

Oracle® Banking Platform

Administrator Guide

Release 2.12.0.0.0

F43567-01

June 2021

Oracle Banking Platform Administrator Guide, Release 2.12.0.0.0

F43567-01

Copyright © 2011, 2021, Oracle and/or its affiliates.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	21
Audience	21
Documentation Accessibility	21
Organization of the Guide	21
Related Documents	22
Conventions	23
1 User Administration with OIM	25
1.1 Creating Users in Oracle Identity Manager (OIM)	25
1.2 Creating Roles in Oracle Identity Manager (OIM)	29
1.3 Assigning Roles to Users in OIM	33
1.4 Locking Users in OIM	37
1.5 Unlocking Users in OIM	39
1.6 Resetting User Password in OIM	40
1.7 User Management Using the Admin Application	44
1.8 Unlocking Users in Oracle WebLogic Server (OWS) Administration Console ..	51
1.9 Creation of first time user to access OBP	56
2 Approvals Management	61
2.1 Discretionary Pricing Assessment (DPA)	61
2.1.1 Setup Details	61
2.1.1.1 Policy Setup in UI	62
2.1.1.2 SOA Composer Rules Setup	67
2.1.1.3 SMS Setup	78
2.1.2 Performing the Transaction	81
2.2 Discretionary Credit Assessment (DCA)	85

2.2.1 Setup Details	86
2.2.1.1 Policy Setup in UI	86
2.2.1.2 SOA Composer Rules Setup	93
2.2.1.3 SMS Setup	93
2.2.2 Performing the Transaction	96
2.3 Enabling Worklist Authorization	98
2.4 BPM Routing Rules Setup	105
3 Defining Task Configuration Rules	121
3.1 Important Rule Artifacts	121
3.1.1 Rules Dictionary	121
3.1.2 Decision Function	122
3.1.3 Rulesets	123
3.2 Inputs to Decision Function	123
3.2.1 Custom Input Attributes	124
3.3 Output from Decision Function	125
3.3.1 List of Configurable Attributes in Rule Outcome	126
3.3.2 List of Functions to Set Rule Outcome	128
4 Data Management	133
4.1 Batch Execution	133
4.1.1 Database Backup	133
4.1.2 Navigate to End of Day Page	133
4.1.3 Cutoff Category Execution	134
4.1.4 End of Day (EOD) Category Execution	136
4.1.5 Internal System EOD Category Execution	138
4.1.6 Beginning of Day (BOD) Category Execution	141
4.1.7 Housekeeping Category Execution	143

4.1.8 Alert Generation Category Execution	145
4.2 Batch Exception Recovery	147
5 Setting Up The Bank And Branch	151
5.1 Common Services Day 0 Setup	151
5.1.1 Core Maintenances	151
5.1.1.1 Head Office Setup	152
5.1.2 Currency Maintenances	152
5.1.3 Calendar Maintenances	153
5.2 Accounting Day 0 Setup	153
5.3 Product Manufacturing Day 0 Setup	154
6 Setting Up Sales Offer Handoff	157
6.1 Day Zero Setup (Configuration)	157
6.1.1 Changes to be done in OBP	157
6.1.2 Changes to be done in ODI	158
7 Setting Up Product	175
7.1 Creating New Product	175
8 Setting Up Channels	181
8.1 Channel Setup	181
8.2 Front End Processing Interface (FEPI)	182
8.2.1 Installation	183
8.2.2 ATM And POS Trace Logs	183
8.2.3 module.channel or cz.module.channel enabling of logs	183
8.2.4 Multiple Instances	183
9 Application Monitoring Using Administration Application	185
9.1 Dynamic Monitoring Service (DMS)	185
9.1.1 Usage	185

9.1.2 Monitoring Application using the OPA001 page	186
9.1.2.1 Monitoring Application Performance (Fast path: OPA001)	186
9.1.2.1.1 Application Performance Summary	186
9.1.2.1.2 Log Level	187
9.1.2.1.3 Application Performance	187
9.2 Batch Performance Monitoring	192
9.2.1 Use Cases	192
9.2.2 Monitoring Batch Performance Using OPA003 page	194
9.2.2.1 Monitor Batch Performance (Fast path: OPA003)	194
9.2.3 Histogram of Shell Attribute Comparison	203
9.3 ODI Batch Handoff Monitoring	203
10 Application Monitoring Using EM Plugin	207
10.1 Monitoring Application Using EM Plugin	207
10.1.1 Oracle Enterprise Manager (EM)	207
10.1.2 UI	208
10.1.3 Host	210
10.1.4 SOA	212
10.1.5 Security Stack (OID and OAM)	212
10.1.6 Document Generation Outbound Components (Documaker, Oracle Analytics Publisher, IPM)	213
10.1.7 ATM and POS (Point Of Sales) Channels	214
10.1.8 Outbound OFSAA call	214
10.1.9 Monitoring Views	214
10.1.9.1 Batch Monitoring	215
10.1.9.2 Application Monitoring	217
10.1.9.2.1 Application Services	217
10.1.9.2.2 User Interface	219

10.1.9.2.3 Origination User Interface	221
10.1.9.2.4 Integration	223
11 Configuration Export-Import Operations	227
11.1 Objective	227
11.2 Export	227
11.3 Import	227
11.4 Export All	227
11.5 Import All	229
11.6 Config Compare	230
11.7 Data Store Configuration	231
11.7.1 DB Data Store	231
11.7.2 File Data Store	231
11.7.2.1 Configuration for Export	232
11.7.2.2 Configuration for Import	232
11.8 How to Export Records	232
11.8.1 Exporting Single Record	233
11.8.2 Exporting All Records	239
11.9 How to Import Records	242
11.9.1 Importing Single Record	242
11.9.1.1 Using API Client	242
11.9.2 Importing All Records	244
11.10 Configuration Comparison	246
11.10.1 Compare Business Configuration (Fast Path: OPA005)	246
11.10.2 Usage	247
11.11 Application Configuration	249
11.11.1 Cache Configuration	249

12 Batch Shells in OBP	251
12.1 Batch Shells Description	251
12.2 Batch Shells Execution Sequence	277
13 Information Lifecycle Management (ILM)	294
13.1 Configuration	294
13.2 Installation	294
13.2.1 Prepare Scripts	294
13.2.2 Create Tablespace	295
13.2.3 Create Partition Script	295
13.2.4 Run Partition Script	298
13.2.5 Create and Register ADO Policies based on Lifecycle Pattern	298
13.2.6 Verify Registered ADO Policies	299
13.3 Policy Execution	299
14 Transparent Data Encryption (TDE)	300
14.1 Configuration	300
14.2 Installation	300
14.2.1 Prepare Scripts to Encrypt Sensitive Data	301
14.2.2 Create TDE Keystore	301
14.2.3 Edit sqlnet.ora file	302
14.2.4 Run Created Alter Script	302
15 Masking Customer Private Data	304
15.1 Configuration	304
15.2 Installation	305
15.2.1 Prepare Scripts to Encrypt Sensitive Data	305
15.2.2 Create Schema for RO and ERO User	305
15.2.3 Execute Created Scripts through Encryption Tool	306

16 Configure ODI for Inbound Document Upload	308
17 Additional Recommendations	312
17.1 SOA Related	312
17.1.1 Enable Auto Purge Job	312
17.1.2 Enable Lazy Loading	312
17.2 BPM Worklist Related	313
17.2.1 Disable Claim Action from Task Details Page	313
17.2.2 Always Open Human Task Details in External Window	314

List of Figures

Figure 1–1 Creating Users in OIM - Log in	25
Figure 1–2 Creating Users in OIM - Manage Section	26
Figure 1–3 Creating Users in OIM - Click Create	27
Figure 1–4 Creating Users in OIM - Enter User Details	28
Figure 1–5 Enter User Details (Continued)	29
Figure 1–6 Creating Roles in OIM - Manage Section	30
Figure 1–7 Creating Roles in OIM - Click Create	31
Figure 1–8 Creating Roles in OIM - Enter Role Details	32
Figure 1–9 Creating Roles in OIM - Role Created Successfully	33
Figure 1–10 Assigning Roles in OIM - Requesting Roles	34
Figure 1–11 Assigning Roles in OIM - Adding to Cart	35
Figure 1–12 Assigning Roles in OIM - Checkout Cart	36
Figure 1–13 Assigning Roles in OIM - Submit Cart	37
Figure 1–14 Locking Users in OIM	38
Figure 1–15 User Locked Successfully	39
Figure 1–16 Unlocking Users in OIM	40
Figure 1–17 Resetting User Password in OIM	41
Figure 1–18 Resetting User Password in OIM - Manually or Auto-generate	42
Figure 1–19 Resetting User Password in OIM - New Password	43
Figure 1–20 Password Reset Successfully	44
Figure 1–21 Adding a User	45
Figure 1–22 Enter Mandatory Details	46
Figure 1–23 Applying Changes	47
Figure 1–24 Adding User to a Group	48

Figure 1–25 Available and Assigned Roles	49
Figure 1–26 Adding User to Assigned Roles Table	50
Figure 1–27 Save Changes	51
Figure 1–28 OWS Log in	52
Figure 1–29 base_domain	53
Figure 1–30 Security tab	54
Figure 1–31 Unlock User	55
Figure 1–32 User Successfully Unlocked	56
Figure 1–33 Log in Oracle Fusion Middleware Control	57
Figure 1–34 Click Application Roles	58
Figure 1–35 Select Administrators Role	58
Figure 1–36 Add Principal	59
Figure 2–1 Log in to OBP UI to Configure Policies	62
Figure 2–2 Search for SM502 in Fast Path	63
Figure 2–3 Policy Management	64
Figure 2–4 Enter Service for Policy Definition	64
Figure 2–5 Effect of the Policy	65
Figure 2–6 Action of the Policy	66
Figure 2–7 Conditions of the Policy	67
Figure 2–8 Log in to SOA Composer	68
Figure 2–9 Find the Rules for Service	69
Figure 2–10 Filter Rules	70
Figure 2–11 Searching Specific Process	71
Figure 2–12 Creating the Rule	72
Figure 2–13 Selecting Rules File	73
Figure 2–14 Adding Attributes to the Rule File	74

Figure 2–15 Adding Attributes to the Rule File	75
Figure 2–16 Validating Rules File	76
Figure 2–17 Saving Rules File	77
Figure 2–18 Publishing Rules File	78
Figure 2–19 Search for Service Using Entire Service Name	79
Figure 2–20 Select the Service to be Configured	80
Figure 2–21 Severity Tab - Add Severity	81
Figure 2–22 Initiating Transaction	82
Figure 2–23 Fee Negotiation	82
Figure 2–24 Log in to Approval Worklist	83
Figure 2–25 Approve or Reject Work item	84
Figure 2–26 Viewing Status of Work item	85
Figure 2–27 Log in to OBP UI to Configure Policies	87
Figure 2–28 Search for SM502 in Fast Path	88
Figure 2–29 Policy Management	89
Figure 2–30 Enter Service for Policy Definition	90
Figure 2–31 Effect of the Policy	91
Figure 2–32 Action of the Policy	92
Figure 2–33 Conditions of the Policy	93
Figure 2–34 Search for Service Using Entire Service Name	94
Figure 2–35 Select the Service to be Configured	95
Figure 2–36 Approval Checks tab - MATRIX_AUTH	96
Figure 2–37 Initiating Origination Task Flow	97
Figure 2–38 Approving the Task	98
Figure 2–39 Select the Service to be Configured	99
Figure 2–40 Search for Service using TASK CODE + Search text (in case of non Origination)	100

Figure 2–41 Approval Checks tab - Add Approval Checks	101
Figure 2–42 Dual Control	102
Figure 2–43 Add New Severity	103
Figure 2–44 Save Severity Details	104
Figure 2–45 Update Severity and Save	105
Figure 2–46 Worklist App - Administration Link	106
Figure 2–47 Task Configuration	107
Figure 2–48 Search Task Types	108
Figure 2–49 Task Editing	109
Figure 2–50 Configuring Rules - Edit Icon	110
Figure 2–51 Configuring Rules - Rules tab	111
Figure 2–52 Expand Rule	112
Figure 2–53 Advanced Settings	113
Figure 2–54 Rule Expansion	114
Figure 2–55 Enabling Rule	115
Figure 2–56 Adding New Row of Condition	116
Figure 2–57 Expanding Payload	117
Figure 2–58 Selecting Fact	118
Figure 2–59 Updating Fact Value	119
Figure 2–60 Select Approver Group	120
Figure 3–1 SOA Composer - Open Rules Dictionary Browser	121
Figure 3–2 SOA Composer – Selecting Rules Dictionary	122
Figure 3–3 SOA Composer – Selecting Decision Function	122
Figure 3–4 SOA Composer - Adding Rulesets to Decision Function	123
Figure 3–5 SOA Composer - Viewing a ruleset	123
Figure 3–6 SOA Composer - Viewing inputs to a Decision Function	124

Figure 3–7 SOA Composer – Example Usage of Custom Attributes	125
Figure 3–8 SOA Composer – Viewing Output of a Decision Function	125
Figure 3–9 SOA Composer – Example for adding Stage Participant	127
Figure 3–10 SOA Composer – Example usage of custom attributes	128
Figure 4–1 End of Day (Fast path:EOD10)	134
Figure 4–2 Cutoff Category - Not Started	135
Figure 4–3 Cutoff Category - Start	135
Figure 4–4 Cutoff Category - Complete	136
Figure 4–5 EOD Category - Not Started	137
Figure 4–6 EOD Category - Start	137
Figure 4–7 EOD Category - In Progress	138
Figure 4–8 EOD Category - Complete	138
Figure 4–9 Internal System EOD Category - Not Started	139
Figure 4–10 Internal System EOD Category - Start	140
Figure 4–11 Internal System EOD Category - In Progress	140
Figure 4–12 Internal System EOD Category - Complete	141
Figure 4–13 BOD Category - Not Started	142
Figure 4–14 BOD Category - Started	142
Figure 4–15 BOD Category - In Progress	143
Figure 4–16 BOD Category - Completed	143
Figure 4–17 Housekeeping Category - Not Started	144
Figure 4–18 Housekeeping Category - In Progress	145
Figure 4–19 Housekeeping Category - Completed	145
Figure 4–20 Alert Generation Category - Not Started	146
Figure 4–21 Alert Generation Category - In Progress	147
Figure 4–22 Exception Details	148

Figure 4–23 Exception Record Details	149
Figure 4–24 Exception record in Worklist application	150
Figure 6–1 Repository Connection Information	159
Figure 6–2 Physical Architecture - New Data Server	160
Figure 6–3 Data Server - Definition	161
Figure 6–4 Data Server - JDBC	161
Figure 6–5 Create New Target Data Server	162
Figure 6–6 New Physical Schema	162
Figure 6–7 Physical Schema - Definition	163
Figure 6–8 Physical Schema - Target Data Server	164
Figure 6–9 Physical Schema List	164
Figure 6–10 Logical Architecture - New Logical Schema	165
Figure 6–11 Logical Schema - Definition	166
Figure 6–12 Physical Schema - Set Context	166
Figure 6–13 New Model	166
Figure 6–14 OBP_SRC Model - Definition	167
Figure 6–15 OBP_SRC Model - Reverse Engineer	167
Figure 6–16 OBI_DEMO_TARGET Model - Definition	168
Figure 6–17 OBI_DEMO_TARGET Model - Reverse Engineer	168
Figure 6–18 Click Reverse Engineer	169
Figure 6–19 OBP_SRC DB Tables	169
Figure 6–20 OBI_DEMO_TARGET DB Tables	170
Figure 6–21 New Variable	170
Figure 6–22 Variable - Definition	171
Figure 6–23 Variable - Refreshing	171
Figure 6–24 Project Settings	172

Figure 6–25 PKG_PM_SALES_OFFER_STG	173
Figure 6–26 PKG_PM_SALES_OFFER_STG Properties	174
Figure 6–27 Export the .csv File	174
Figure 7–1 Log on to UCM	175
Figure 7–2 Sitemap	176
Figure 7–3 Administration - Product	176
Figure 7–4 Products View	177
Figure 7–5 Product List	177
Figure 7–6 Product Activities	178
Figure 7–7 More Info - Select Category	178
Figure 7–8 More Info - Select Type	179
Figure 8–1 Transaction Message Flows	181
Figure 9–1 Developers	185
Figure 9–2 IT Technical Staff	186
Figure 9–3 Monitoring Application Performance	186
Figure 9–4 Application Performance Summary	187
Figure 9–5 Log Level	187
Figure 9–6 Alert State	188
Figure 9–7 Select Task Code	190
Figure 9–8 Selection of Desired Transaction	190
Figure 9–9 Transaction Details	191
Figure 9–10 Transaction Metrics	191
Figure 9–11 Alert and Trend Details	191
Figure 9–12 Failure Events	192
Figure 9–13 Developers	193
Figure 9–14 IT Technical Staff	193

Figure 9–15 IT Technical Staff - Monitor Batch Stats	193
Figure 9–16 Batch Performance Monitoring	194
Figure 9–17 Batch Performance Monitoring - Shell Details	194
Figure 9–18 Batch Performance Monitoring - Relative Performance Summary	195
Figure 9–19 Shell Details	197
Figure 9–20 Shell Details - DDA Standing Instructions	197
Figure 9–21 View of Batch Run	198
Figure 9–22 Stream Based Shells	199
Figure 9–23 Exception Log	199
Figure 9–24 Report Based Shells	200
Figure 9–25 Status of Report Based Shell	201
Figure 9–26 Exception Report	201
Figure 9–27 Sample Report	202
Figure 9–28 Exception Log Table	202
Figure 9–29 Graphs	203
Figure 9–30 Input Parameters for Batch Handoff	204
Figure 9–31 Execution Unit	205
Figure 9–32 Abort Statistics	205
Figure 9–33 Failure Error Description	206
Figure 10–1 Oracle Enterprise Manager	208
Figure 10–2 UI Cluster in EM	209
Figure 10–3 WebLogic Domain for UI	209
Figure 10–4 Metrics Chart	210
Figure 10–5 Host Cluster in EM	211
Figure 10–6 Host Target in EM	211
Figure 10–7 Metrics Chart	212

Figure 10–8 Viewing Process List	212
Figure 10–9 OID WebLogic Domain	213
Figure 10–10 Document Generation Status	213
Figure 10–11 Oracle Analytics Publisher Deployment	214
Figure 10–12 EM Monitoring	214
Figure 10–13 Web Monitoring	214
Figure 10–14 Database Server Info	215
Figure 10–15 Batch Monitoring Status	216
Figure 10–16 Batch Configuration	216
Figure 10–17 WebLogic Service Info	217
Figure 10–18 Application Metrics of Application Services for all servers in cluster	218
Figure 10–19 Application Metrics of Application Services for selected server	218
Figure 10–20 Application Metric for all UI servers in cluster	220
Figure 10–21 Application Metrics of UI components for selected server	220
Figure 10–22 Application Metrics of Origination UI Components for all UI servers in cluster	222
Figure 10–23 Application Metrics of Origination UI components for selected server	222
Figure 10–24 Application Metrics of all outbound services called from all host serv- ers in cluster	224
Figure 10–25 Application Metrics of all outbound services called from selected server	224
Figure 11–1 File Data Store	232
Figure 11–2 Exported Data	241
Figure 11–3 Exported Files	242
Figure 11–4 Importing Data Using SOAP UI - Storing Response	246
Figure 11–5 Entity Comparison	247
Figure 11–6 Entity Comparison Results	247

Figure 11–7 Progress Bar	248
Figure 11–8 Comparison Details	248
Figure 11–9 Attributes Difference	249
Figure 13–1 Partition Script - SQL Statement	297
Figure 13–2 Utility Table Creation Script	298
Figure 13–3 Verify ADO Policies	299
Figure 16–1 Credentials	309
Figure 16–2 Example of premissions	309
Figure 16–3 Connection details	310
Figure 17–1 Auto Purge	312
Figure 17–2 Lazy Loading Settings	313
Figure 17–3 Claim Action	313
Figure 17–4 Enable External Window option	314

List of Tables

Table 2–1 List of Functions for Fee Configuration	61
Table 2–2 List of Functions for UDM Configuration	61
Table 2–3 Facts Required for the MCD Approvals	85
Table 9–1 Alert State	188
Table 9–2 Category Details	195
Table 9–3 Shell Details	196
Table 9–4 Stream Details	198
Table 9–5 Reports Table	200
Table 10–1 Notations in EM	207
Table 10–2 Details of the Application Metrics table of Application Services	218
Table 10–3 Details of the Application Metrics table of UI Components	220
Table 10–4 Details of the Application Metrics table of Origination UI Components	222
Table 10–5 Details of the Application Metrics table of all Outbound Services	224
Table 12–1 Batch Shell Descriptions	251
Table 12–2 Shell Execution Sequence	277
Table 13–1 Values for ILM Configuration	294
Table 14–1 TDE Configuration	300
Table 15–1 TDE Configuration	304

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 setup.

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

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Organization of the Guide](#)
- [Related Documents](#)
- [Conventions](#)

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 that have purchased support 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 User Administration with OIM](#)

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, worklist authorization related activities, and SOA Composer rules setup to be performed as an administrator.

[Chapter 3 Defining Task Configuration Rules](#)

This chapter describes various configurations that can be done for human tasks.

[Chapter 4 Data Management](#)

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

[Chapter 5 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, which the administrator can use to perform a bank and a branch setup.

[Chapter 6 Setting Up Sales Offer Handoff](#)

This chapter describes the sales offer handoff related activities to be performed as an administrator.

[Chapter 7 Setting Up Product](#)

This chapter describes the process of setting up the product. Products need to be configured manually from the Sales Offer Handoff file. The steps for manually adding Product in OCH are explained in this chapter.

[Chapter 8 Setting Up Channels](#)

This chapter describes the process of setting up the channels.

[Chapter 9 Application Monitoring Using Administration Application](#)

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

[Chapter 10 Application Monitoring Using EM Plugin](#)

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

[Chapter 11 Configuration Export-Import Operations](#)

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

[Chapter 12 Batch Shells in OBP](#)

This chapter describes the batch shells used in Oracle Banking Platform and their execution sequence.

[Chapter 13 Information Lifecycle Management \(ILM\)](#)

This chapter describes the configuration, installation, and policy setup of Information Lifecycle Management (ILM).

[Chapter 14 Transparent Data Encryption \(TDE\)](#)

This chapter describes the configuration, installation, and policy setup of Transparent Data Encryption (TDE).

[Chapter 15 Masking Customer Private Data](#)

This chapter describes the configuration, installation, and policy setup to mask customer private data categories as sensitive or Personally Identifiable Information (PII).

[Chapter 16 Configure ODI for Inbound Document Upload](#)

This chapter provides the steps to configure ODI for Inbound Document Upload

[Chapter 17 Additional Recommendations](#)

This chapter provides specific recommendations to be considered for implementation:

Related Documents

For more information, see the following documentation:

- For installation and configuration information, see the Oracle Banking Platform Localization Installation Guide - Silent Installation guide.
- For a comprehensive overview of security for Oracle Banking, see the Oracle Banking Platform Security Guide.
- For the complete list of Oracle Banking licensed products and the Third Party licenses included with the license, see the Oracle Banking Platform Licensing Guide.
- For information related to customization and extension of Oracle Banking, see the Oracle Banking Platform Extensibility Guide for Host, SOA, and UI.
- For information on the functionality and features of the Oracle Banking product licenses, see the respective Oracle Banking Platform Functional Overview documents.
- For recommendations of secure usage of extensible components, see the Oracle Banking Platform Secure Development Guide.

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

1 User Administration with OIM

This chapter describes all user management related activities to be performed by an administrator for the application.

