity Comparison Results



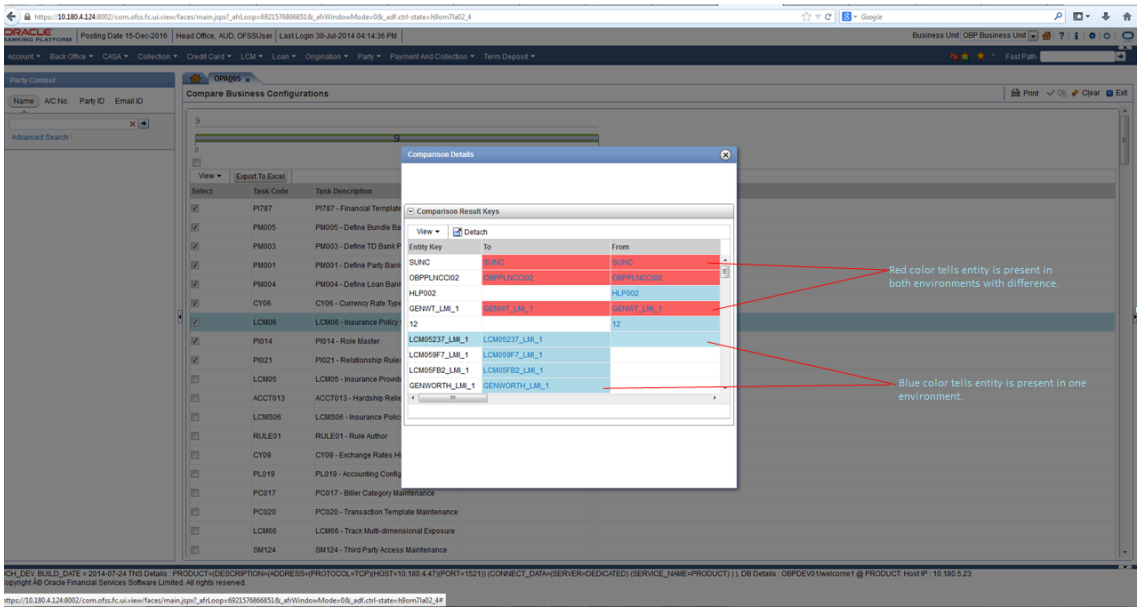
Progress bar denoting that the compare operation has finished.

Figure 8–7 Progress Bar



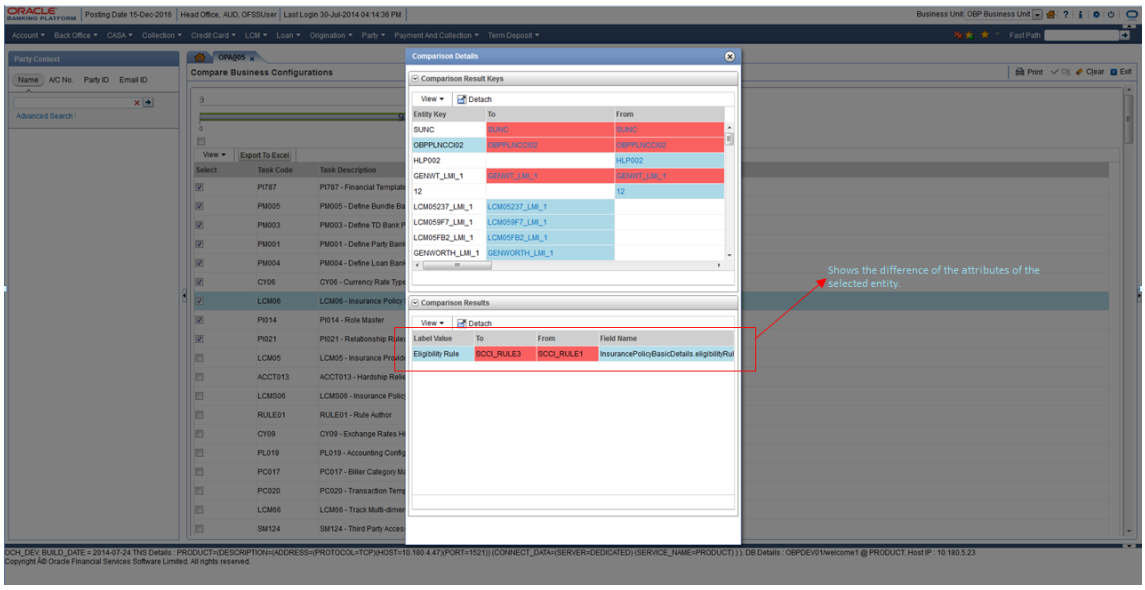
3. Select the % match to launch a pop up which shows the comparison result with different keys associated to it.

Figure 8–8 Comparison Details



4. Select any of the key to see the difference of its associated attributes in the two environment.

Figure 8–9 Attributes Difference



8.11 Application Configuration

This topic describes the application configuration details.

8.11.1 Cache Configuration

Configuration cache is where we cache configuration information (stored in the configuration tables in database or some files) for every application on each server in the farm.

The entire application configuration to be cached is pre-defined in **Preferences.xml**.

Sample entries in Preferences.xml are as below:

Preferences.xml

```
<Preferences>
   <Nodes>
       <Preference name="jdbcpreference"
PreferencesProvider="com.ofss.fc.infra.config.impl.PropertiesFileConfigProvider"
parent=""          propertyFileName="jdbc.properties"  syncTimeInterval="600000"
/>
      <Preference name="ConfigurationVariable"
PreferencesProvider="com.ofss.fc.infra.config.impl.DBBasedPropertyProvider"
parent="jdbcpreference" propertyFileName="select prop_id, prop_value from flx_fw_
config_var_b" syncTimeInterval="600000" />
     <Preference name="ChannelConstants"
PreferencesProvider="com.ofss.fc.infra.config.impl.JavaConstantsConfigProvider"
parent="jdbcpreference"
propertyFileName="com.ofss.fc.common.ChannelConstantsConfiguration"
syncTimeInterval="600000" />
<Preference name="JSONServiceMap" overriddenBy="JSONServiceMapOverride"
PreferencesProvider="com.ofss.fc.infra.config.impl.JavaConstantsConfigProvider"
parent="jdbcpreference"propertyFileName="com.ofss.fc.common.JSONServiceConfig"
syncTimeInterval="600000" />
```

```
.....
</Nodes>
</Preferences>
```

Important parameters in preferences.xml are as follows:

- **PreferencesProvider**: DB based provider, File base providor or Java constant base provider.
- **propertyFileName**: Describes the configuration source. Either sql query, file name or fully qualified Java constant class name.
- **syncTimeInterval**: Refresh time
- **name**: Acts as configuration key in the cache
- **parent**: Enables building the dependency hierarchy
- **overriddenBy**: This parameter specifies the name of preference which will override the current one.