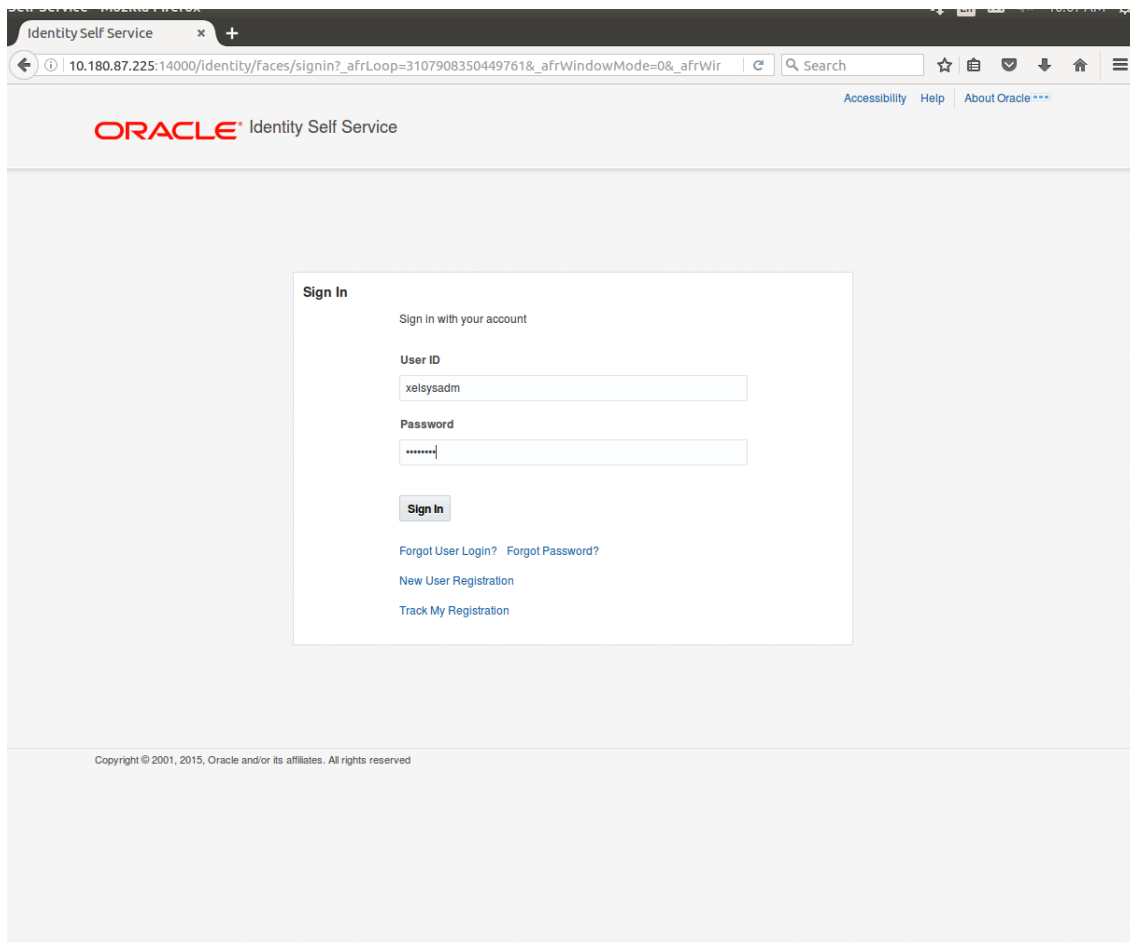
1.1 Creating Users in Oracle Identity Manager (OIM)

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

To create users in OIM:

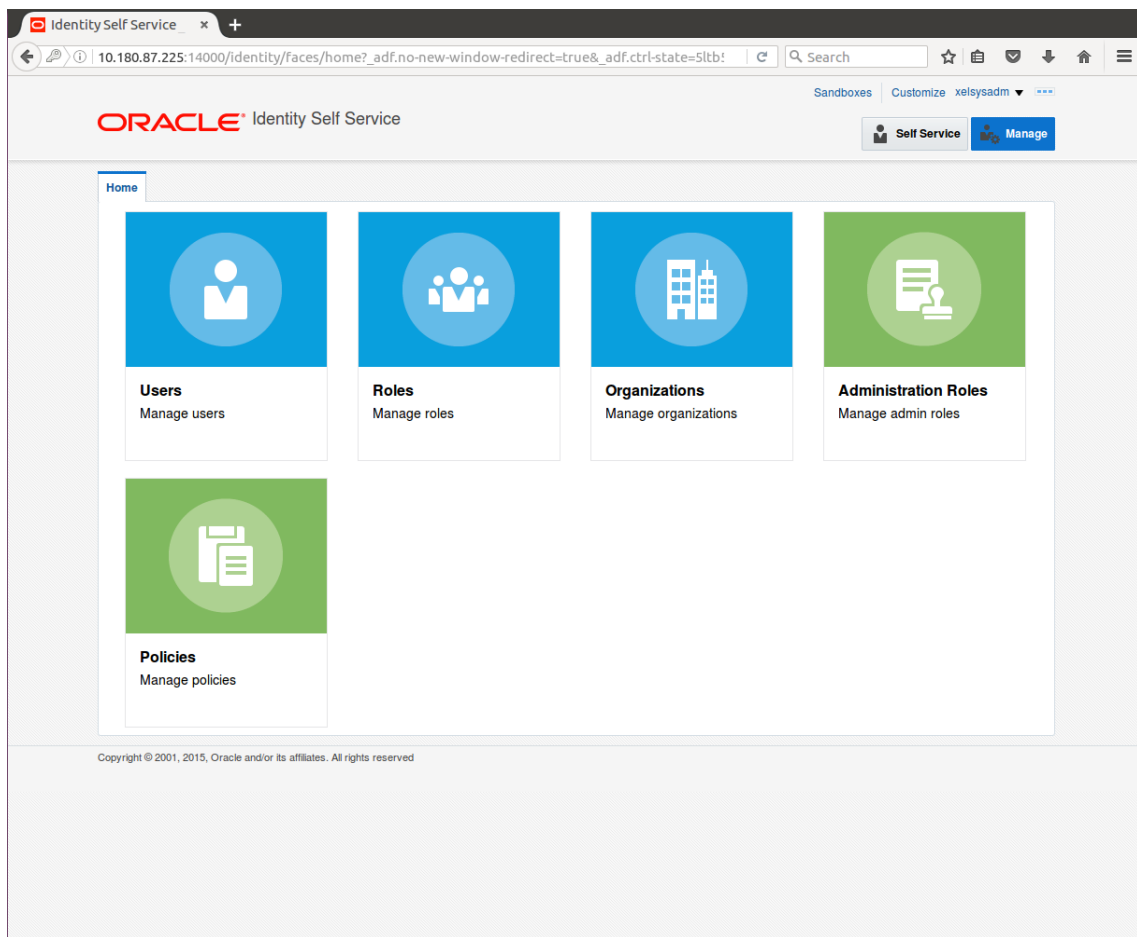
1. Log in to OIM with the User ID as **xelsysadm** and the relevant <Password>.

Figure 1–1 Creating Users in OIM - Log in



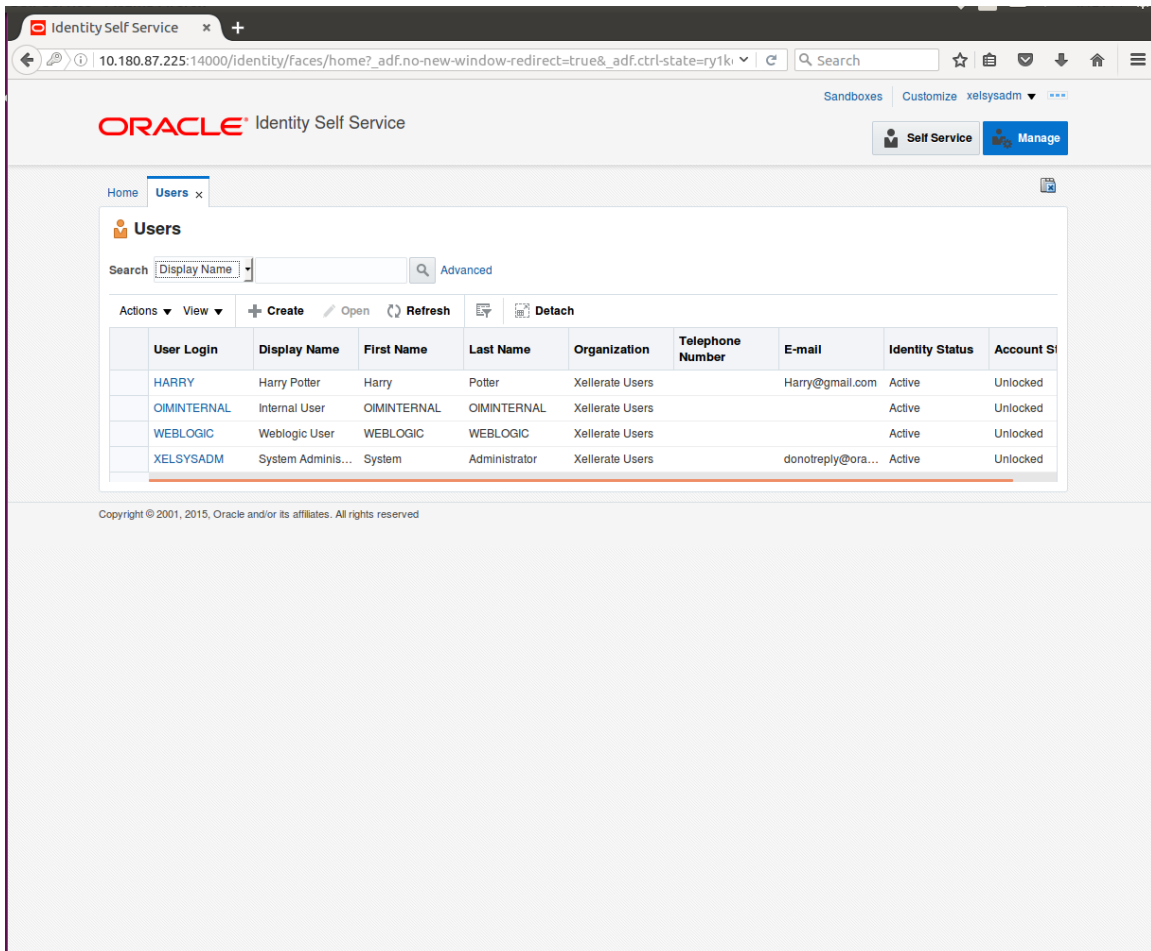
2. Click **Users** under the Manage section.

Figure 1–2 Creating Users in OIM - Manage Section



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 Creating Users in OIM - Click Create



The screenshot displays the Oracle Identity Self Service interface. At the top, the browser address bar shows the URL: 10.180.87.225:14000/identity/faces/home?_adf.no-new-window-redirect=true&_adf.ctrl-state=ry1k. The page title is "Identity Self Service" and the Oracle logo is visible. The main content area is titled "Users" and includes a search bar with "Display Name" selected and a "Create" button. Below the search bar is a table of users with the following data:

User Login	Display Name	First Name	Last Name	Organization	Telephone Number	E-mail	Identity Status	Account Status
HARRY	Harry Potter	Harry	Potter	Xellerate Users		Harry@gmail.com	Active	Unlocked
OIMINTERNAL	Internal User	OIMINTERNAL	OIMINTERNAL	Xellerate Users			Active	Unlocked
WEBLOGIC	Weblogic User	WEBLOGIC	WEBLOGIC	Xellerate Users			Active	Unlocked
XELSYSADM	System Administrator	System	Administrator	Xellerate Users		donotreply@ora...	Active	Unlocked

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

Figure 1–4 Creating Users in OIM - Enter User Details

The screenshot shows the Oracle Identity Self Service interface for creating a user. The browser address bar shows the URL: 10.180.87.225:14000/identity/faces/home?_adf.no-new-window-redirect=true&_adf.ctrl-state=ry1k. The page title is "ORACLE Identity Self Service". The user is logged in as "xelsysadm". The "Create User" form is displayed with the following sections and fields:

- Request Information:**
 - Effective Date:
 - Justification:
- Basic Information:**
 - First Name:
 - Middle Name:
 - Last Name:
 - E-mail:
 - Manager:
 - Organization:
 - User Type:
 - Display Name:
- Account Settings:**
 - User Login:
 - Password:
 - Confirm Password:
- Account Effective Dates:**
 - Start Date:
 - End Date:
- Provisioning Dates:** (Section header only)

At the top right of the form, there are three buttons: "Submit" (green), "Save As..." (grey), and "Cancel" (grey).

Figure 1–5 Enter User Details (Continued)

The screenshot shows a web browser window with the URL `10.180.87.225:14000/identity/faces/home?_adf.no-new-window-redirect=true&_adf.ctrl-state=ry1k`. The page title is "Identity Self Service". The form contains the following sections and fields:

- Confirm Password:** A single text input field.
- Account Effective Dates:** Two date pickers for "Start Date" and "End Date".
- Provisioning Dates:** Two date pickers for "Provisioning Date" and "Deprovisioning Date".
- Contact Information:** A grid of fields including Telephone Number, Home Phone, Fax, Mobile, Pager, Home Postal Address, Postal Address, Postal Code, PO Box, State, Street, and Country.
- Preferences:** A dropdown for "Locale" and a text field for "Timezone".
- Other Attributes:** Fields for Common Name, Department Number, Employee Number, Generation Qualifier, Hire Date, Locality Name, Initials, and Title.

At the bottom of the page, there is a copyright notice: "Copyright © 2001, 2015, Oracle and/or its affiliates. All rights reserved."

6. Click **Submit**.

On completion of this procedure the user gets created in OIM, and gets synced in OID.

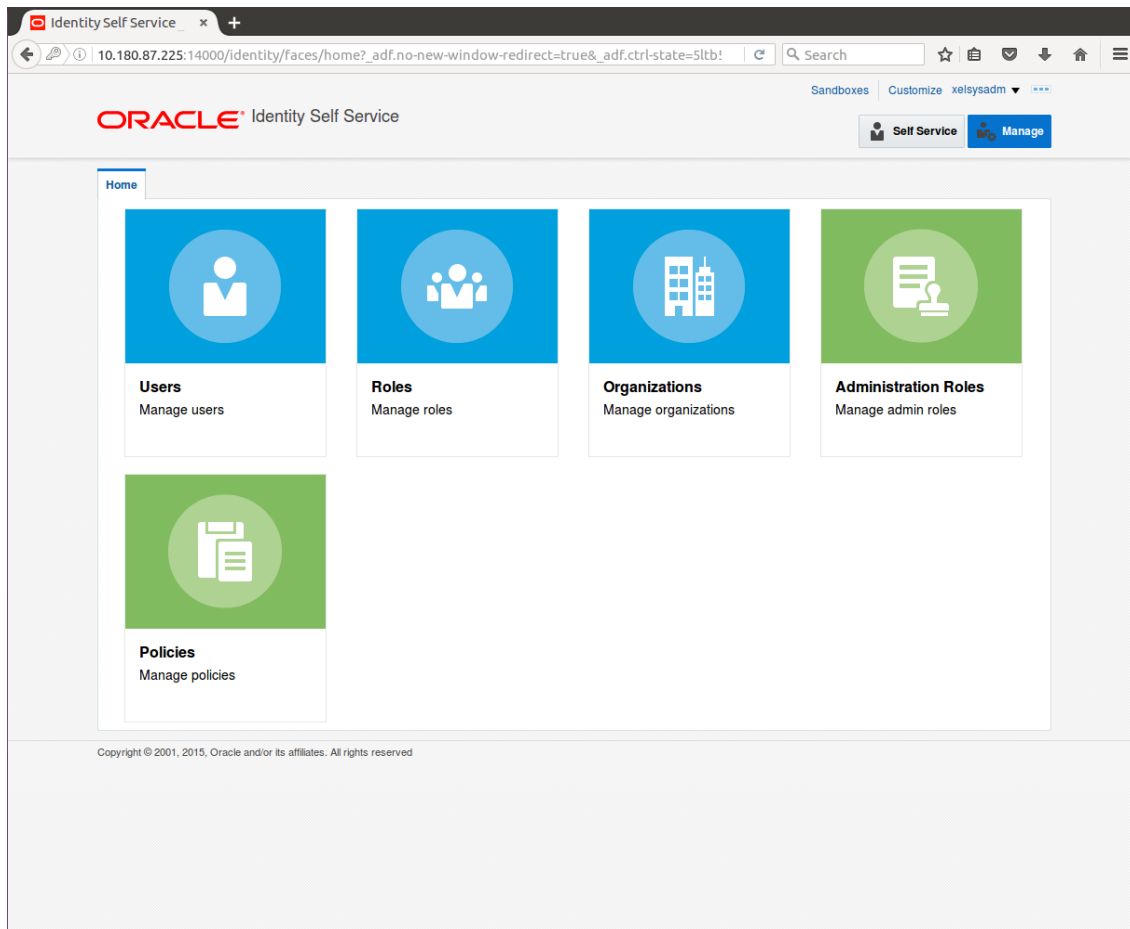
1.2 Creating Roles in Oracle Identity Manager (OIM)

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

To create roles in OIM:

1. Click **Roles** under the Manage section.

Figure 1–6 Creating Roles in OIM - Manage Section



2. In the **Search Roles** page, search for existing roles. The Search Results appear.
3. Click **Create** in the Search Results section to create a new Role.

Figure 1–7 Creating Roles in OIM - Click Create

The screenshot shows the Oracle Identity Self Service interface. The browser address bar indicates the URL: 10.180.87.225:14000/identity/faces/home?_adf.no-new-window-redirect=true&_adf.ctrl-state=5ltb: . The page title is "ORACLE Identity Self Service". The user is logged in as "xelsysadm". The "Roles" page is active, showing a search bar and a table of roles. The table has two columns: "Name" and "Role Description".

Name	Role Description
ALL USERS	Default role for all users
Administrators	Administrators role for SOA
BIReportAd...	Administrators role for BI Publisher Reports
OPERATORS	Operator role
SELF OPER...	Operator role for self registration
SYSTEM AD...	System Administrator role for OIM

4. Fill the role details.

Figure 1–8 Creating Roles in OIM - Enter Role Details

The screenshot shows the Oracle Identity Self Service interface for creating a role. The browser address bar shows the URL: 10.180.87.225:14000/identity/faces/home?_adf.no-new-window-redirect=true&_adf.ctrl-state=51tb! . The page title is "ORACLE Identity Self Service". The user is logged in as "xelsysadm". The "Create Role" wizard is active, showing a progress bar with steps: Back, Attributes (current), Hierarchy, Access Policy, Members, Organizations, and Summary. The "Attributes" step is highlighted. The "General Role Information" section contains the following fields:

- Name: TestFullAccess
- Display Name: TestFullAccess
- Role E-mail: (empty)
- Role Description: TestFullAccess
- Owned By: System Administrator

The "Catalog Attributes" section contains the following fields:

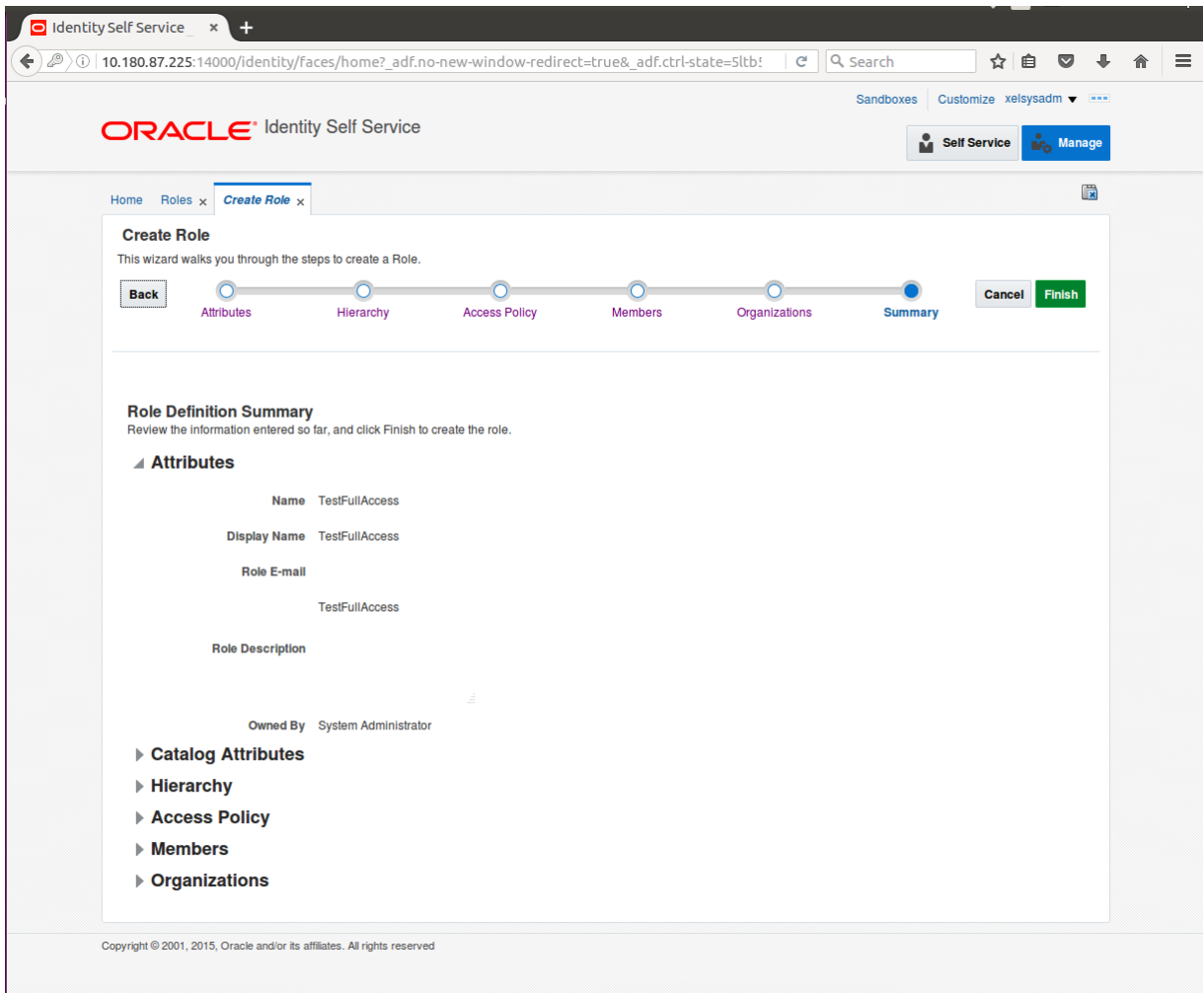
- Category: Role
- Audit Objective: (empty)
- Risk Level: (dropdown menu)
- User Defined Tags: (empty)
- Approver User: (empty)

5. Click **Finish**. The role is created successfully.

This role creates a group in OID.

While running the PIT (Policy Import tool), the Enterprise role (OIM role or OID group in this scenario) is mapped to the Application Role in OES.

Figure 1–9 Creating Roles in OIM - Role Created Successfully



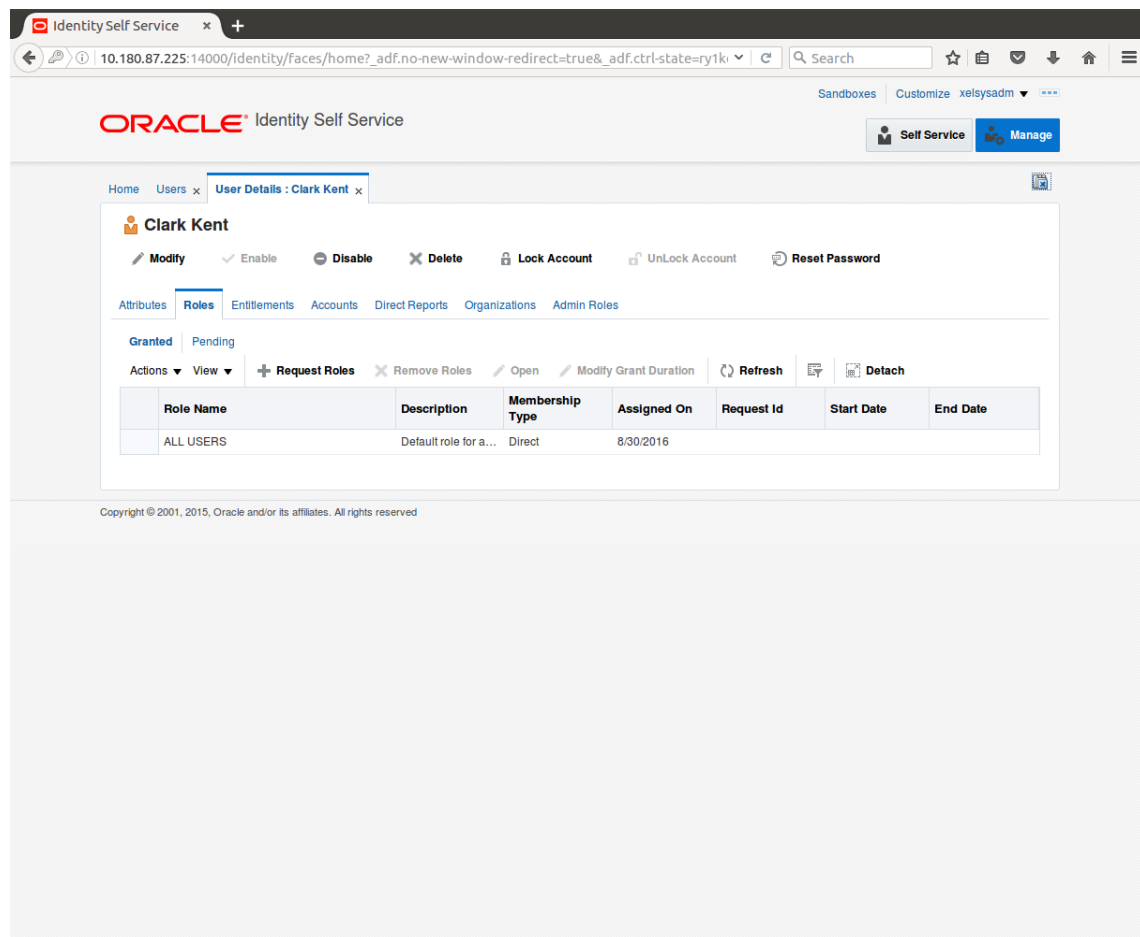
1.3 Assigning Roles to Users in OIM

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

To assign a role to a user:

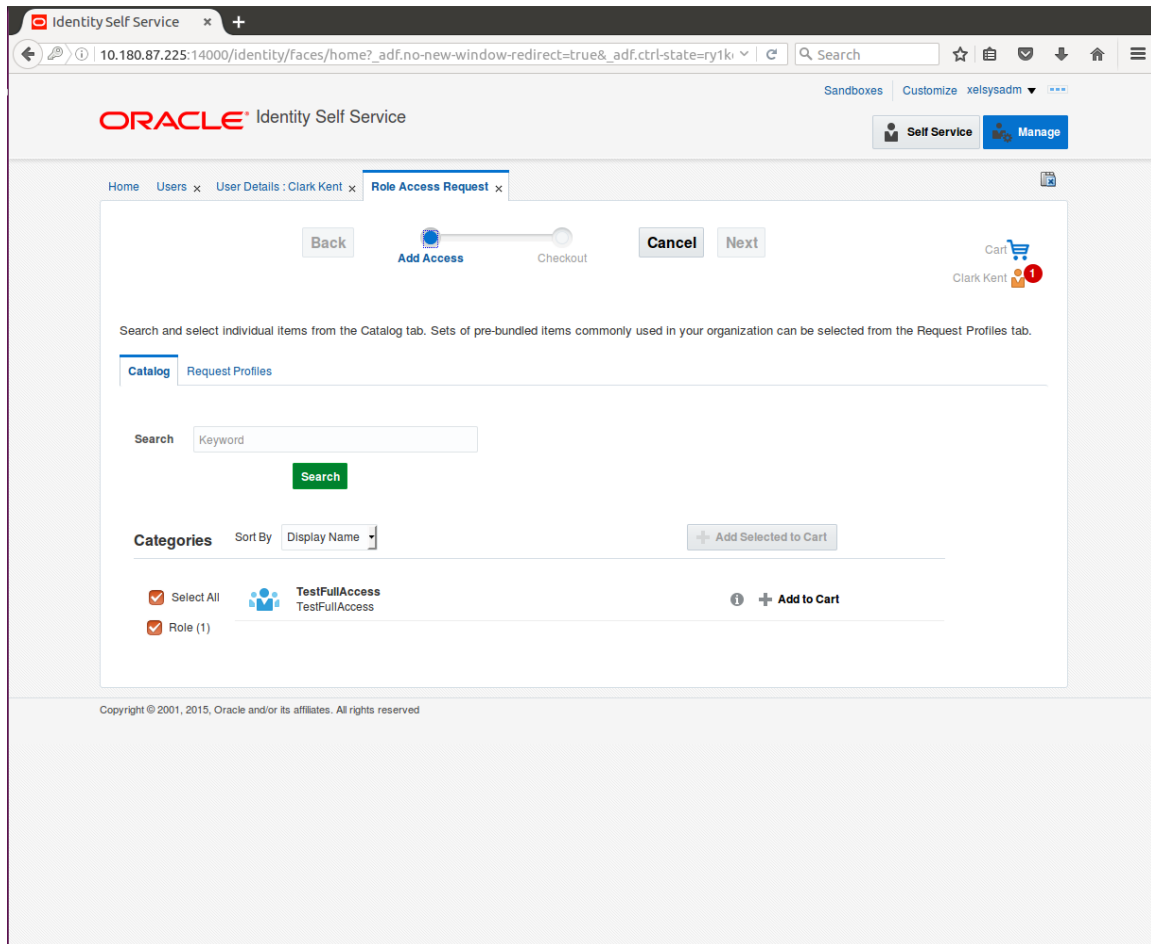
1. Log in to OIM.
2. Navigate to the **Roles Tab** under the User.
3. Click **Request Roles**.

Figure 1–10 Assigning Roles in OIM - Requesting Roles



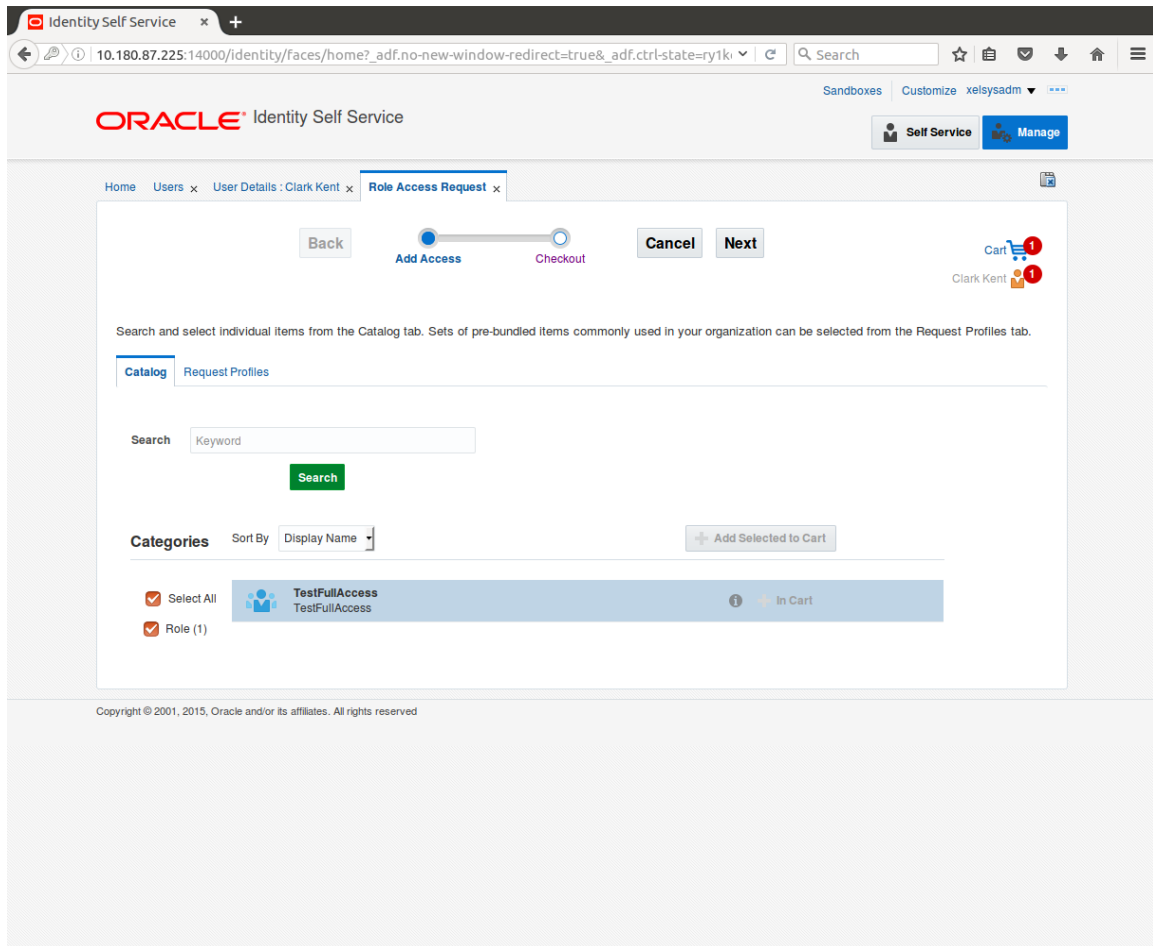
4. In the **Catalog** page, select the required role and click **Add to Cart**. The item gets added to the cart.

Figure 1–11 Assigning Roles in OIM - Adding to Cart



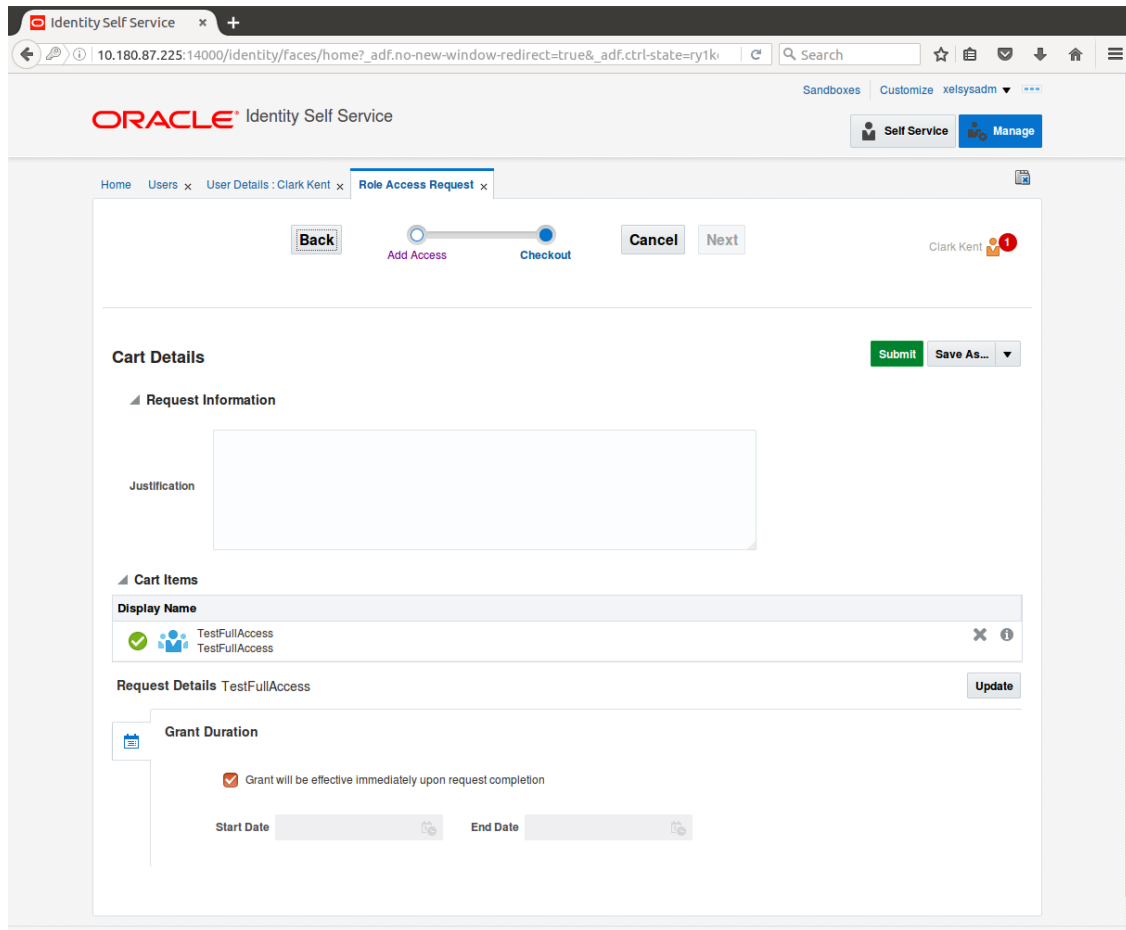
5. Click **Checkout**.

Figure 1–12 Assigning Roles in OIM - Checkout Cart



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

Figure 1–13 Assigning Roles in OIM - Submit Cart



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

1.4 Locking Users in OIM

This section explains how to lock the user in OIM.

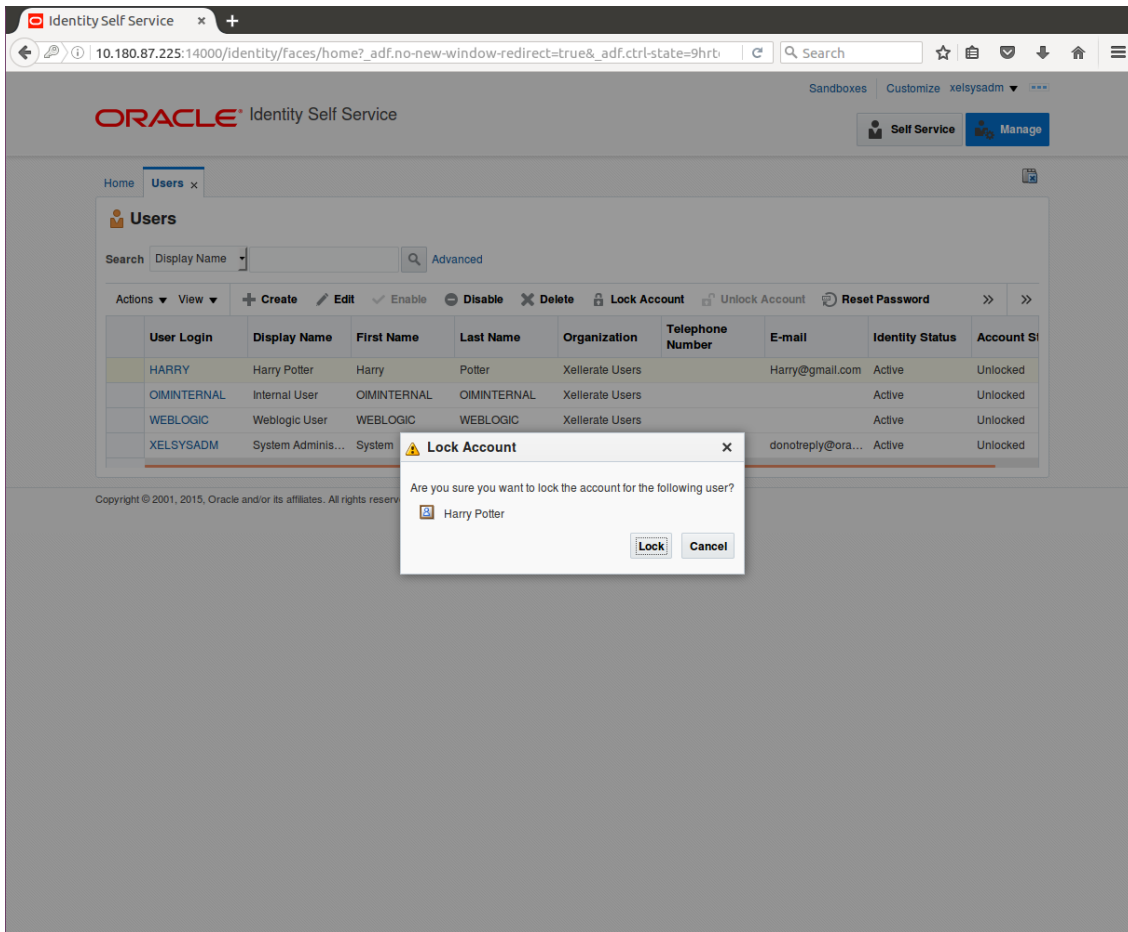
To lock a user:

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 1–14 Locking Users in OIM



The user is locked successfully.

Figure 1–15 User Locked Successfully

The screenshot shows the Oracle Identity Self Service interface. At the top, a green message box states "Account locked successfully". Below this, the "Users" section is visible, featuring a search bar and a table of users. The table includes columns for User Login, Display Name, First Name, Last Name, Organization, Telephone Number, E-mail, Identity Status, and Account Status. The user "HARRY" is listed with a status of "Active" and "Locked".

User Login	Display Name	First Name	Last Name	Organization	Telephone Number	E-mail	Identity Status	Account Status
HARRY	Harry Potter	Harry	Potter	Xellerate Users		Harry@gmail.com	Active	Locked
OIMINTERNAL	Internal User	OIMINTERNAL	OIMINTERNAL	Xellerate Users			Active	Unlocked
WEBLOGIC	Weblogic User	WEBLOGIC	WEBLOGIC	Xellerate Users			Active	Unlocked
XELSYSADM	System Adminis...	System	Administrator	Xellerate Users		donotreply@ora...	Active	Unlocked

1.5 Unlocking Users in OIM

This section explains how to unlock the user in OIM.

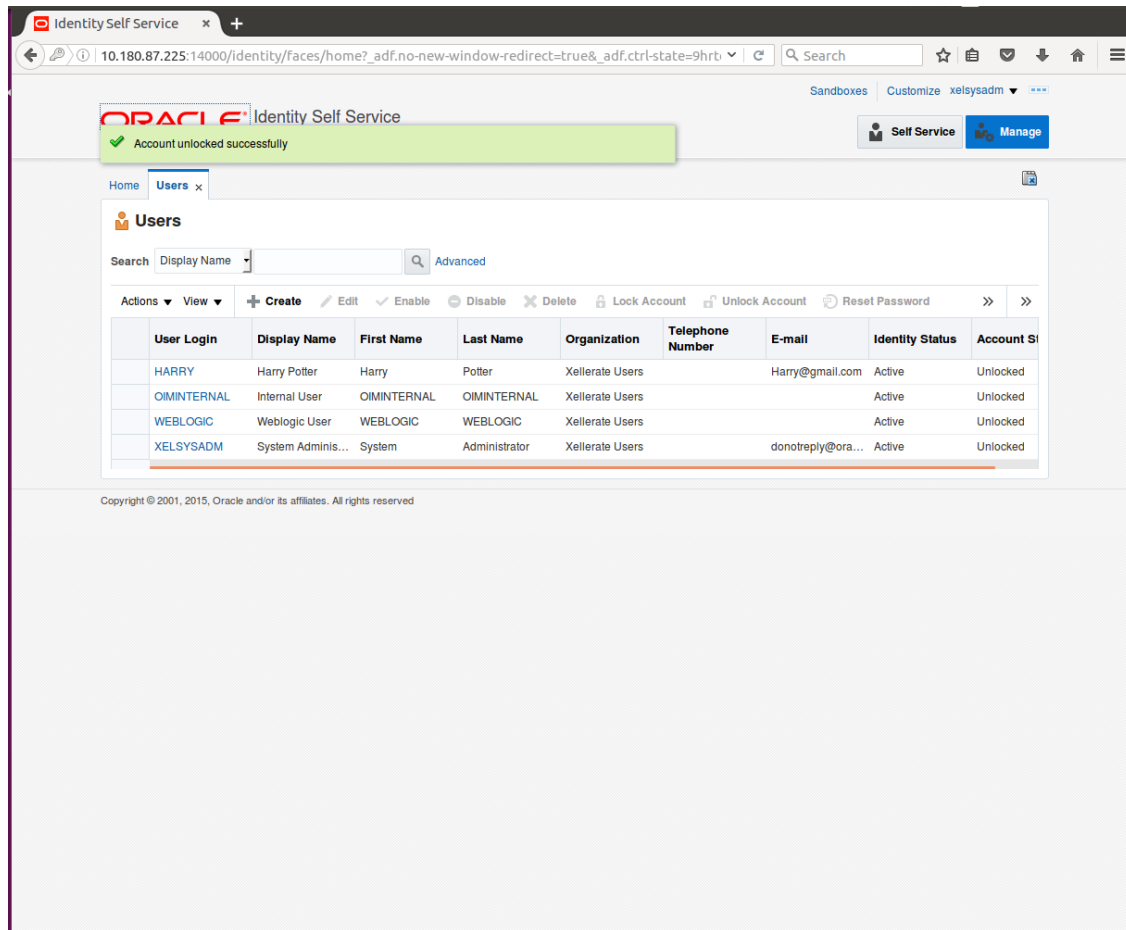
To unlock a user:

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 1–16 Unlocking Users in OIM



The user is unlocked successfully.

1.6 Resetting User Password in OIM

This section explains how to reset user password in OIM.

1. Log in to OIM.
2. Click **Reset Password** to reset a user password.

Figure 1–17 Resetting User Password in OIM

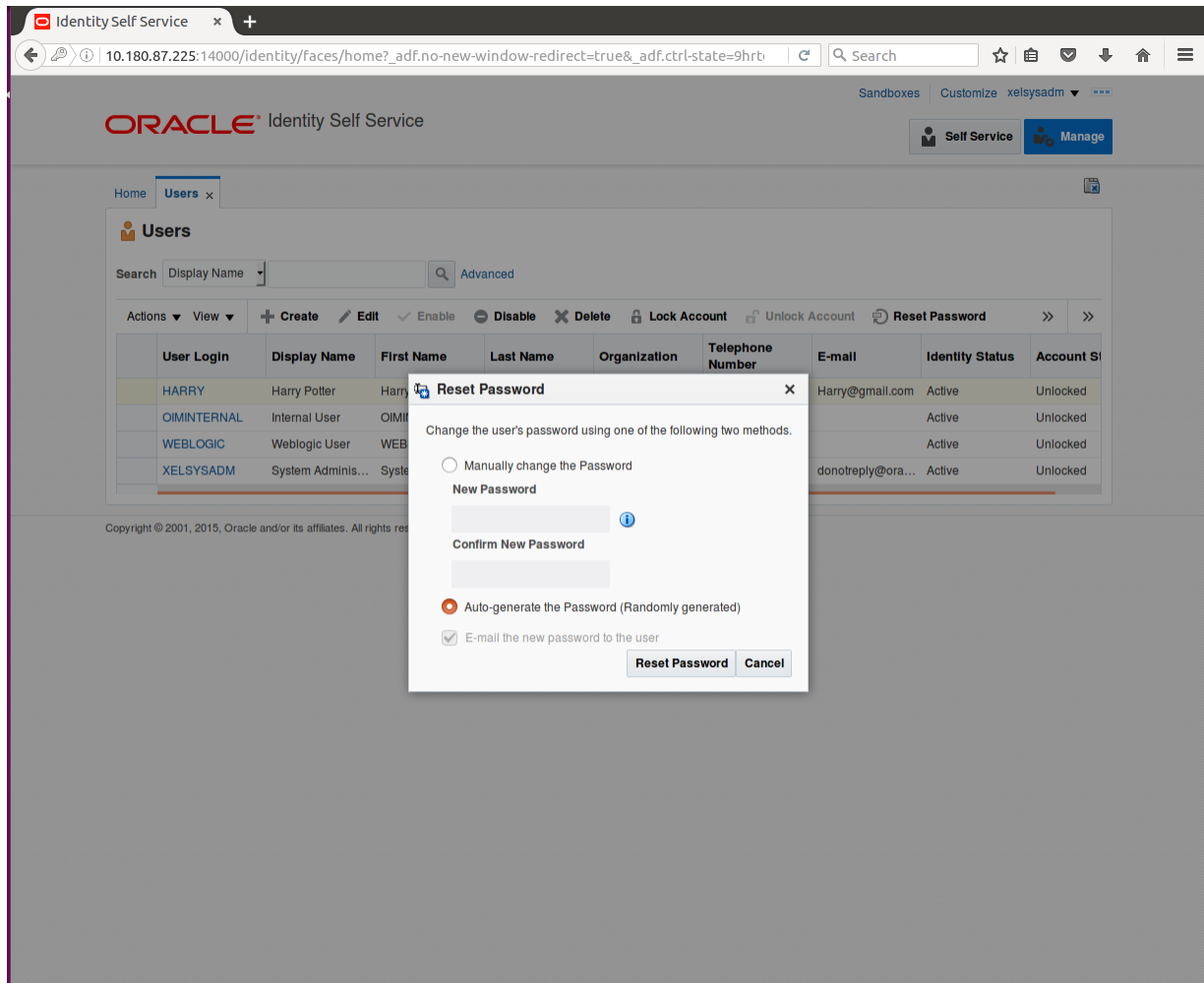
The screenshot shows the Oracle Identity Self Service interface. The main content area is titled 'Users' and contains a search bar and a table of users. The table has the following data:

User Login	Display Name	First Name	Last Name	Organization	Telephone Number	E-mail	Identity Status	Account Status
HARRY	Harry Potter	Harry	Potter	Xellerate Users		Harry@gmail.com	Active	Unlocked
OIMINTERNAL	Internal User	OIMINTERNAL	OIMINTERNAL	Xellerate Users			Active	Unlocked
WEBLOGIC	Weblogic User	WEBLOGIC	WEBLOGIC	Xellerate Users			Active	Unlocked
XELSYSADM	System Adminis...	System	Administrator	Xellerate Users		donotreply@ora...	Active	Unlocked

Below the table, there is a copyright notice: Copyright © 2001, 2015, Oracle and/or its affiliates. All rights reserved.

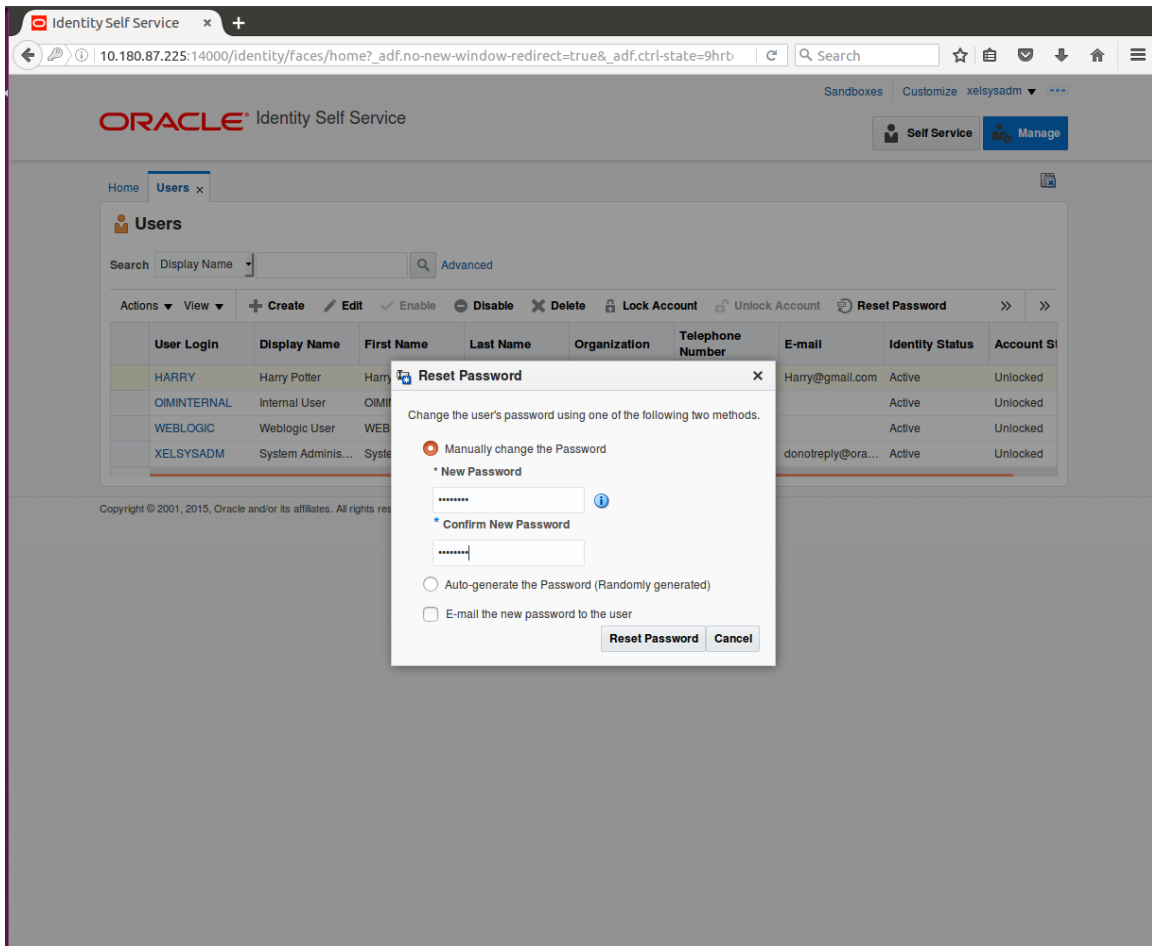
The **Reset Password** dialog box appears.

You can select either **Manually change the Password** option to change the password manually or select the **Auto-generate the password (Randomly generated)** option to enable auto generation of the password.

Figure 1–18 Resetting User Password in OIM - Manually or Auto-generate

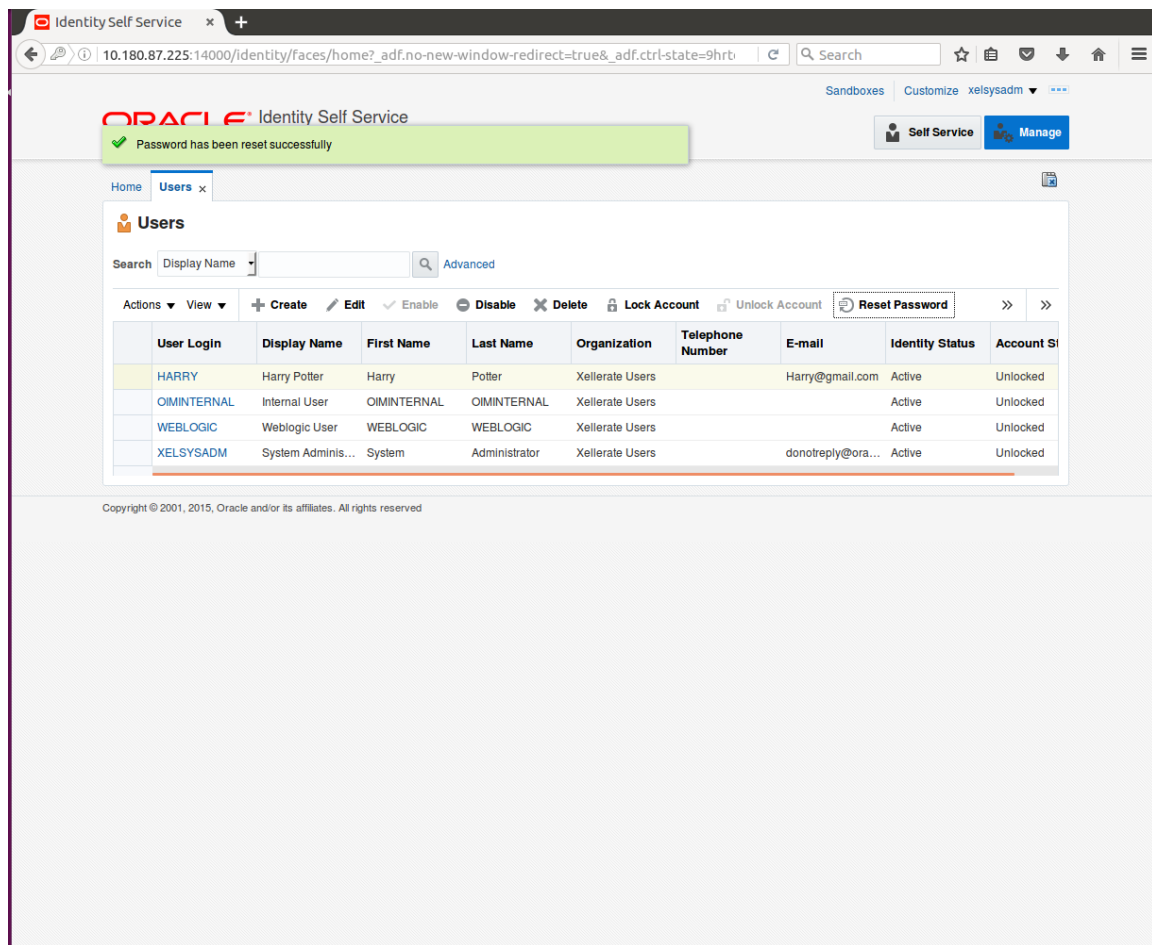
3. If you select the **Manually change the Password** option, enter the new password in the **New Password** and the **Confirm New Password** fields.

Figure 1–19 Resetting User Password in OIM - New Password



The user password is reset successfully.

Figure 1–20 Password Reset Successfully



1.7 User Management Using the 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 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 1–21 Adding a User

The screenshot displays the Oracle Banking Platform Admin Application interface. The browser address bar shows the URL: `https://10.180.84.177:8002/com.ofss.fc.ui.view.admin/faces/admin.jspx?_afrcLoop=31010305172428t`. The page title is "Oracle Banking Platform" and the posting date is "15-Jan-2016". The main content area is titled "User Management" and includes a search filter, a table of user details, and a user details form.

User Management

Search Filter

Username

User Details

Username	Target Unit	Branch	Delete
----------	-------------	--------	--------

User Details Form

Username	Preferred Language
First Name	Accreditation
Last Name	Brand
Email	2FA Status
Password	Forum Nick Name
Confirm password	Party Id
Home Branch	Last Logged In Date Time
Manager	2FA Inactive Begin Date
Target Unit	2FA Inactive End Date

4. Enter the mandatory fields required for creating a user.

Figure 1–22 Enter Mandatory Details

The screenshot shows the Oracle Banking Platform Admin Application interface. The page title is "User Management". At the top, there is a search filter with a "Username" input field. Below the search filter is a table with columns "Username", "Target Unit", "Branch", and "Delete". The table is currently empty. Below the table is the "User Details Form" section, which contains various input fields for user information. The form includes fields for Username (Harry), First Name (Harry), Last Name (Potter), Email (Harry@gmail.com), Password (masked with dots), Confirm password (masked with dots), Preferred Language, Accreditation, Brand, 2FA Status, Forum Nick Name, Party Id, Home Branch (1010), Last Logged In Date Time, Manager, 2FA Inactive Begin Date, Target Unit (3LBL_BU_PB), and 2FA Inactive End Date. The form also has buttons for "Edit", "Apply changes", and "Assign Roles".

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

Figure 1–23 Applying Changes

The screenshot shows the Oracle Banking Platform Admin Application interface. The browser address bar displays the URL: `https://10.180.84.177:8002/com.ofss.fc.ui.view.admin/faces/admin.jspx?_afrcLoop=32073103220256`. The page title is "Oracle Banking Platform" and the posting date is "15-Jan-2016".

The main content area is titled "User Management" and includes a "Search Filter" section with a "Username" input field and a search button. Below this is a "User Details" table with the following data:

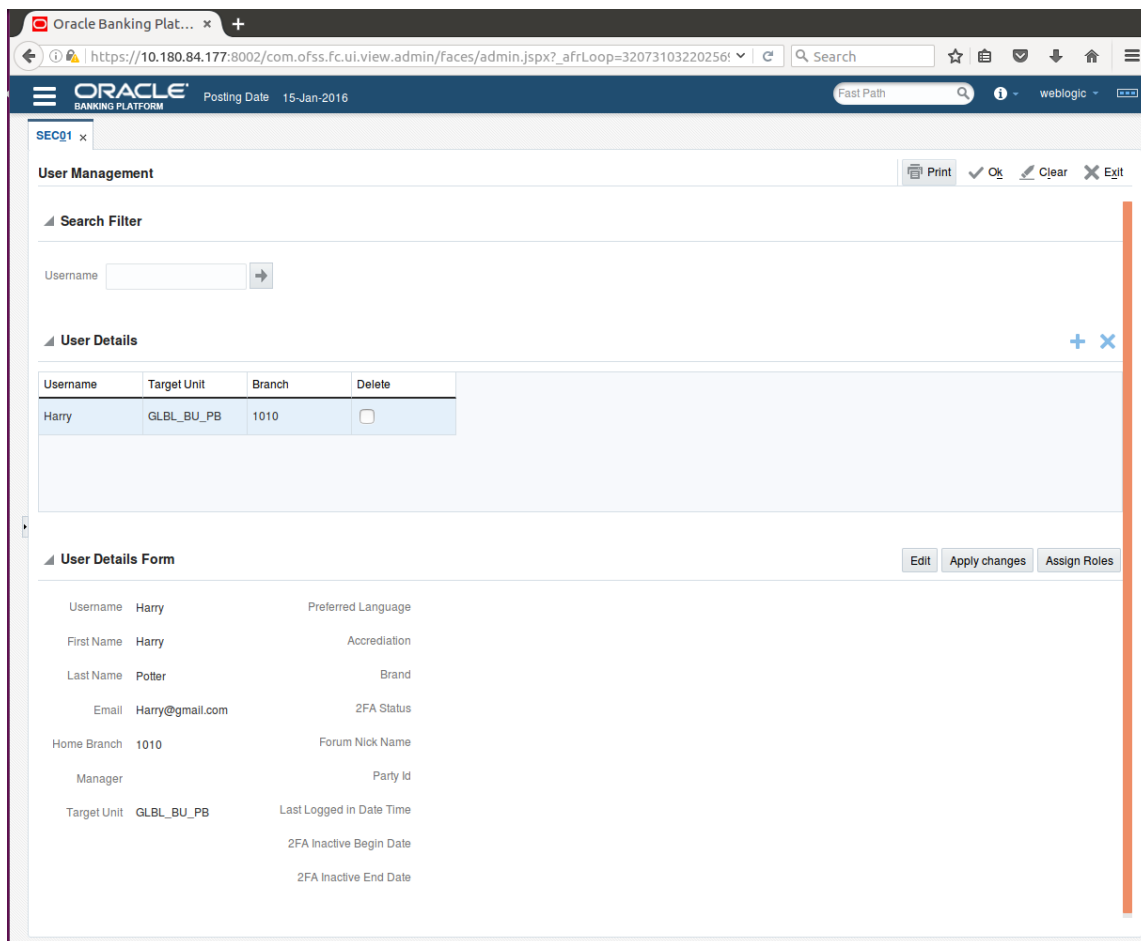
Username	Target Unit	Branch	Delete
Harry	GLBL_BU_PB	1010	<input type="checkbox"/>

Below the table is a "User Details Form" section with buttons for "Edit", "Apply changes", and "Assign Roles". The form displays the following user details:

Username	Harry	Preferred Language
First Name	Harry	Accreditation
Last Name	Potter	Brand
Email	Harry@gmail.com	2FA Status
Home Branch	1010	Forum Nick Name
Manager		Party Id
Target Unit	GLBL_BU_PB	Last Logged In Date Time
		2FA Inactive Begin Date
		2FA Inactive End Date

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

Figure 1–24 Adding User to a Group



The available and assigned roles appear.

Figure 1–25 Available and Assigned Roles

The screenshot displays the Oracle Banking Platform User Management interface. The page title is "User Management" and it includes a search filter for "Username" and a "User Details" section. The "User Details" section shows a table with one user, "Harry", associated with "GLBL_BU_PB" and "1010". Below this, the "Groups" section shows two tables: "All Roles" and "Assigned Roles". The "All Roles" table contains one role, "TestFullAccess", and the "Assigned Roles" table is empty, displaying "No data to display.".

User Management

Search Filter

Username

User Details

Username	Target Unit	Branch	Delete
Harry	GLBL_BU_PB	1010	<input type="checkbox"/>

Groups

All Roles

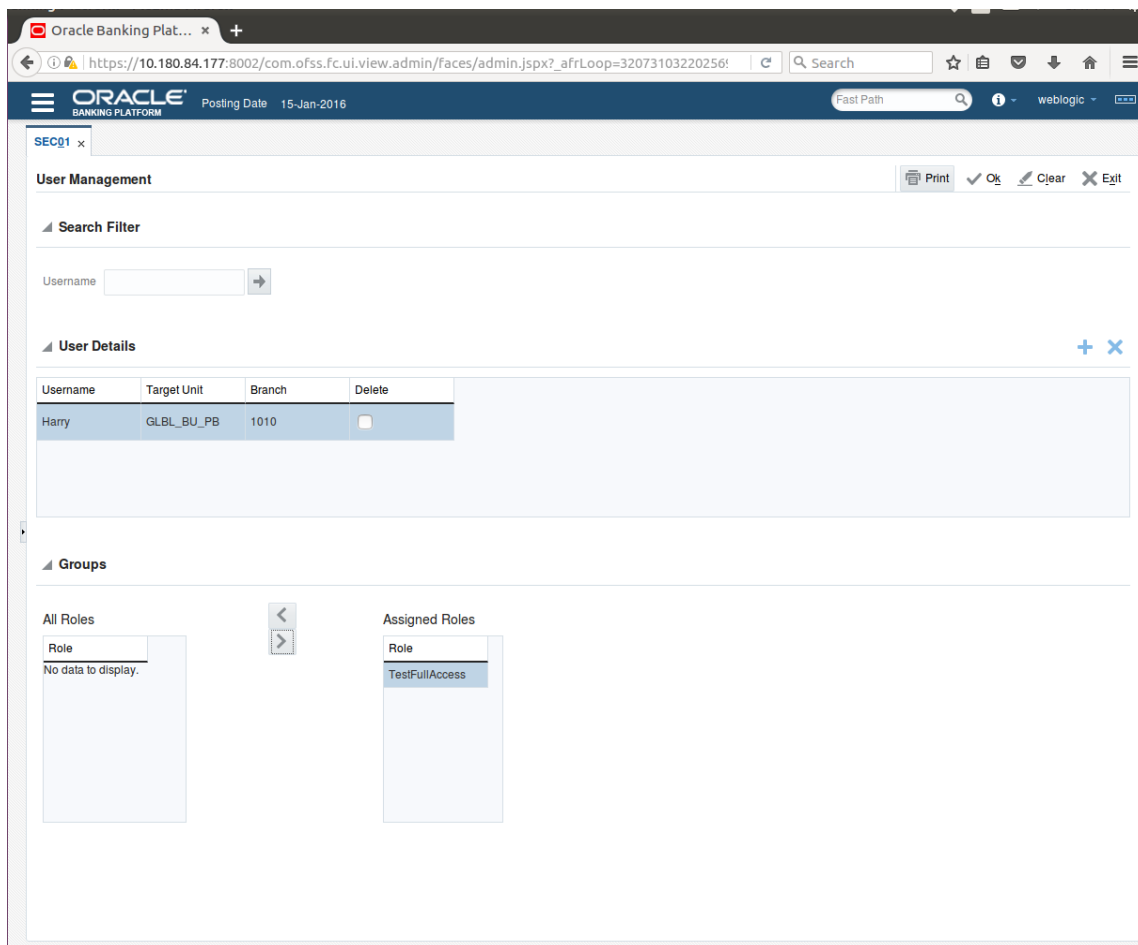
Role
TestFullAccess

Assigned Roles

Role
No data to display.

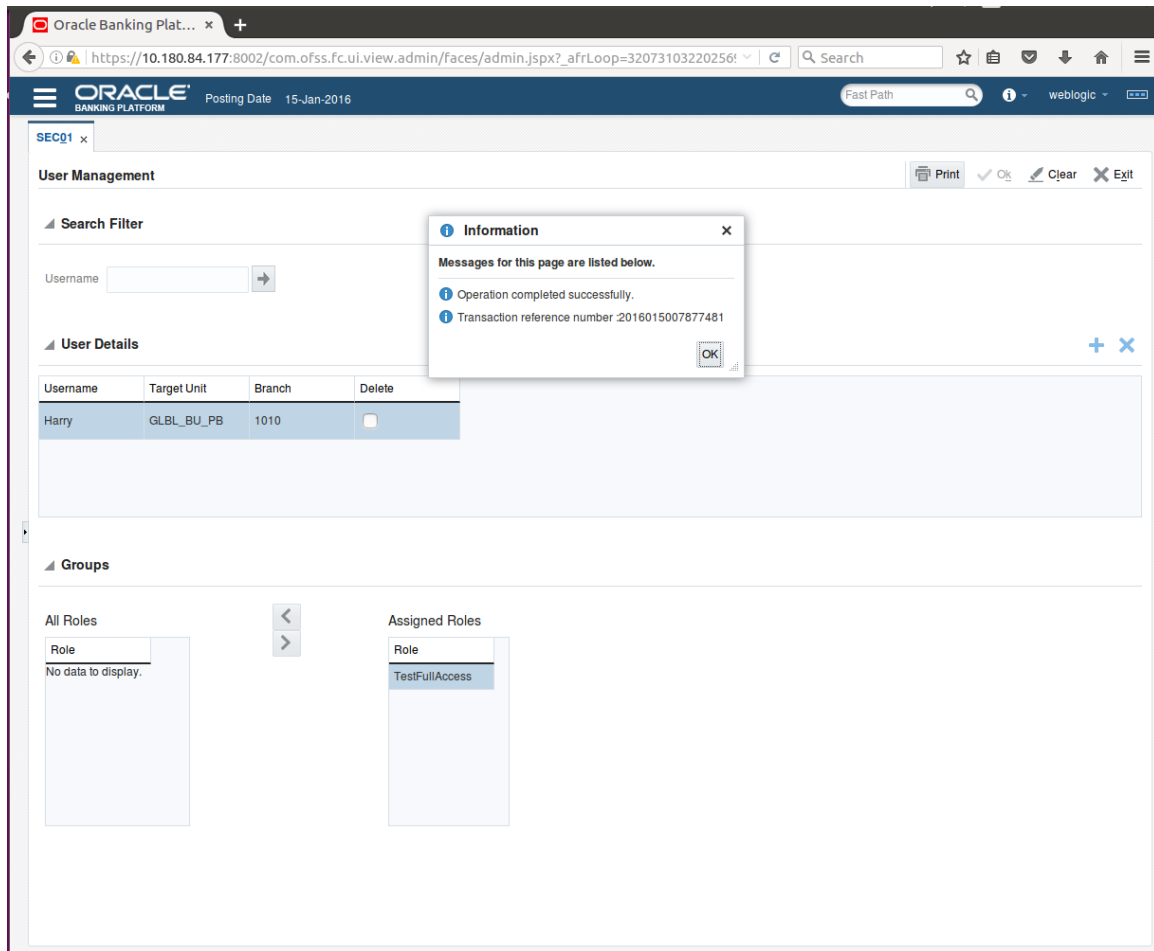
7. Select the group to add user and move it to the **Assigned Roles** table.

Figure 1–26 Adding User to Assigned Roles Table



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

Figure 1–27 Save Changes



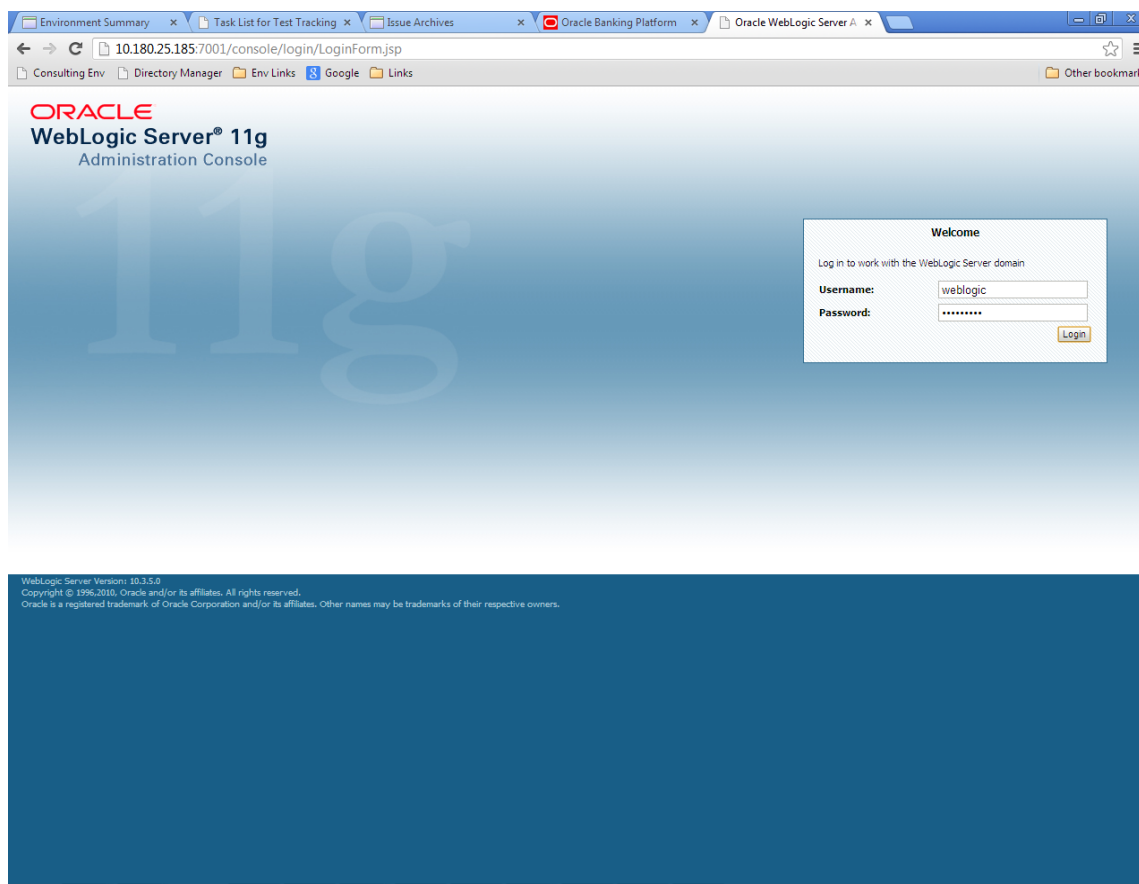
1.8 Unlocking Users in Oracle WebLogic Server (OWS) Administration Console

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:

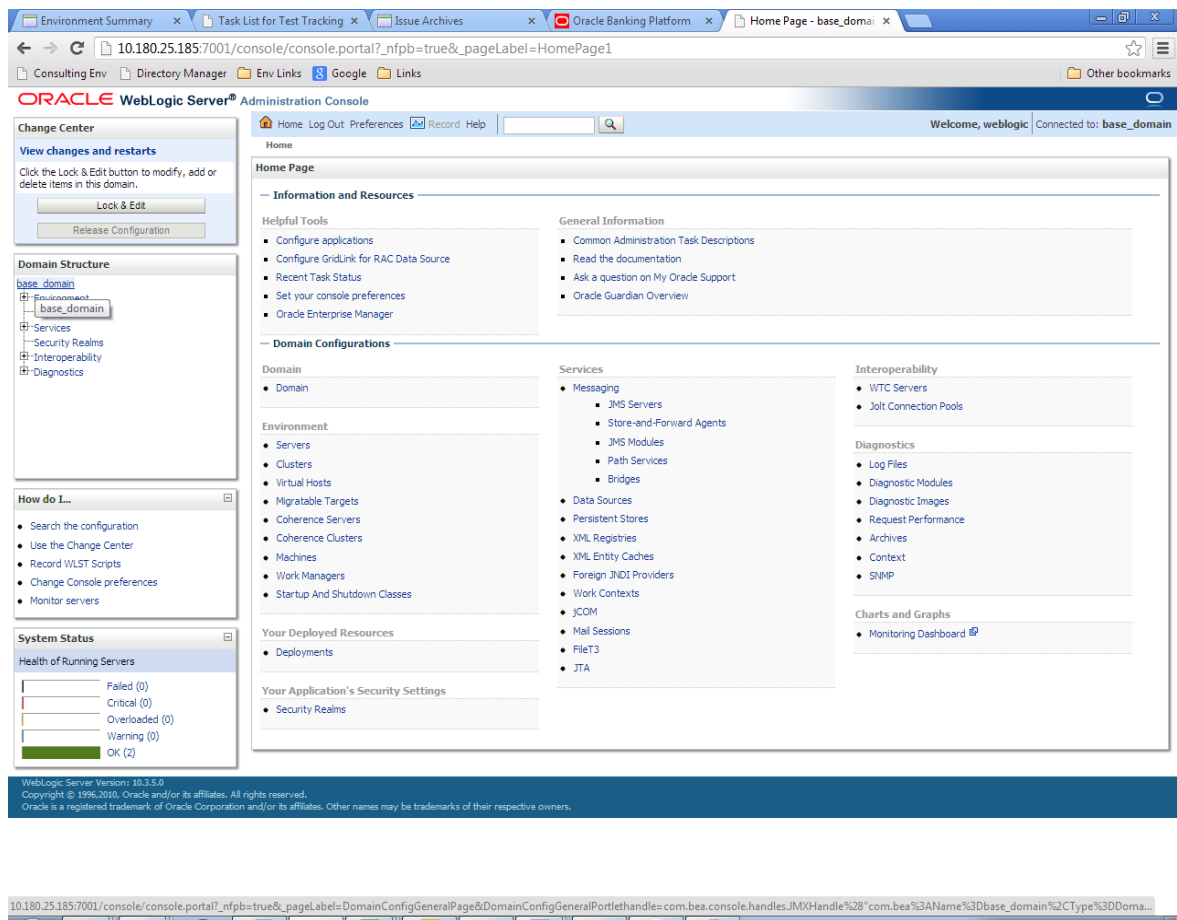
1. Log in to OWS. The **Home Page** of OWS Administration Console appears.

Figure 1–28 OWS Log in



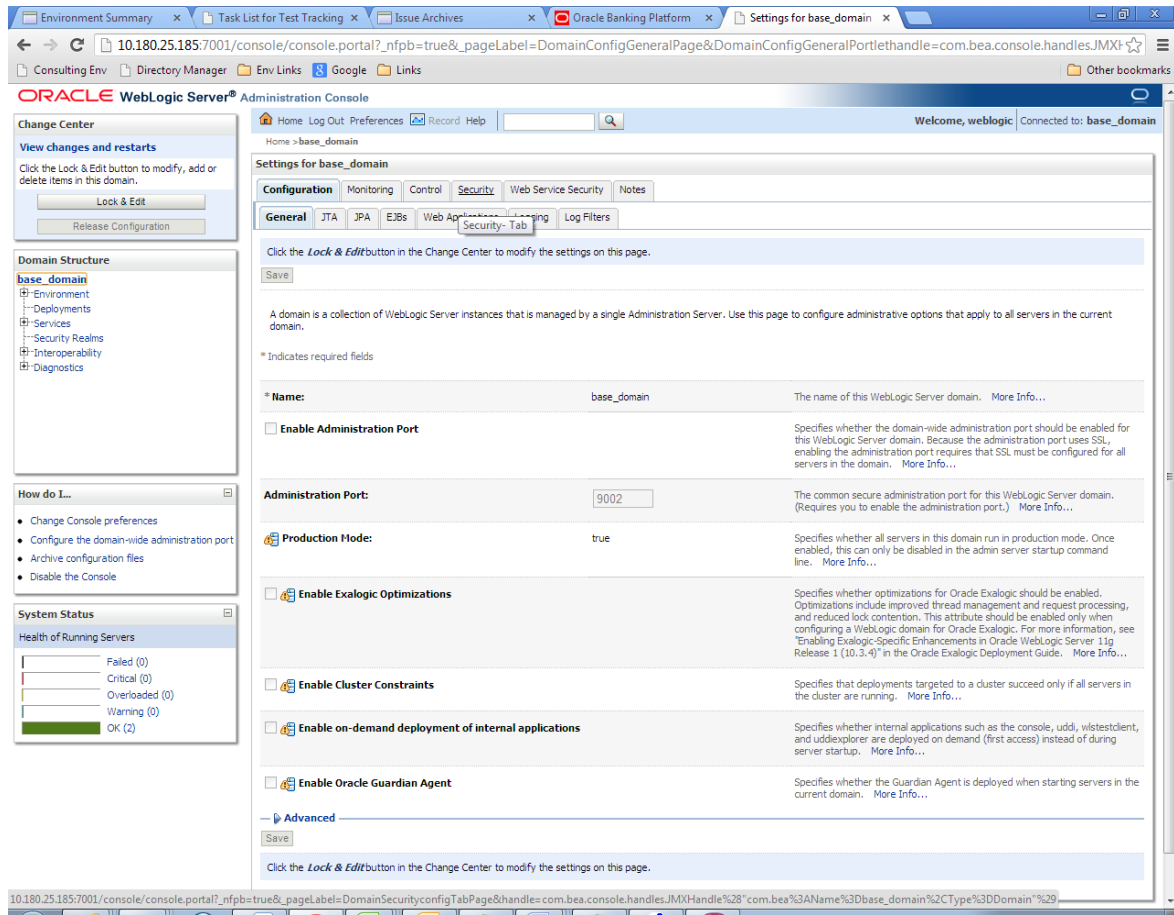
2. In the **Domain Structure** section, click the **base_domain** link.

Figure 1–29 base_domain



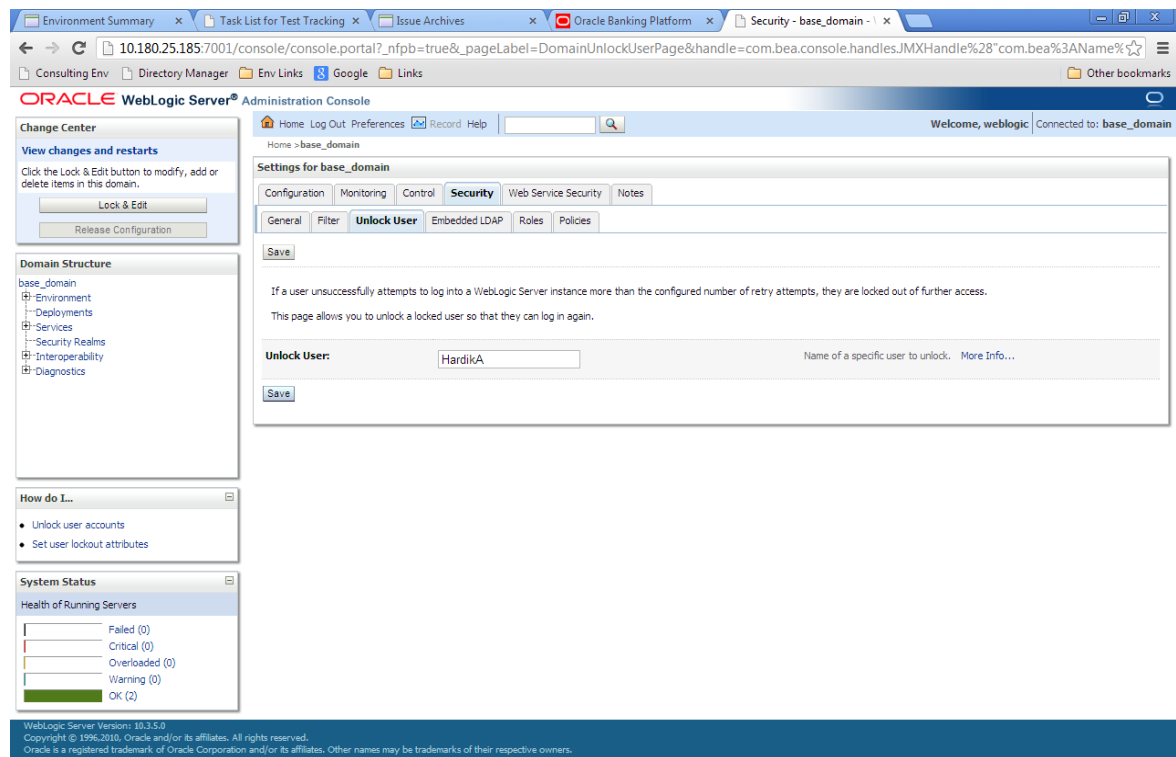
3. In the **Settings for base_domain** page that appears, click the **Security** tab.

Figure 1–30 Security tab



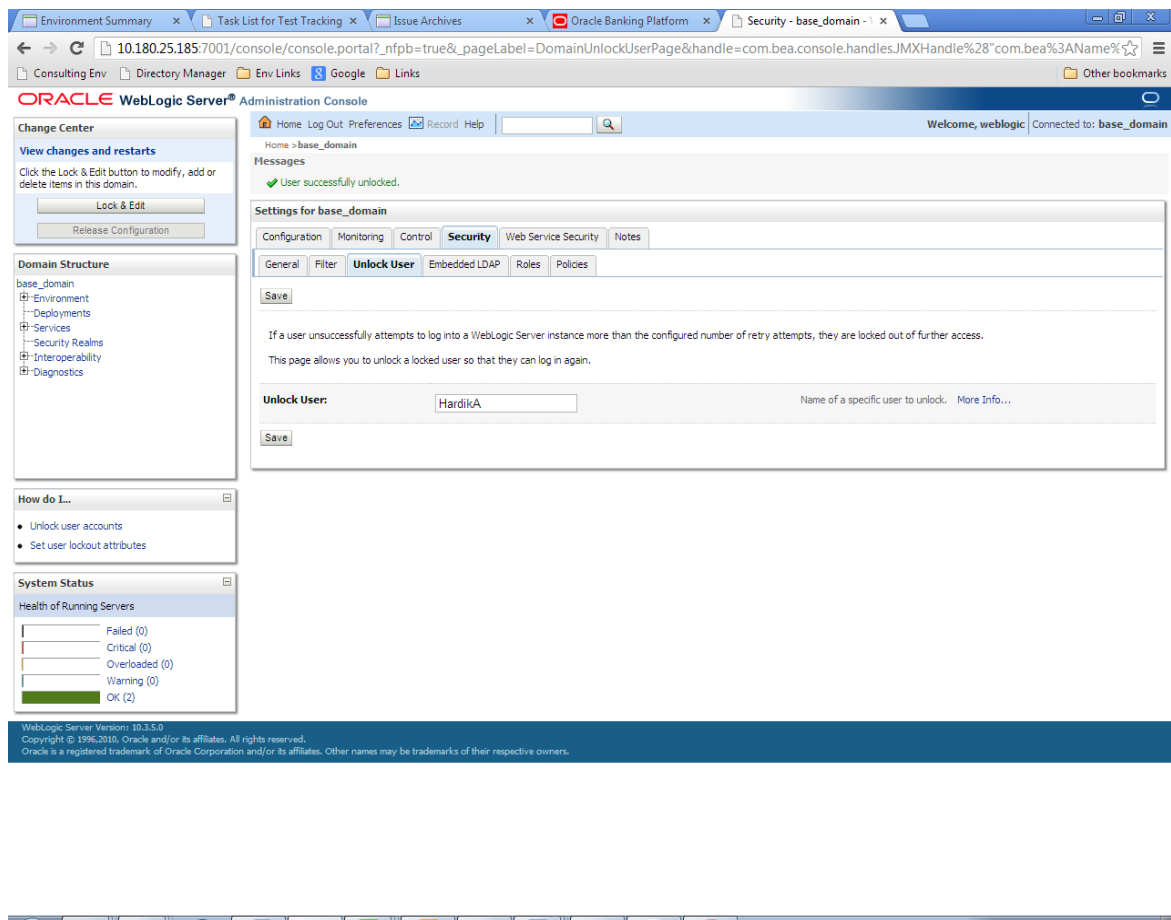
4. Click the **Unlock User** tab.
5. In the **Unlock User** field, enter the User ID to unlock the user.

Figure 1–31 Unlock User



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

Figure 1–32 User Successfully Unlocked



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

1.9 Creation of first time user to access OBP

This section explains the procedure to create the first bank user having access to the application.

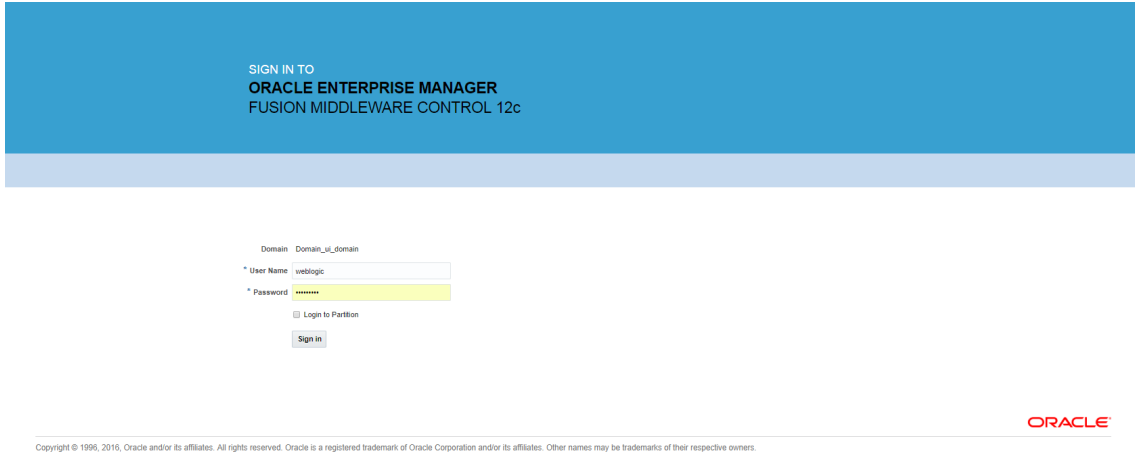
Note

Make the default authenticator as sufficient in host console and reorder it below OID Authenticator. Also change 'cn' attribute to 'uid' in the All Users Filter and User From Name Filter in OID Authenticator provider specific properties.

1. Log in to OIM using the admin user `xelsysadm`. Create a new role in OIM as described in [Section 1.2 Creating Roles in Oracle Identity Manager \(OIM\)](#). For example, Developer. This creates a group in OID (Developer).
2. Log in to admin application using the weblogic user. Create a user as described in [Section 1.7 User Management Using the Admin Application](#). For example, john.doe.

3. Add the user (john.doe) to the Developer.
4. Map the application role Administrators to the Enterprise Group Developer in EM (refer screenshots below). After doing this, the user should have access to all artifacts assigned to the 'Administrators' role. These access rights can be viewed in OES.

Figure 1–33 Log in Oracle Fusion Middleware Control



SIGN IN TO
ORACLE ENTERPRISE MANAGER
FUSION MIDDLEWARE CONTROL 12c

Domain Domain_id_domain

* User Name weblogic

* Password

Login to Partition

Sign in

ORACLE

Copyright © 1996, 2016, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Figure 1–34 Click Application Roles

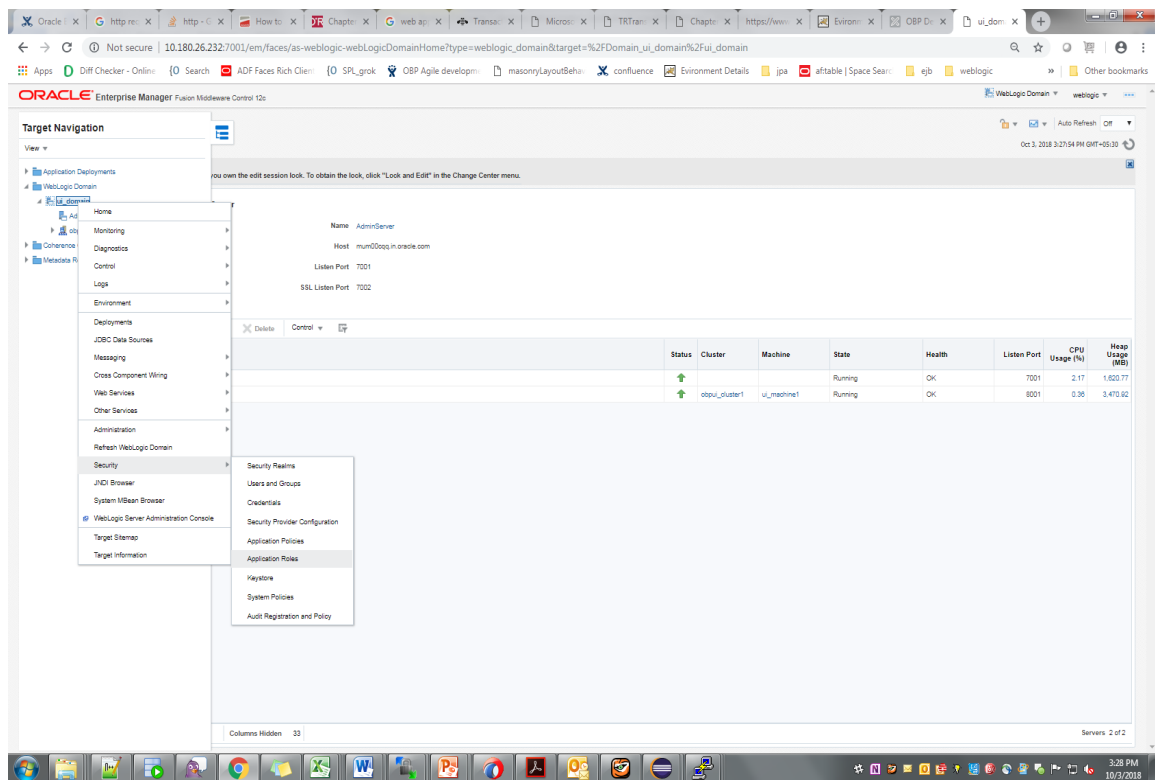


Figure 1–35 Select Administrators Role

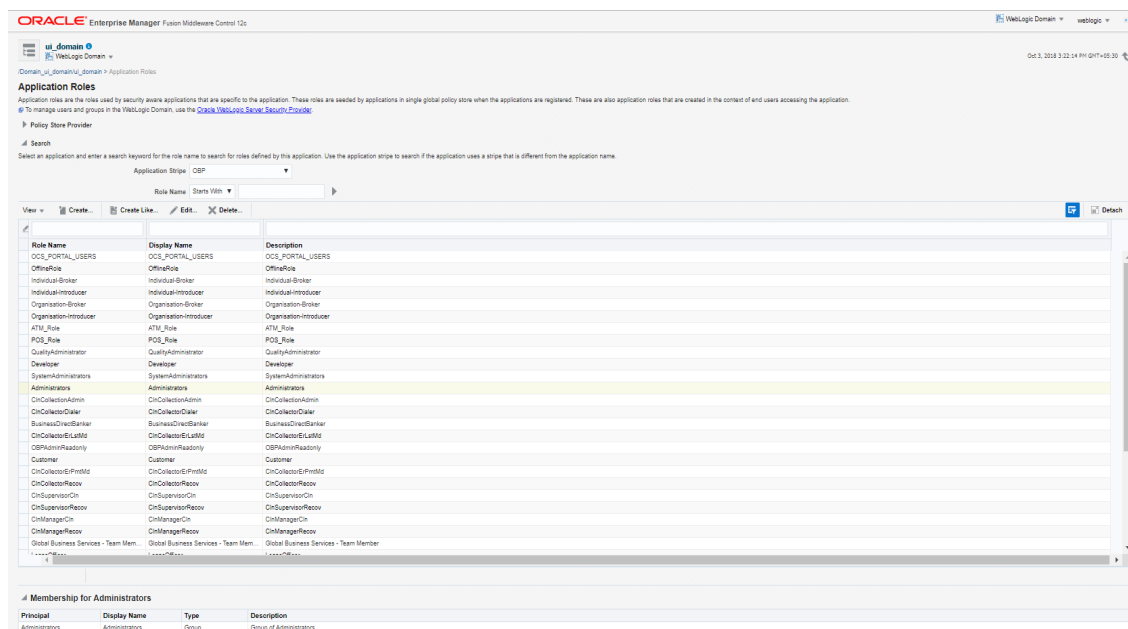


Figure 1–36 Add Principal

The screenshot displays the Oracle Enterprise Manager interface for editing an application role. The main page is titled "Edit Application Role : Administrators" and shows the role's details, including its name, display name, and description. A "Members" section is visible, showing a list of principals. Overlaid on this is the "Add Principal" dialog box, which allows for searching and selecting application roles to grant permissions to. The dialog includes search criteria, a table of searched principals, and an advanced option for entering a principal name directly.

Application Role Details:

- Application Stripe: OBP
- Role Name: Administrators
- Display Name: Administrators
- Description: Administrators

Members:

Principal	Display Name	Description
Developer	Developer	Group of Developer users

Add Principal Dialog:

Specify criteria to search and select the application roles that you want to grant permissions to.

Search:

- Type: Group
- Principal Name: Starts With: Developer
- Display Name: Starts With: []

Searched Principals:

Principal	Display Name	Description
Developer	Developer	Group of Developer users

Advanced Option:

- Check to enter principal name here instead of searching from above. This option can be used for advanced scenarios related to custom administrators.

2 Approvals Management

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

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 Functions for Fee Configuration

Sr.No	Function
1	Loan account configuration (New or Amend)
2	CASA account configuration
3	Term deposit account configuration
4	Overdraft account configuration (New or Account)
5	Origination or Application processing Fees

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 is called from the following screens:

Table 2–2 List of Functions for UDM Configuration

Sr. No	Function
1	Loan account configuration (New or Amend)
2	CASA account configuration
3	Term deposit account configuration
4	Overdraft account configuration (New or Account)

2.1.1 Setup Details

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

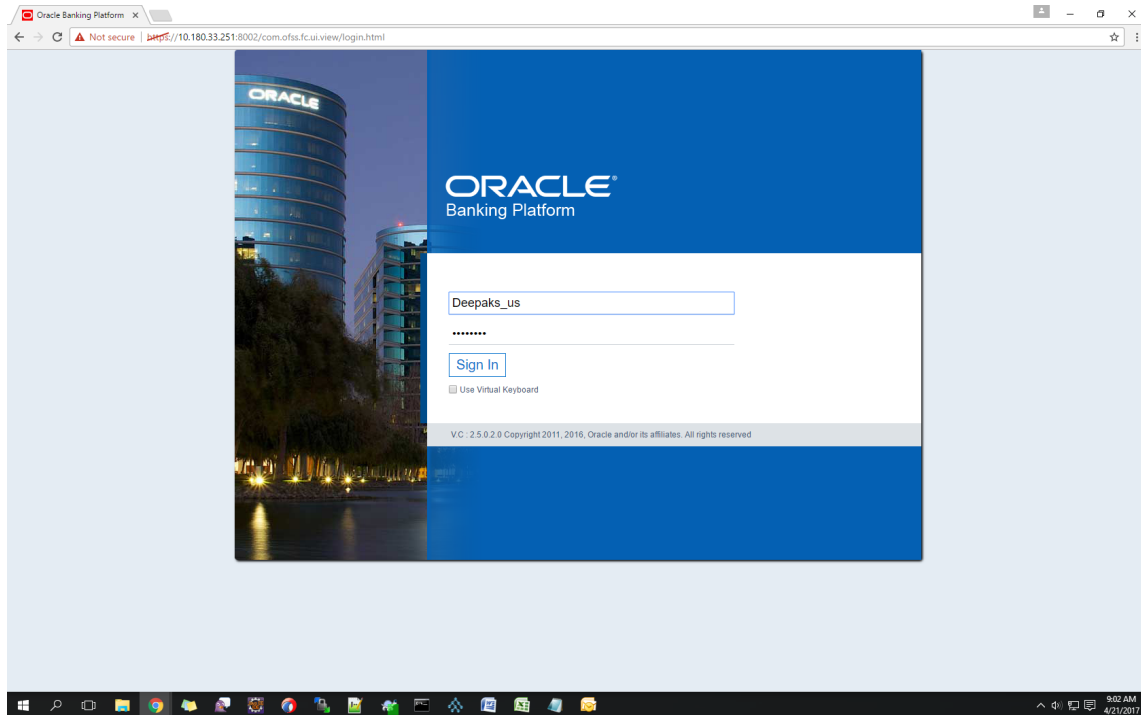
2.1.1.1 Policy Setup in UI

To initiate, the user needs to set up policies in UI 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 OBP UI policy setup:

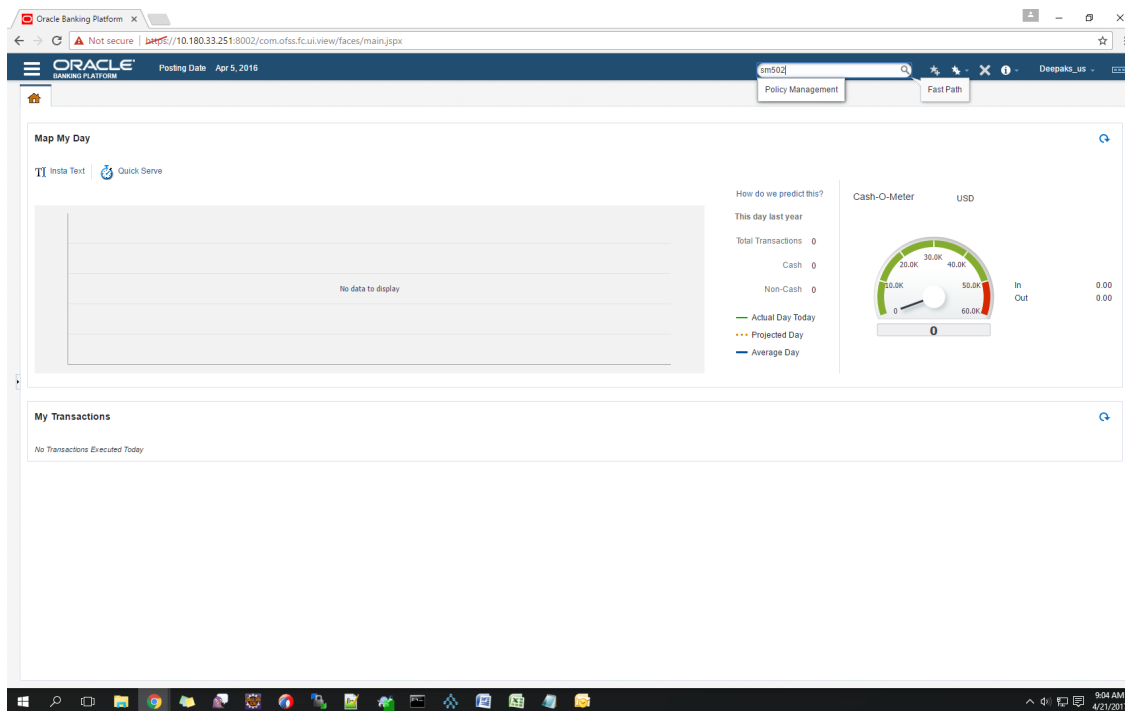
1. Log in to the OBP UI as a valid user.

Figure 2–1 Log in to OBP UI to Configure Policies



2. Enter SM502 in the fast path.

Figure 2–2 Search for SM502 in Fast Path



3. Click **Search**. The following screen appears.

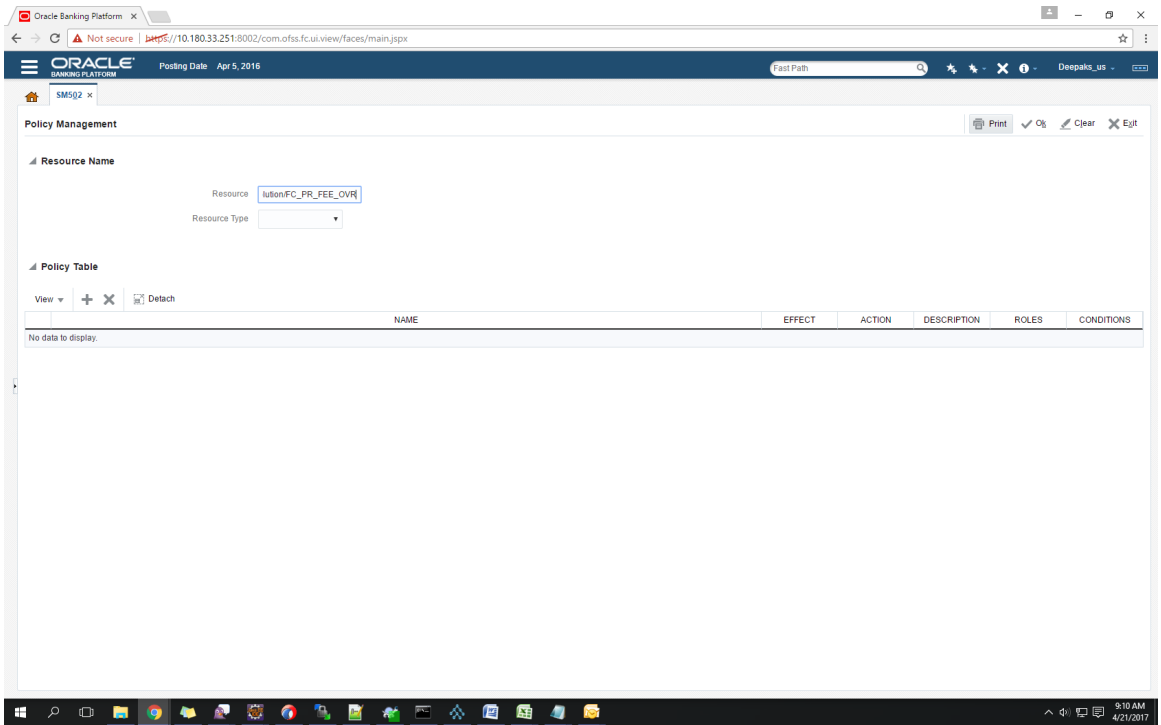
Enter the service for which policy needs to be defined.

The service needs to be given as [Service_ID]/FC_PR_FEE_OVR or [Service_ID]/FC_PR_UDM_OVR as shown above.

For example,

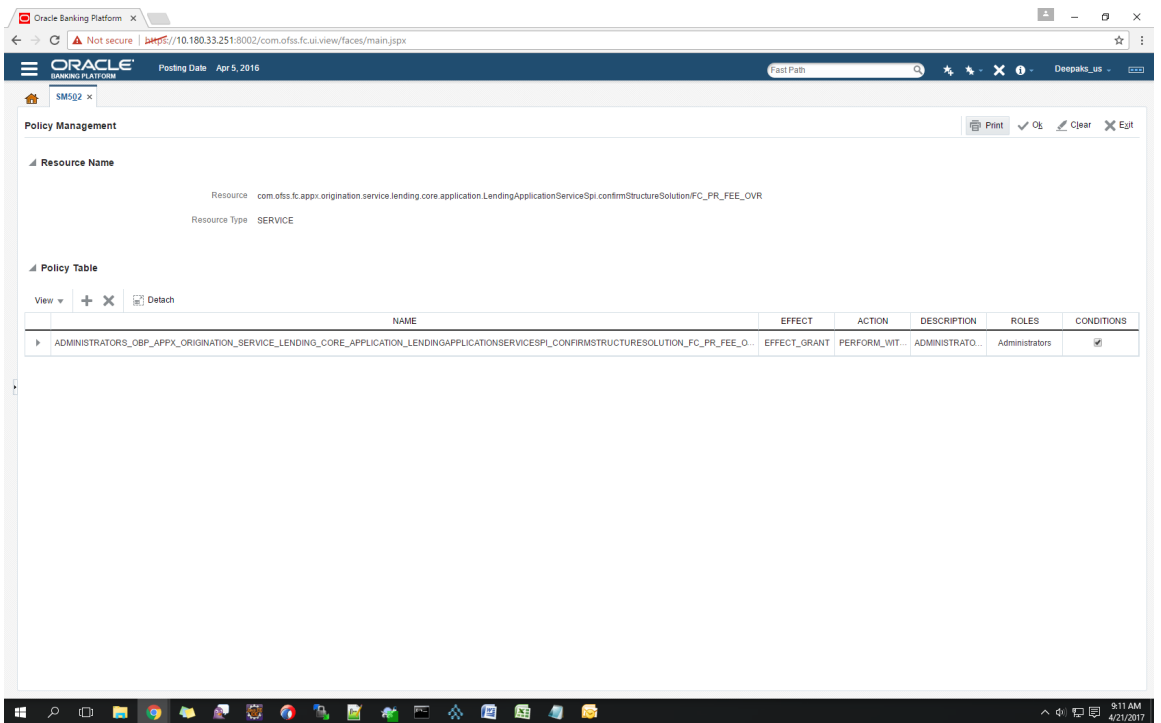
com.ofss.fc.appx.origination.service.lending.core.application.LendingApplicationServiceSpi.confirmStructureSolution/FC_PR_FEE_OVR

Figure 2–3 Policy Management



The following screen appears:

Figure 2–4 Enter Service for Policy Definition



- 4. Click the + sign if you need to add a new policy along with the Role for which the policy is to be enabled.

Figure 2-5 Effect of the Policy

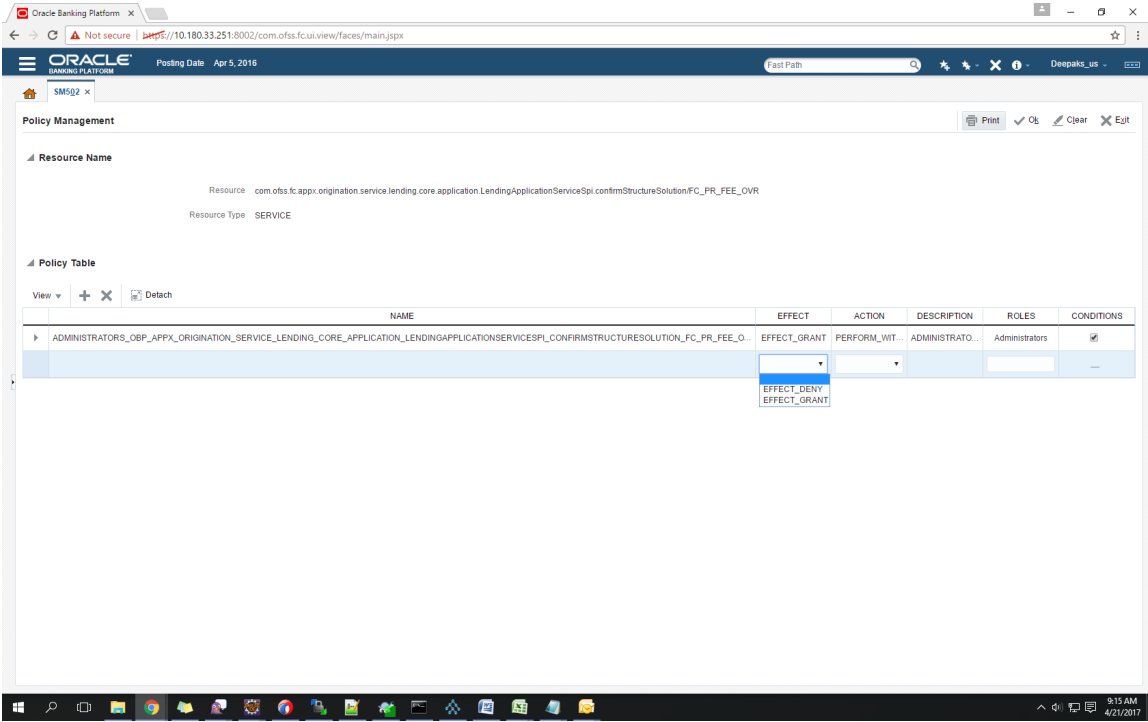
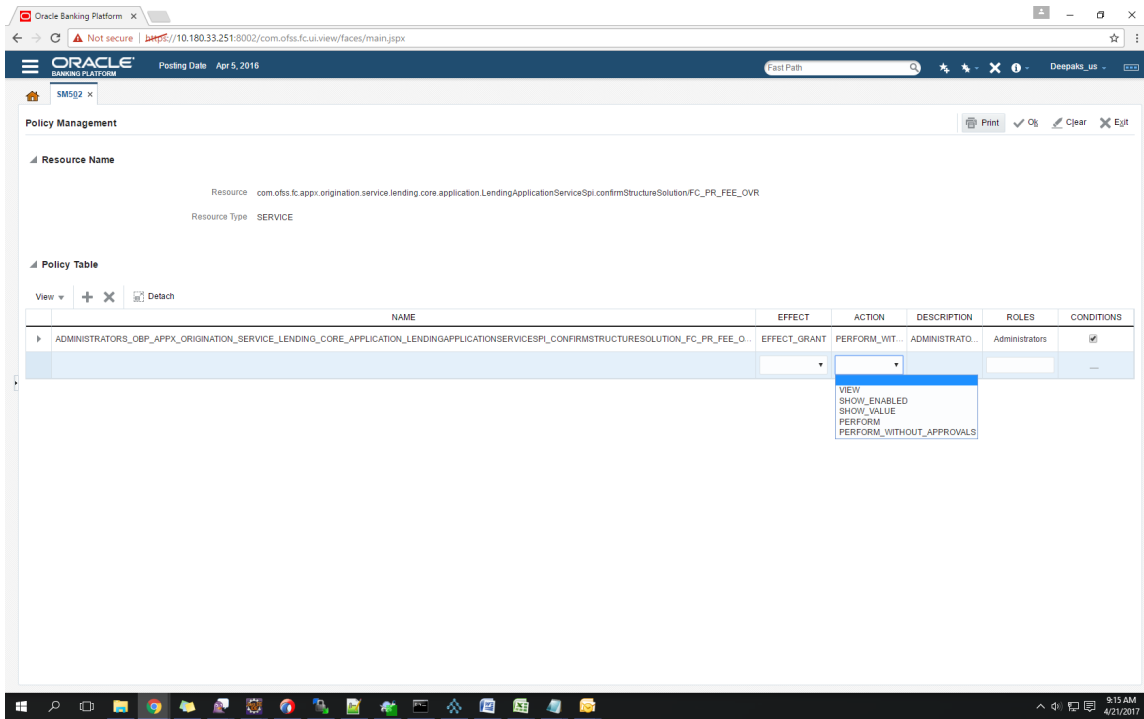
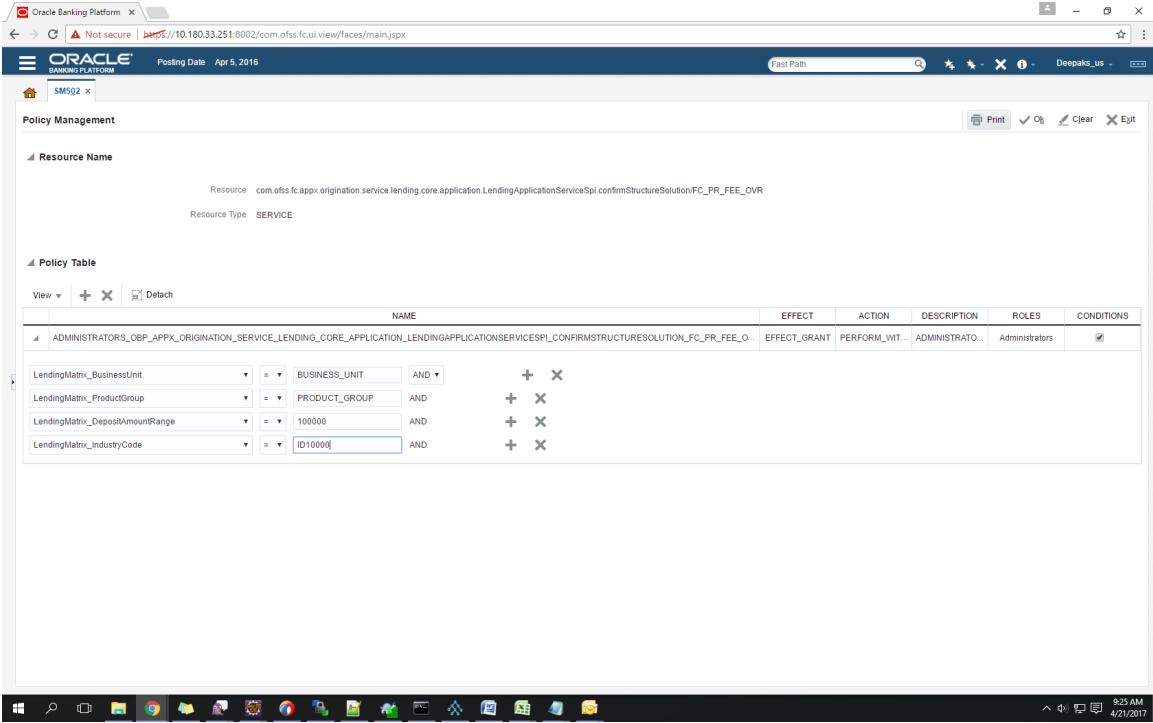


Figure 2–6 Action of the Policy



5. To add conditions, select the check box in the conditions column. This displays a collapsible section arrow in the first column. Click the arrow to display the conditions panel as shown below. Enter the conditions needed for the policy.

Figure 2–7 Conditions of the Policy



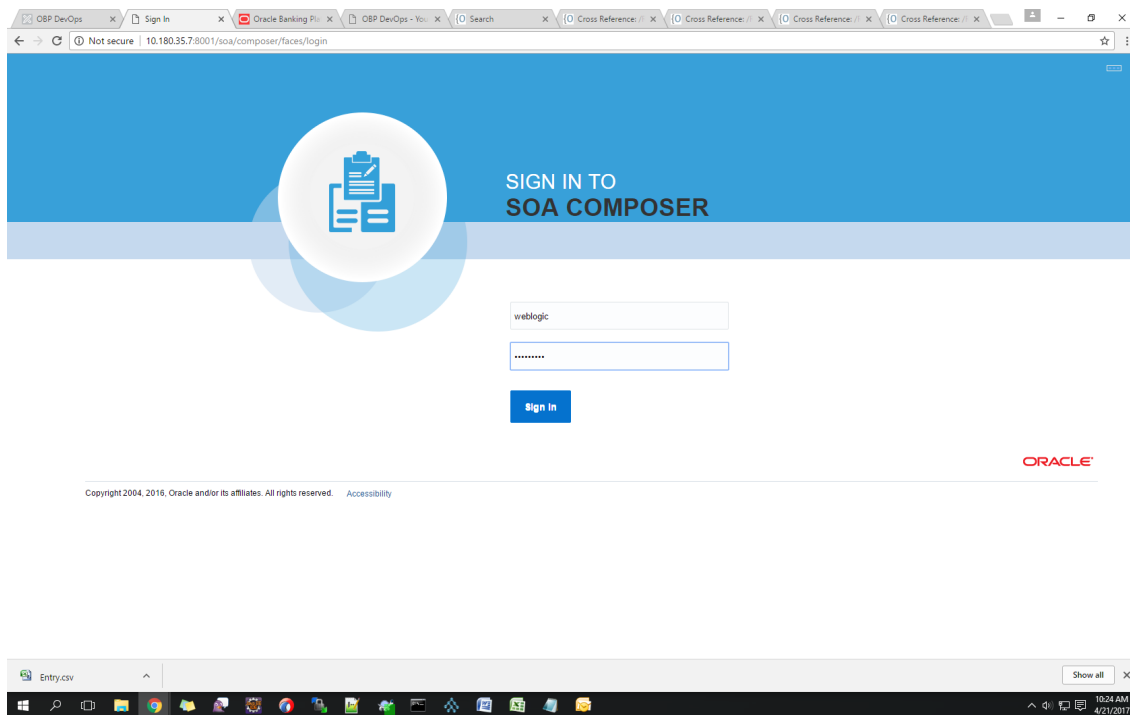
2.1.1.2 SOA Composer Rules Setup

After the rules are set up in UI, you must set up approval rules in SOA composer. These rules should cover all scenarios in which the transaction can come out without being auto approved at the policy level. Any transaction which does not trigger the rules at SOA Composer level is auto rejected.

The following procedure is performed during SOA Composer rules setup:

1. Log in to the SOA Composer application as a user with administrative privileges.

Figure 2–8 Log in to SOA Composer



2. In the **Filter** field in the left panel, enter the name of the rule for the service.

For example:

HT_LendingSpi_ConfirmStructureSolutionRules.rules

Figure 2-9 Find the Rules for Service

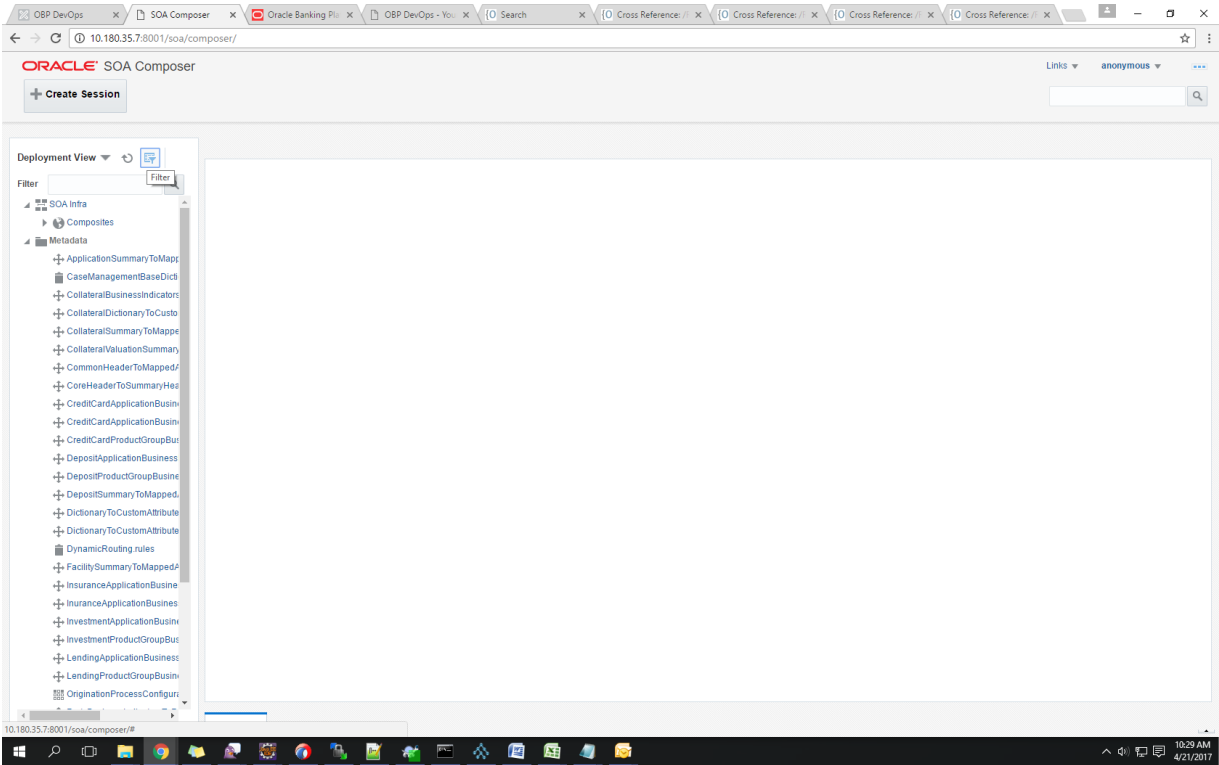
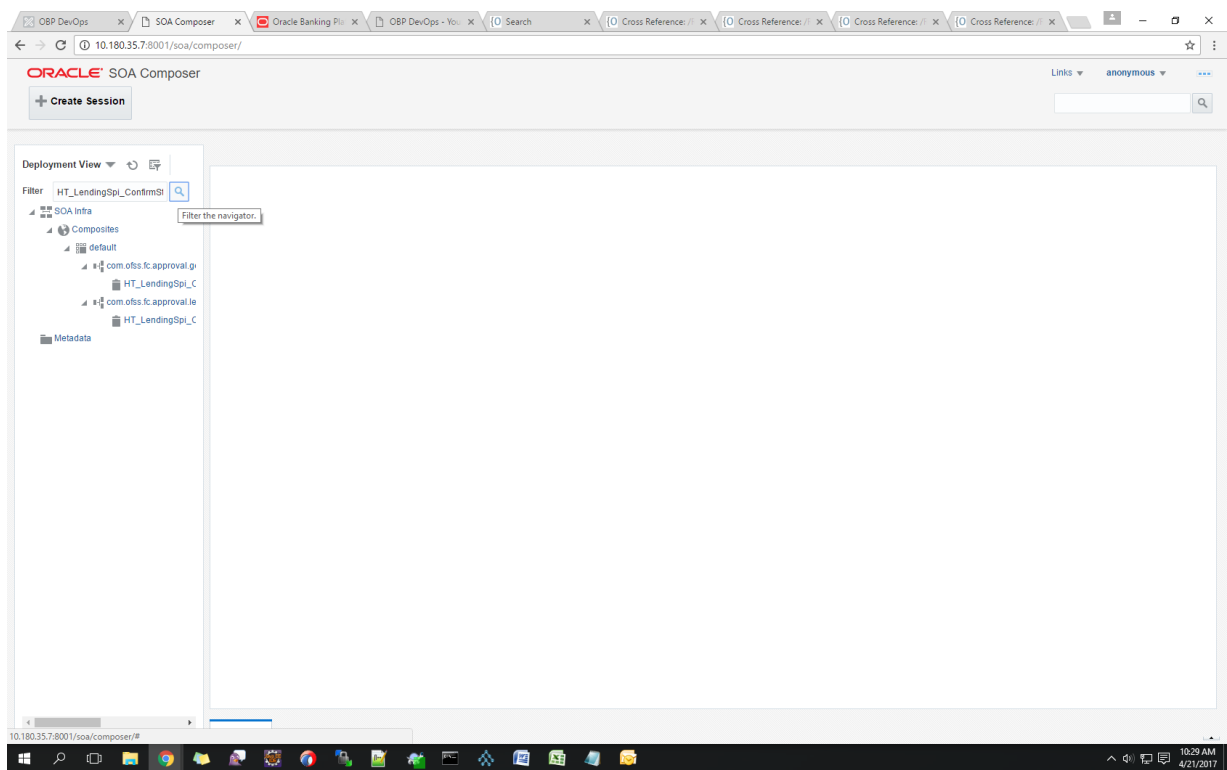
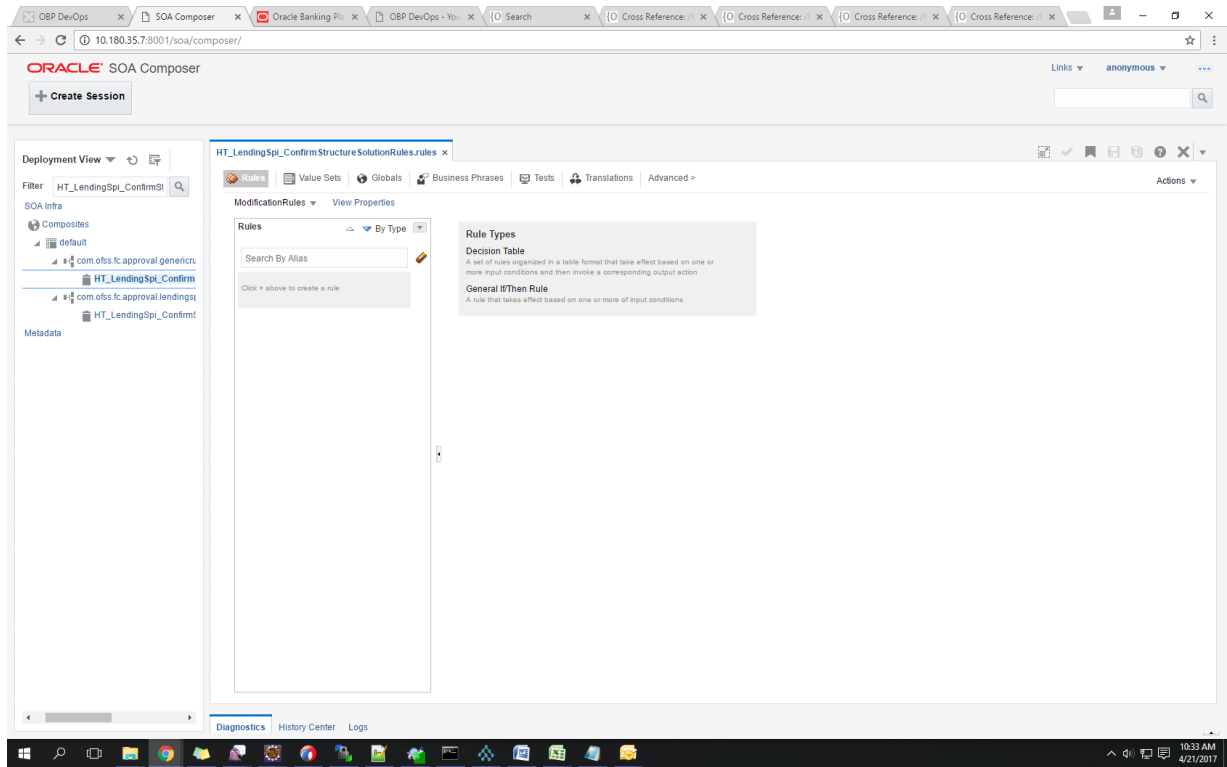


Figure 2–10 Filter Rules



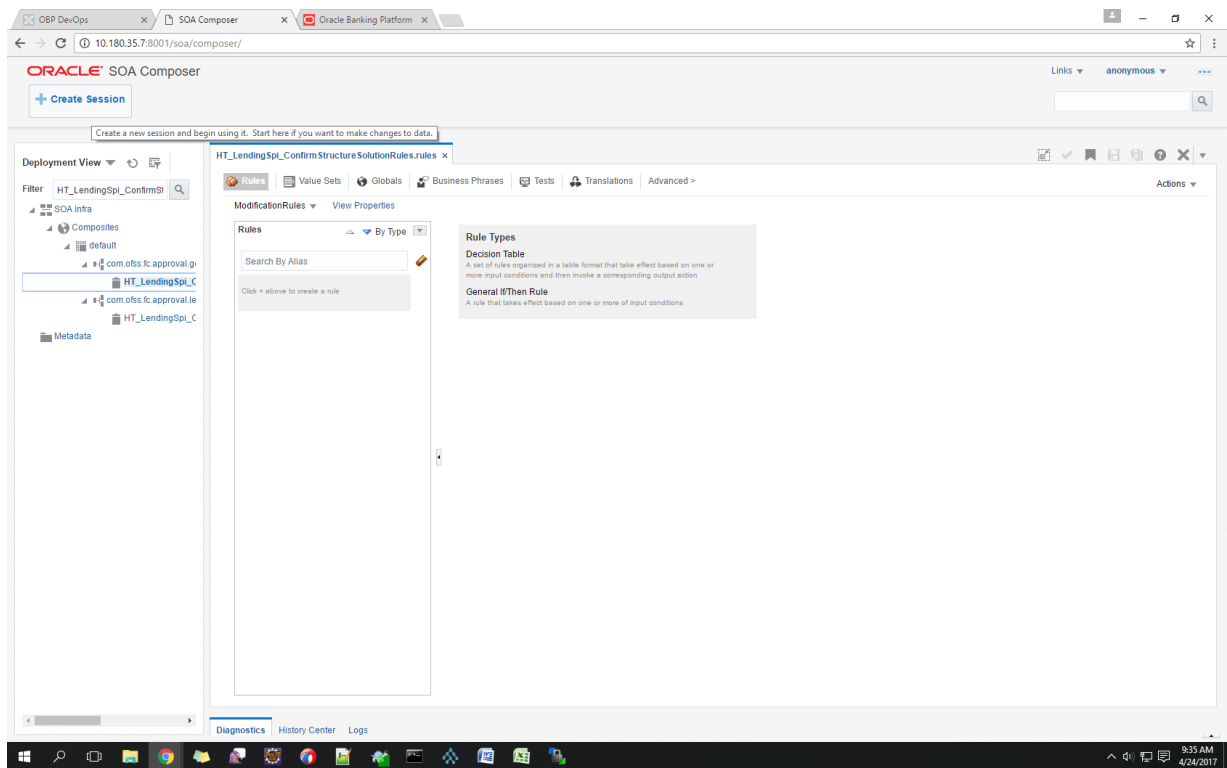
3. Click the rules file present in the process beginning with name `com.ofss.fc.approval.genericrulesapprovalspi.executeapprovalrules`.

Figure 2–11 Searching Specific Process



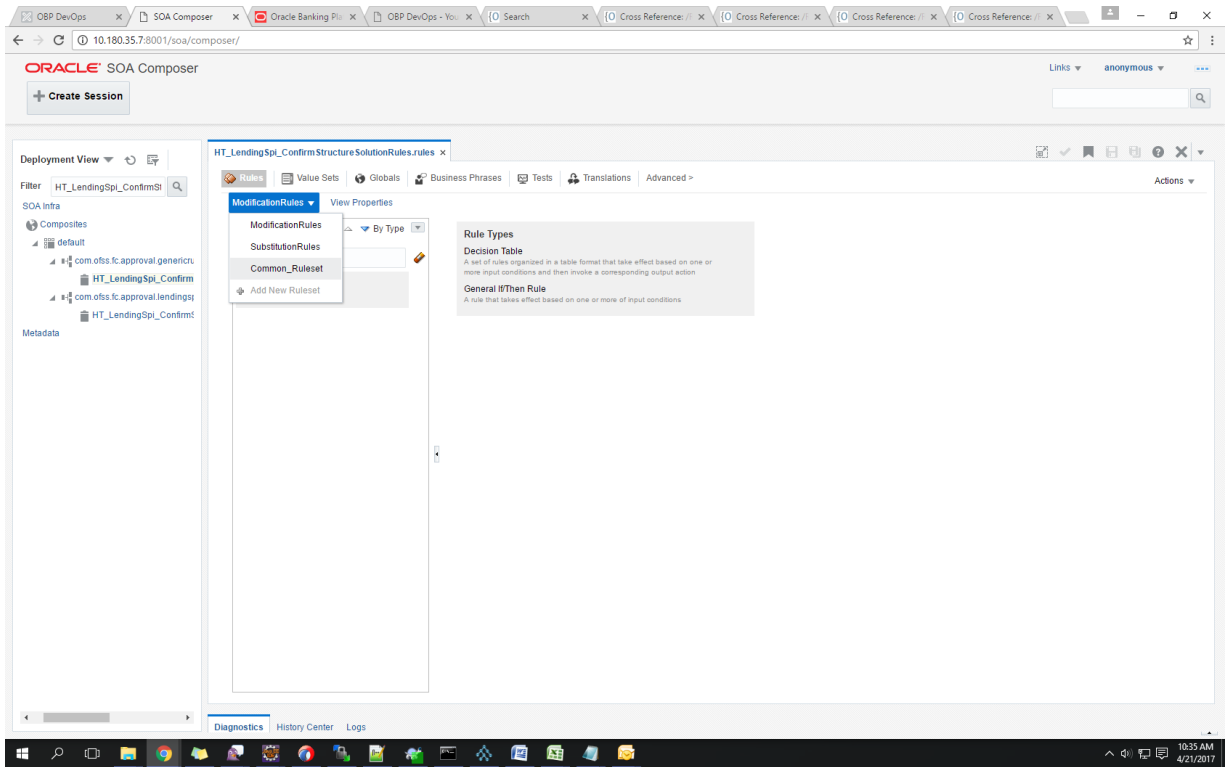
4. Click the **Create Session** tab to modify the rules.

Figure 2–12 Creating the Rule



5. From the **ModificationRules** list, select the **Common_Ruleset** option as shown below.

Figure 2–13 Selecting Rules File



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

Note

The rule being created must be Active rule.

Figure 2–14 Adding Attributes to the Rule File

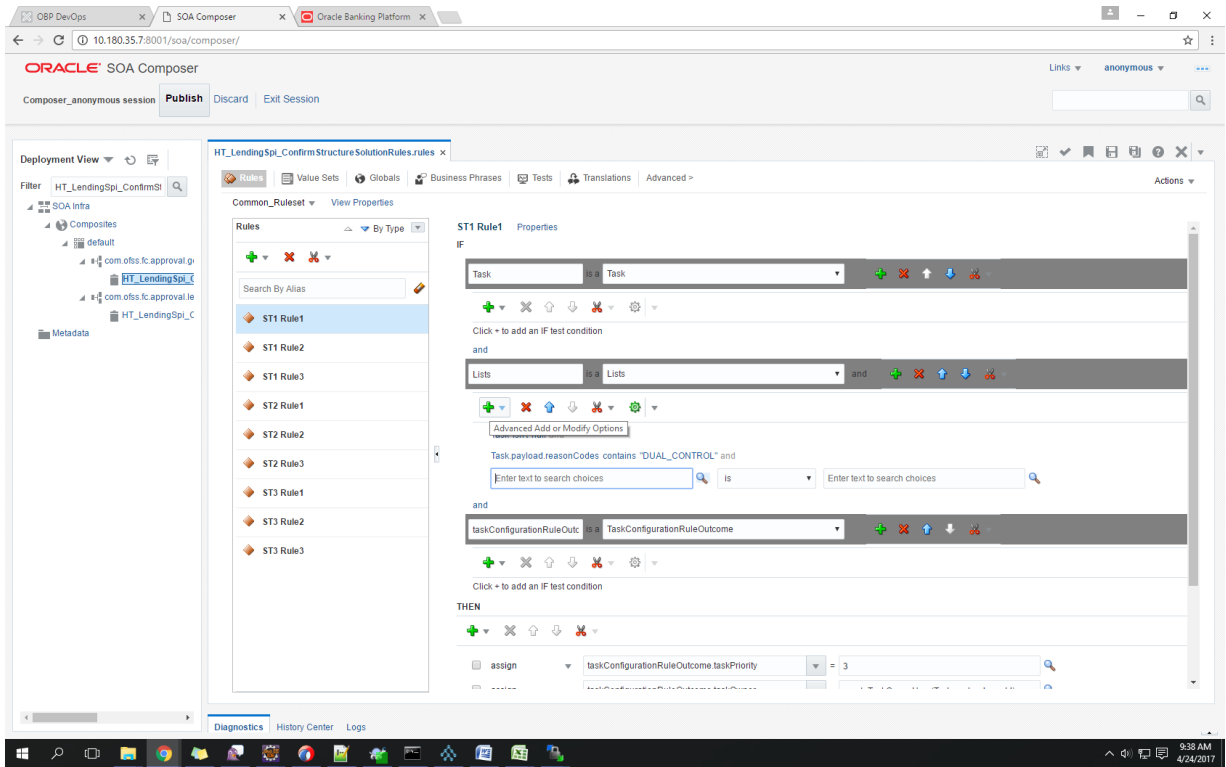
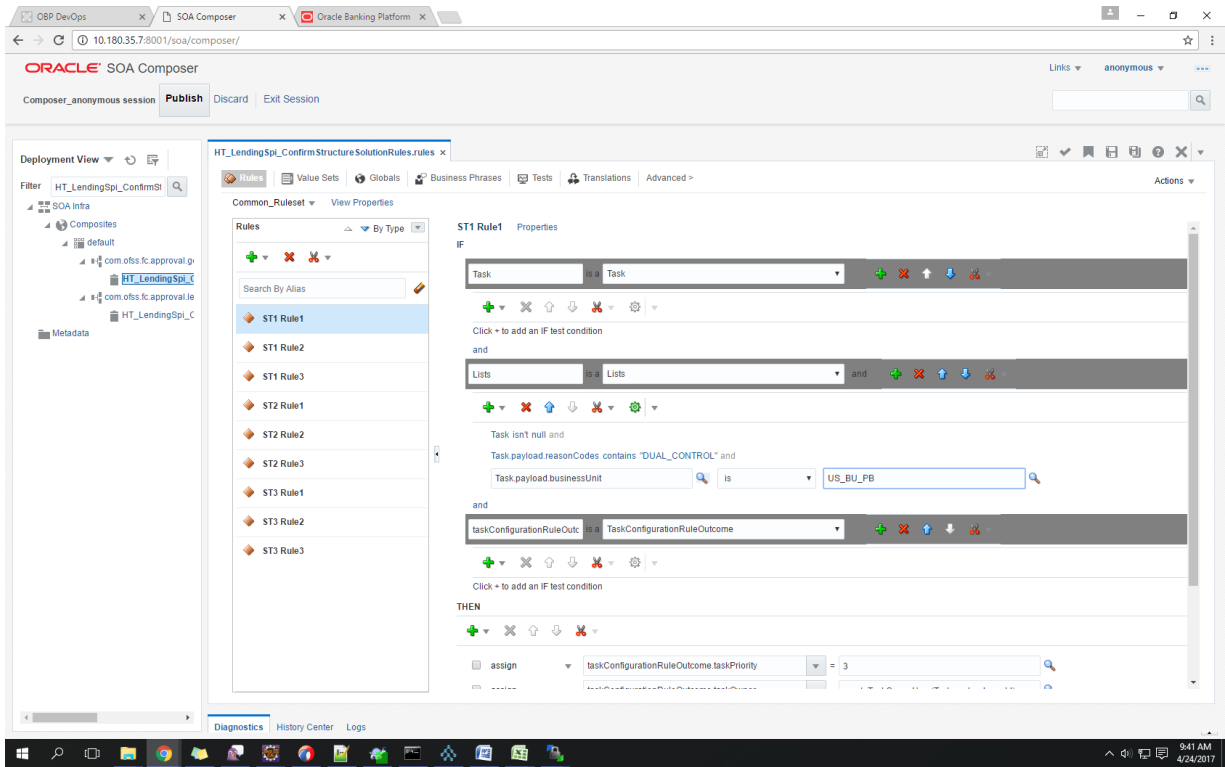


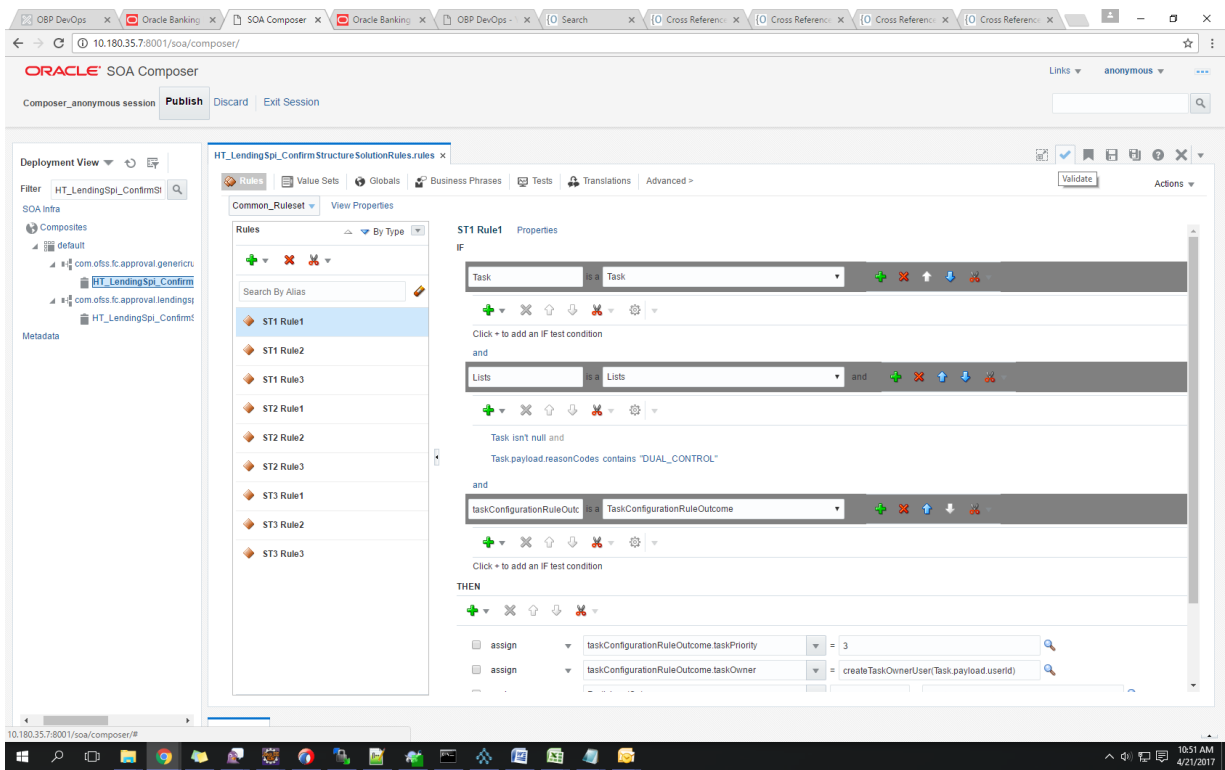
Figure 2–15 Adding Attributes to the Rule File



7. Click **Validate** to validate the rules file.

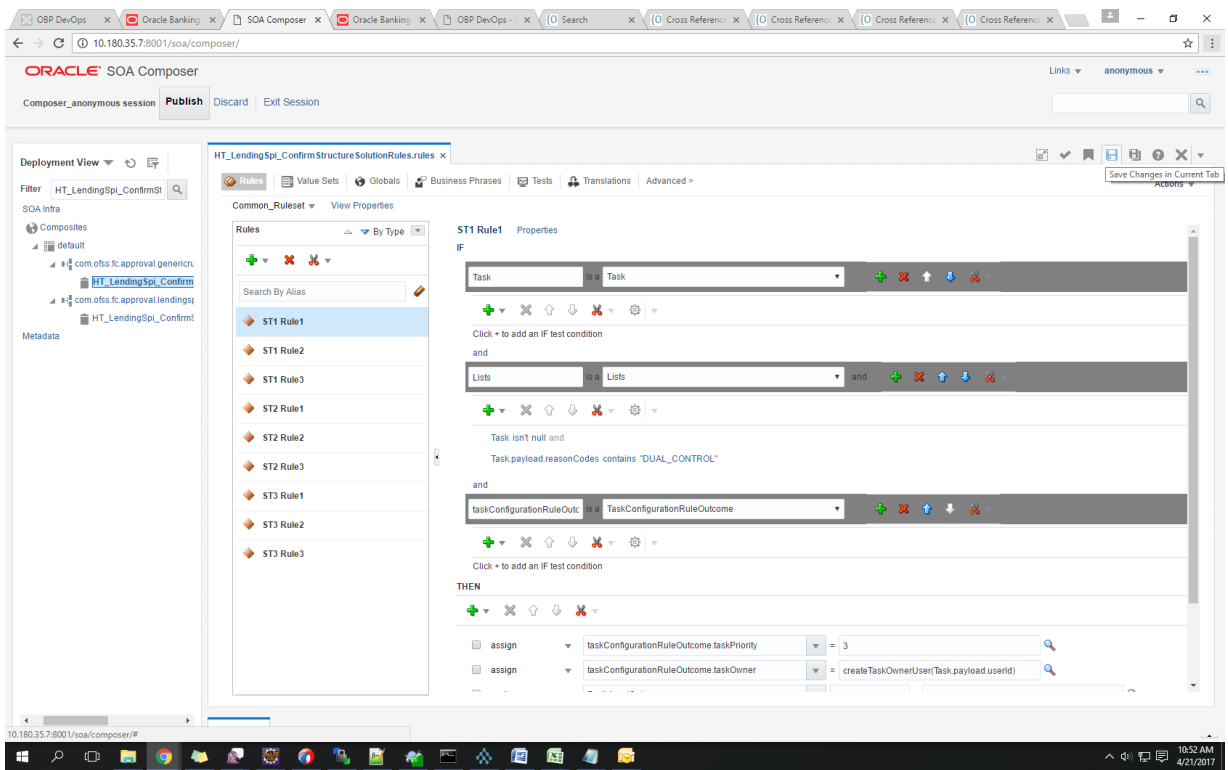
2.1 Discretionary Pricing Assessment (DPA)

Figure 2–16 Validating Rules File



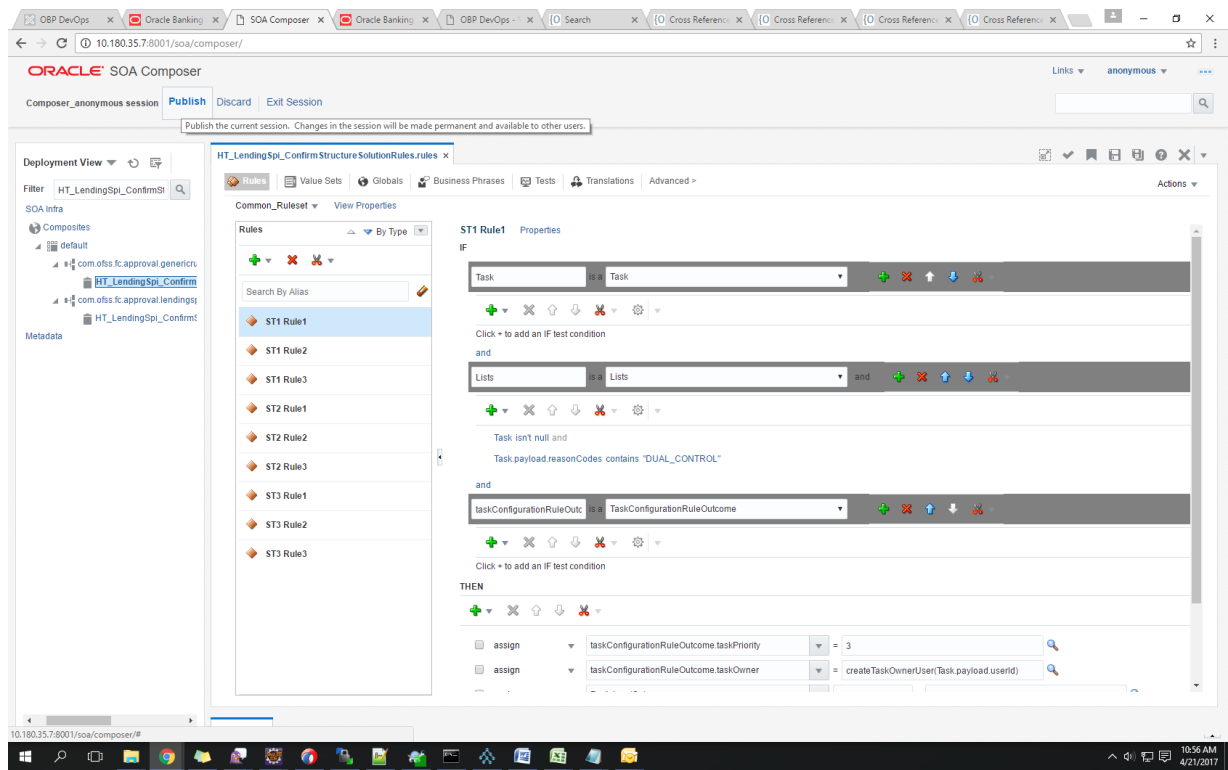
8. Click **Save** to save the rules.

Figure 2–17 Saving Rules File



9. Click **Publish** to publish the rules once editing of rules is done.

Figure 2–18 Publishing Rules File



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** tab in **Artifact Dependency Map (Fast Path: SM500)** page.

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.

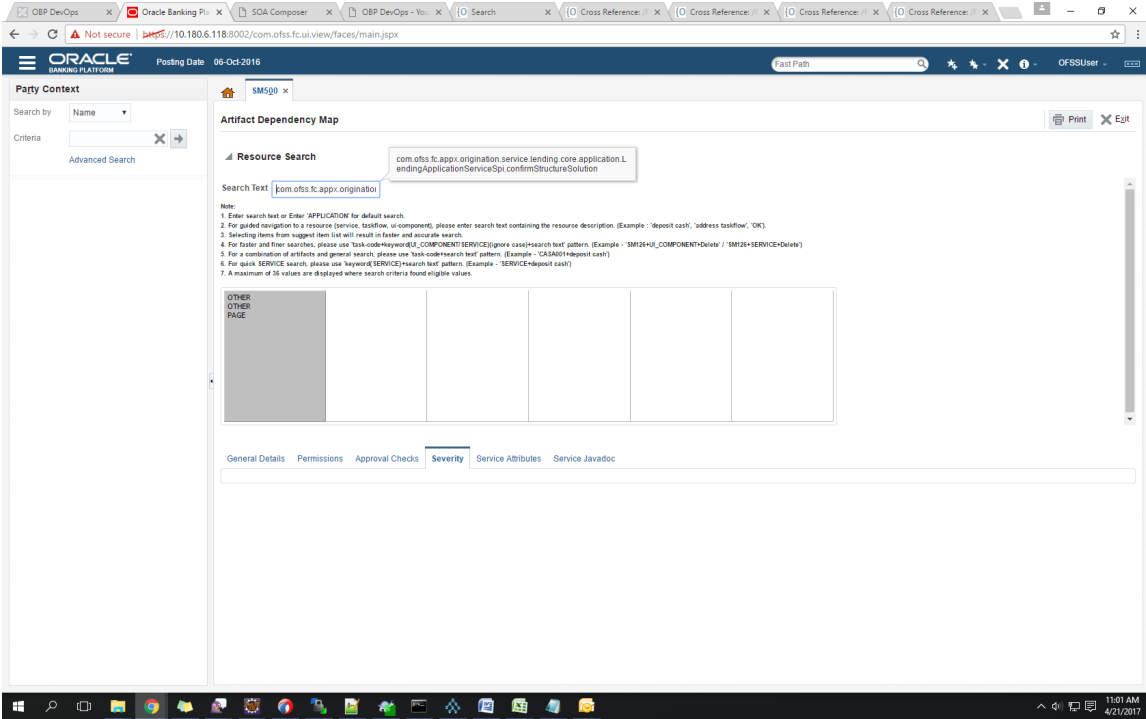
The following procedure is performed during SMS setup:

1. In the **Search Text** field, enter the service name.

For example, to search for a service, search text can be as follows:

- com.ofss.fc.appx.Origination.service.lending.core.application.LendingApplicationServiceSpi.confirmStructureSolution

Figure 2–19 Search for Service Using Entire Service Name



- 2. Navigate to service node by following highlighted path (in grey color) and select the service node.

2.1 Discretionary Pricing Assessment (DPA)

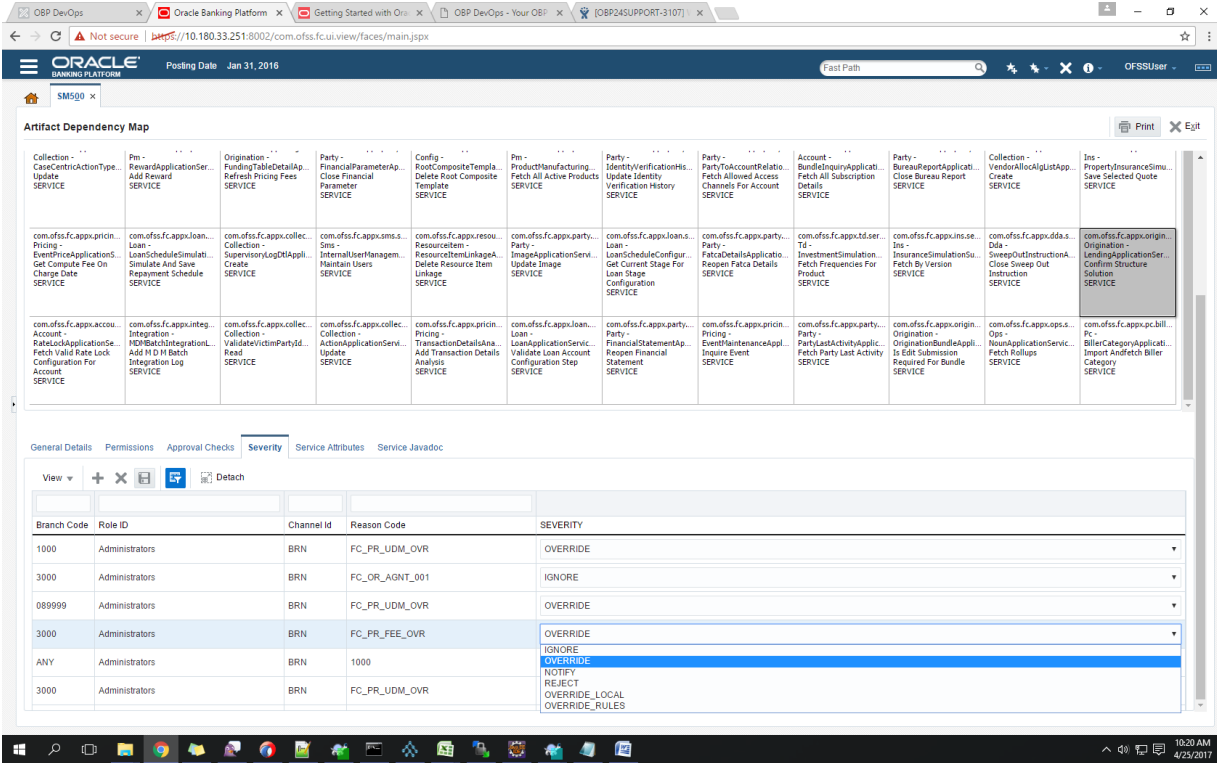
Figure 2–20 Select the Service to be Configured

The screenshot displays the Oracle Banking Platform Administrator interface. At the top, there are browser tabs for 'OBP DevOps', 'Oracle Banking Platform', 'Getting Started with OBP', 'OBP DevOps - Your OBP', and 'OBP24SUPPORT-3107'. The main window shows the 'Artifact Dependency Map' for the application 'OTHER - OTHER-PageDefn - OTHER page'. Below this, a grid of service artifacts is visible, including services like 'com.ofss.fc.app.collec...', 'com.ofss.fc.app.pricing...', and 'com.ofss.fc.app.loan...'. The 'Severity' tab is selected, showing a table with columns for Branch Code, Role ID, Channel Id, Reason Code, and SEVERITY. The table contains several rows with values such as '1000', 'Administrators', 'BRN', and 'FC_PR_LIDM_OVR'. A toolbar at the bottom of the table includes buttons for 'View', 'Add', 'Delete', and 'Save'.

Branch Code	Role ID	Channel Id	Reason Code	SEVERITY
1000	Administrators	BRN	FC_PR_LIDM_OVR	OVERRIDE
3000	Administrators	BRN	FC_OR_AIGNT_001	IGNORE
089999	Administrators	BRN	FC_PR_LIDM_OVR	OVERRIDE
3000	Administrators	BRN	FC_PR_FEE_OVR	OVERRIDE
ANY	Administrators	BRN	1000	IGNORE

3. Click the **Severity** tab to maintain severity for the resource. Severity can be maintained using add row, delete, and save buttons in the toolbar.

Figure 2–21 Severity Tab - Add Severity



2.1.2 Performing the Transaction

Following procedure explains how to perform a transaction:

2.1 Discretionary Pricing Assessment (DPA)

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

Figure 2–22 Initiating Transaction

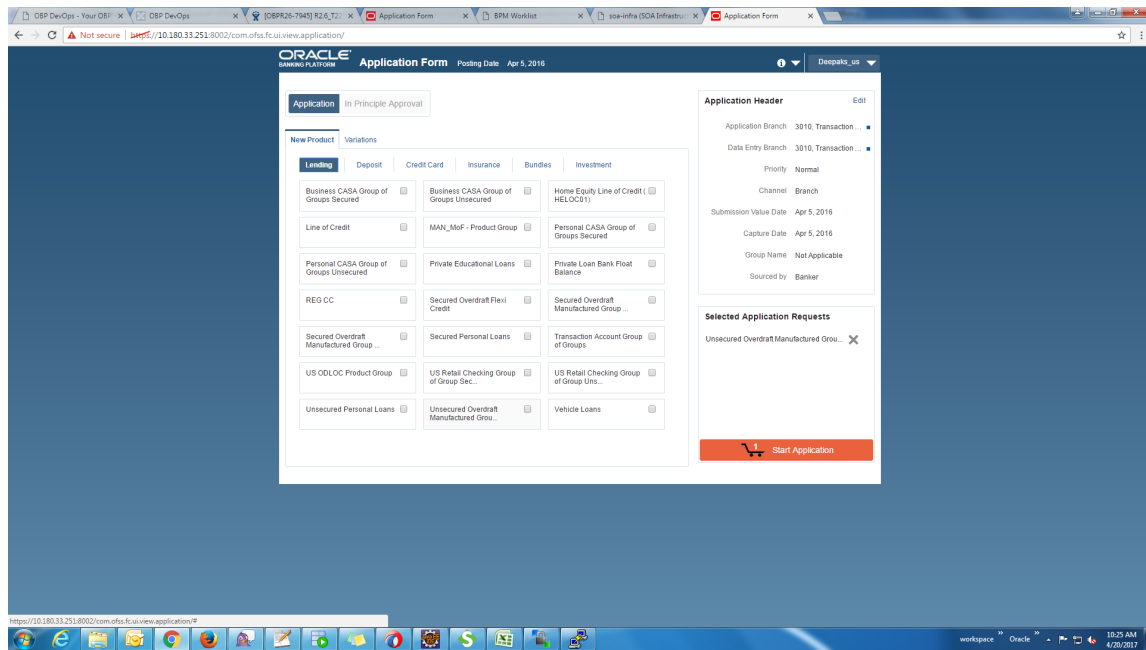
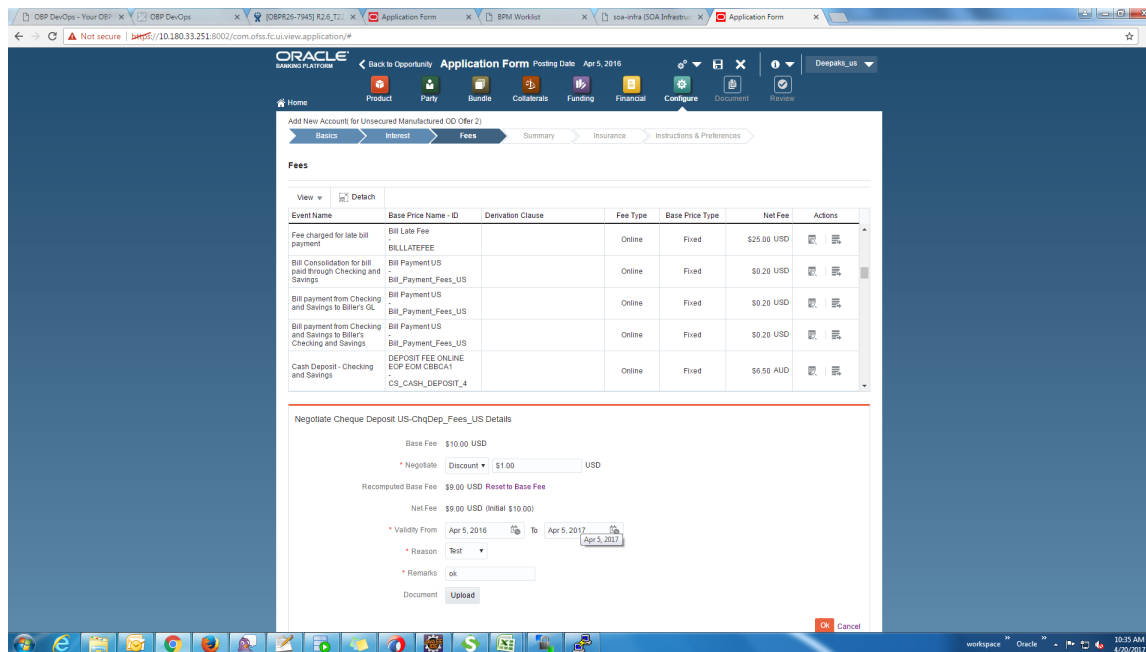
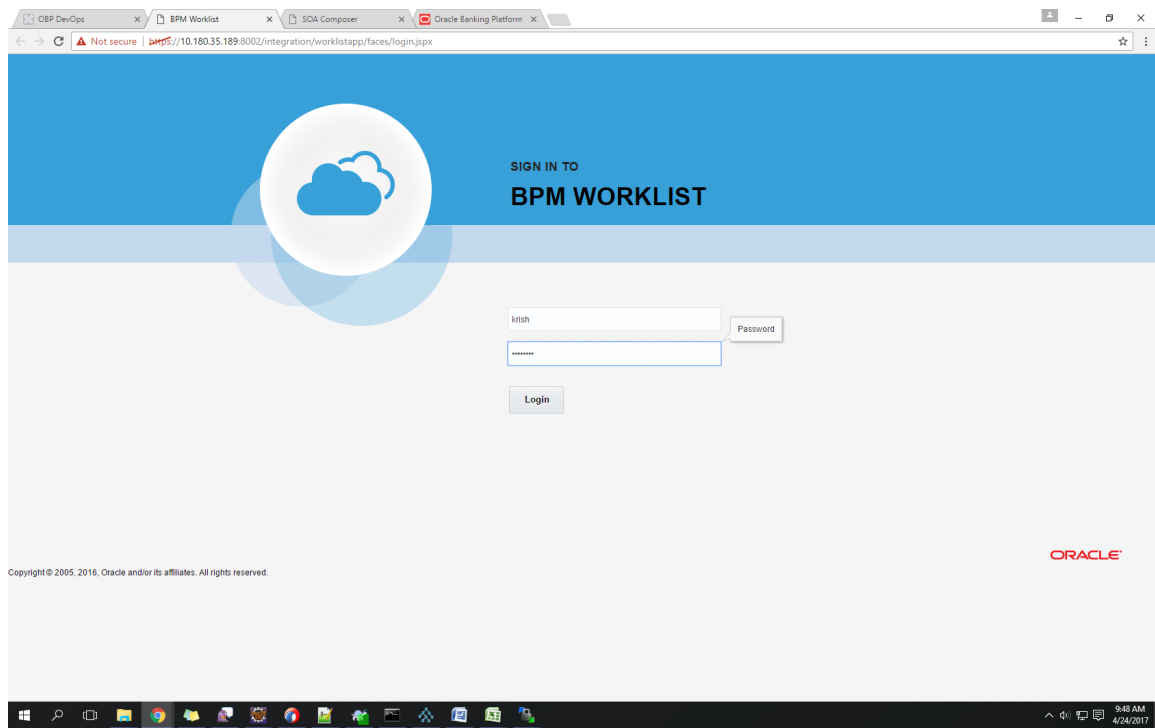


Figure 2–23 Fee Negotiation



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

Figure 2–24 Log in to Approval Worklist



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

2.1 Discretionary Pricing Assessment (DPA)

Figure 2–25 Approve or Reject Work item

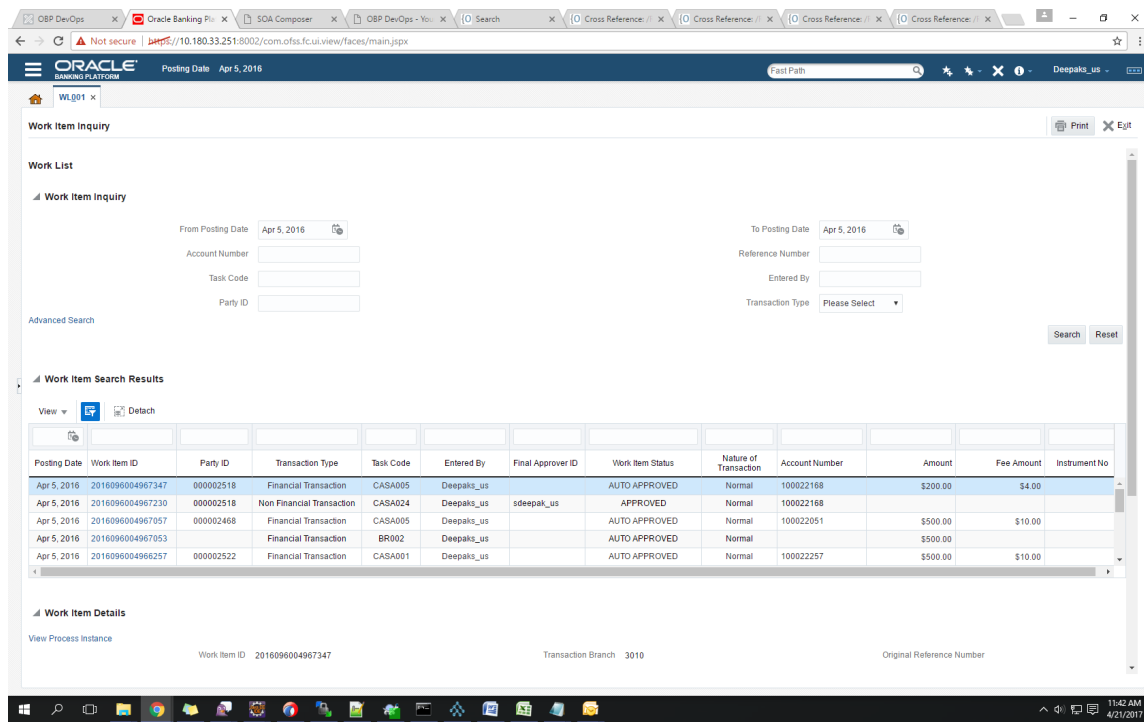
The screenshot displays the Oracle BPM Worklist interface. At the top, there are browser tabs for 'OBP DevOps', 'Oracle Banking Platform', 'Application Form', and 'BPM Worklist'. The main header shows 'ORACLE BPM Worklist' and a user profile for 'krish'. Below the header is a navigation bar with 'Views' and 'Actions' menus. The main content area features a table of work items with columns for 'Status', 'Title', 'Number', 'Creator', 'Assigned', and 'Priority'. The first row is highlighted, and its details are shown in a modal window below. The modal window title is 'Approval Required for Structure Solution (FC_PR_Disbursement_Fee+LN_DISBURSEMENT_FEE+LN_DISBURSEMENT_FEE_OVR) of Lending Application'. It includes a description of the work item, a 'Funds Table' with columns for 'Outlay' and 'Funds', and an 'Actions' menu with options like 'Approve', 'Reject', 'Create Work Activity', 'Request Information...', 'Reassign...', 'Create Subtask...', 'Escalate', 'Renew', and 'Suspend'. The 'Funds Table' shows the following data:

Outlay	Funds
▶ Purchase or Cost	\$10,000.00
▶ Bank Fees	\$585.60
▶ Misc Fees	\$165.00

4. View the status of the work item.

The initiator user can see the status of the work item from the **Work Item Inquiry (Fast path: WL001)** page (only for non origination task).

Figure 2–26 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
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.

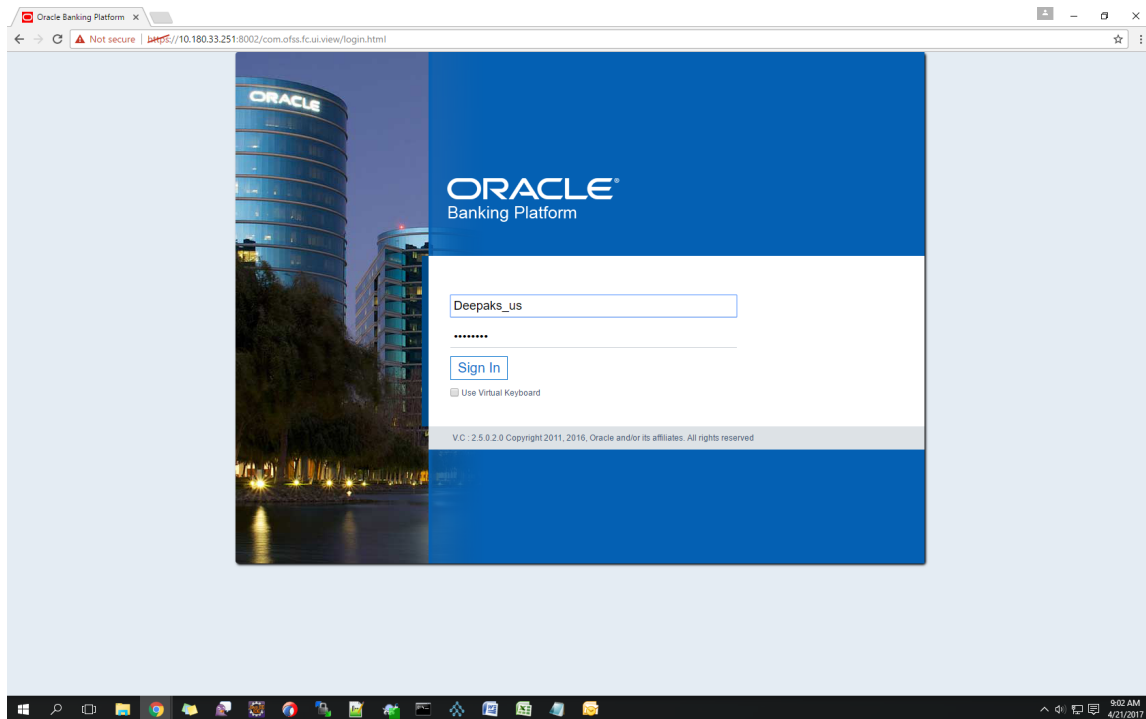
2.2.1.1 Policy Setup in UI

To initiate, the user needs to set up policies in UI 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 OBP UI policy setup:

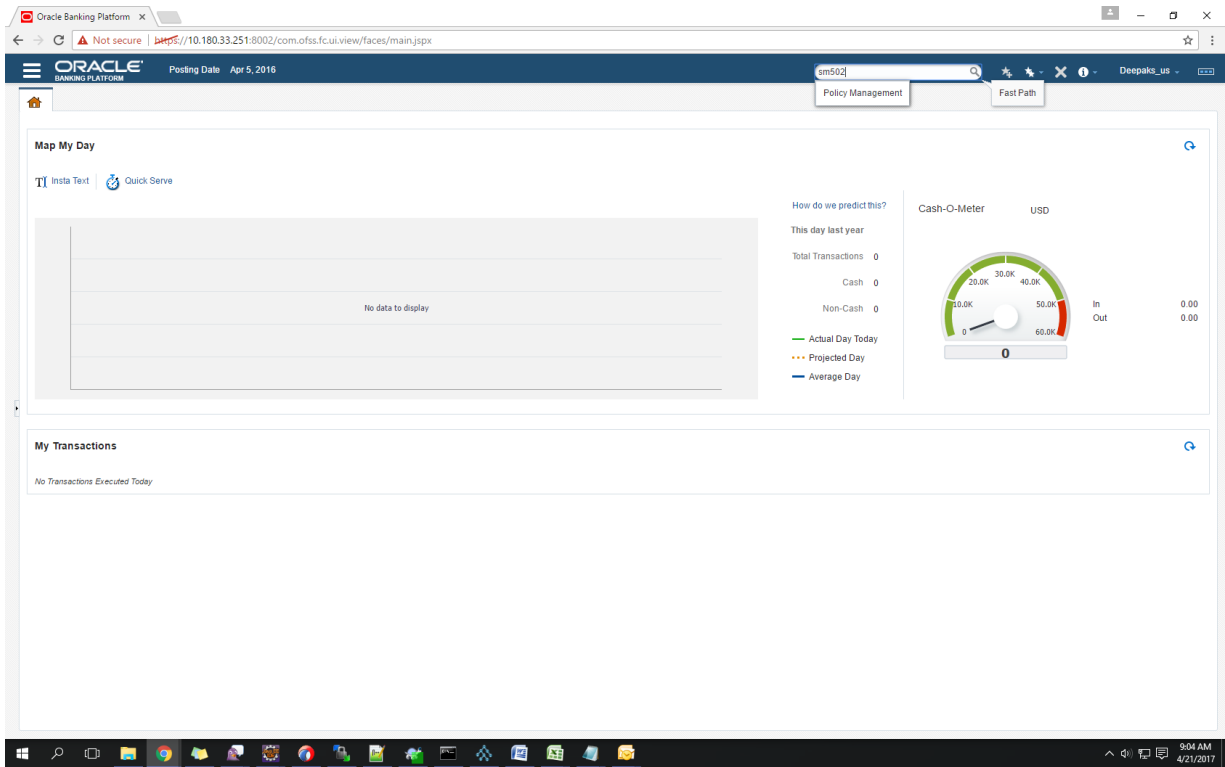
1. Log in to the OBP UI as a valid user.

Figure 2–27 Log in to OBP UI to Configure Policies



2. Enter SM502 in the fast path.

Figure 2–28 Search for SM502 in Fast Path

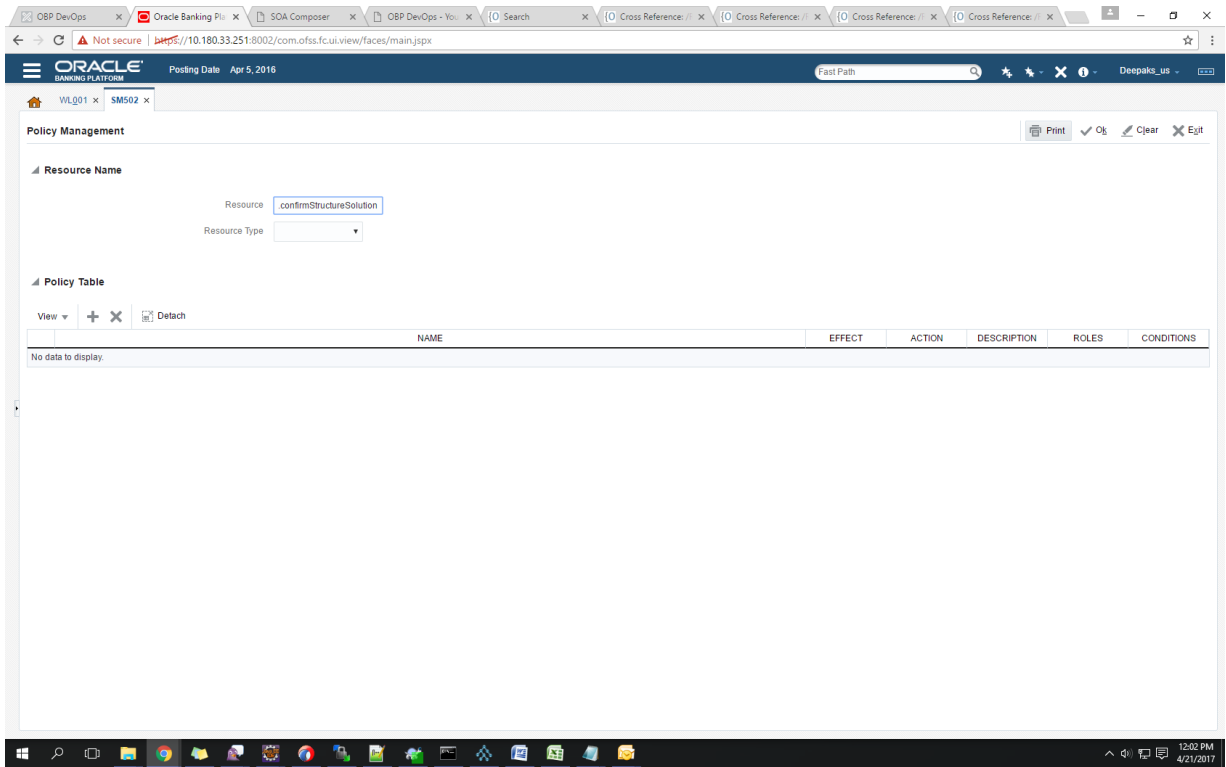


3. Click **Search**. The following screen appears.

Enter the service for which policy needs to be defined.

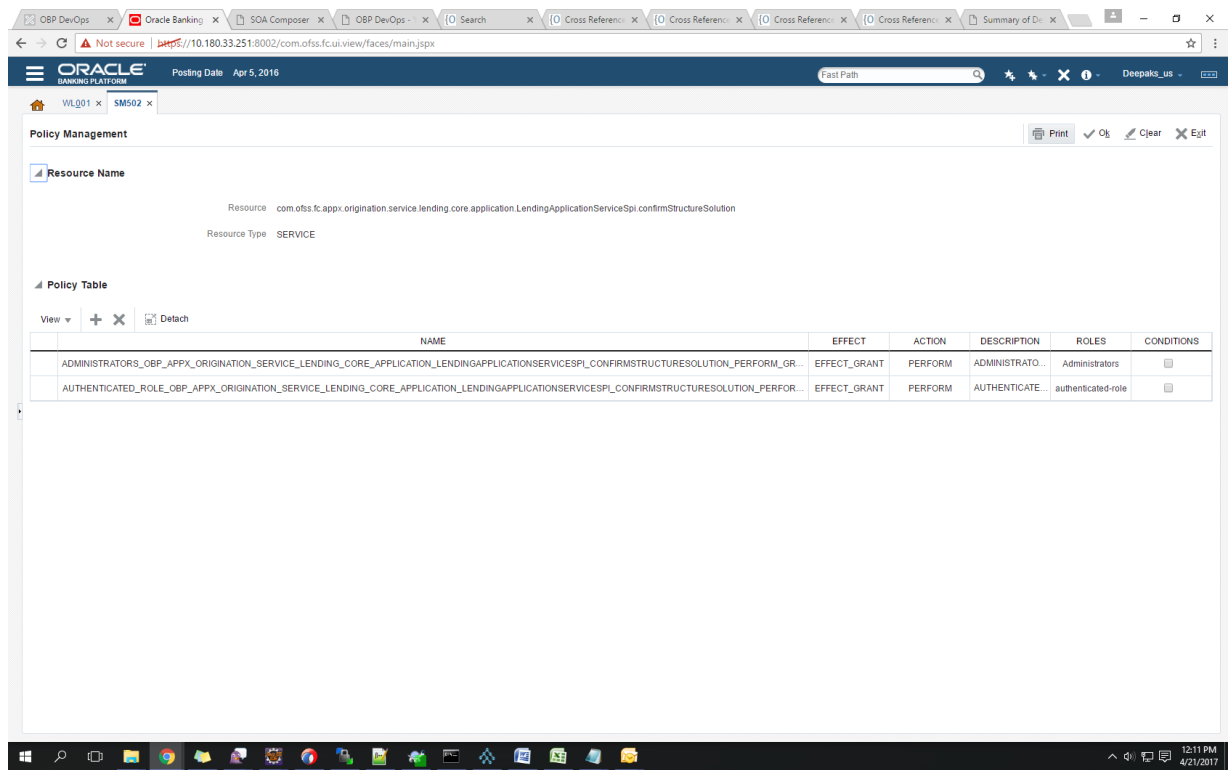
For example,
`com.ofss.fc.appx.OriginationService.Lending.Core.Application.LendingApplicationServiceSpi.confirmStructureSolution.`

Figure 2–29 Policy Management



The following screen appears:

Figure 2–30 Enter Service for Policy Definition



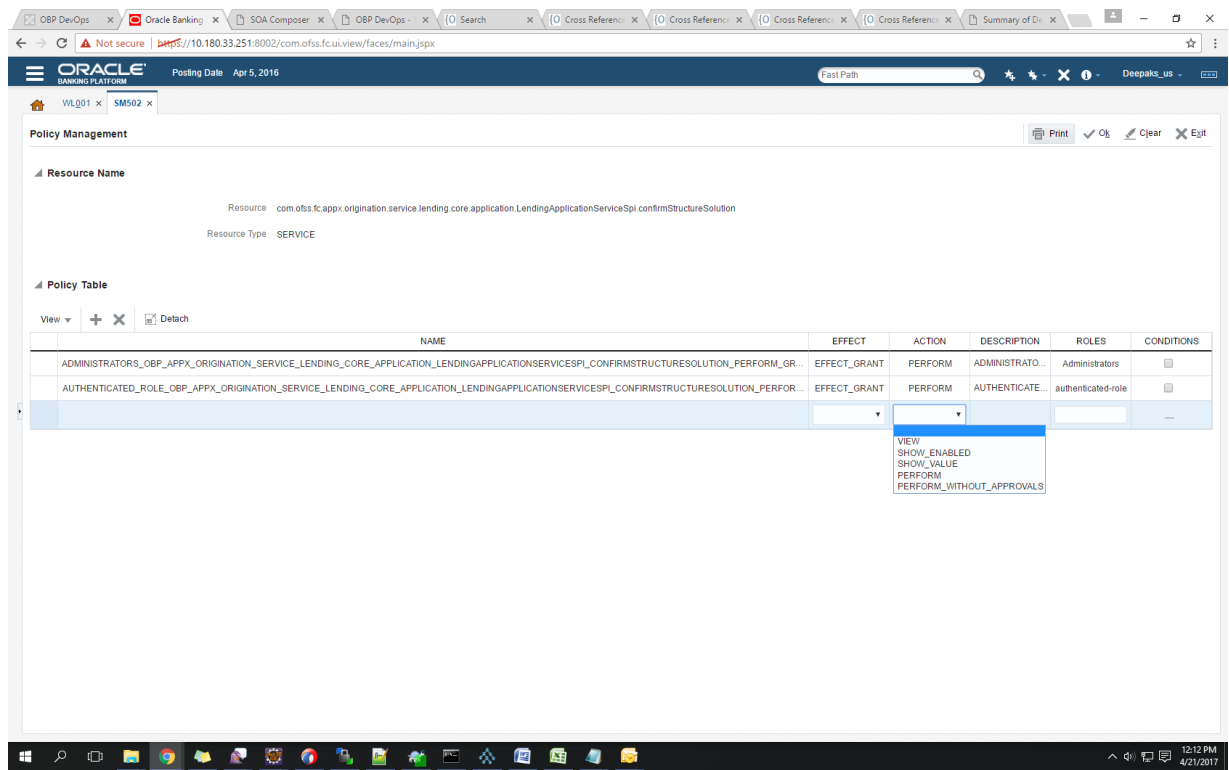
4. Click the + sign if you need to add a new policy along with the Role for which the policy is to be enabled. The rest of the fields are automatically populated.

Figure 2–31 Effect of the Policy

The screenshot shows the Oracle Policy Management interface. The 'Resource Name' section displays the resource path: `com.ots.fc.appx.OriginationService.Lending.Core.Application.LendingApplicationServiceSpi.confirmStructureSolution` with a resource type of 'SERVICE'. The 'Policy Table' section contains a table with columns: NAME, EFFECT, ACTION, DESCRIPTION, ROLES, and CONDITIONS. Two rows are visible, both with 'EFFECT_GRANT' and 'PERFORM' values. A dropdown menu is open over the 'EFFECT' column of the second row, showing options: 'EFFECT_DENY' and 'EFFECT_GRANT'.

NAME	EFFECT	ACTION	DESCRIPTION	ROLES	CONDITIONS
ADMINISTRATORS_OBP_APPX_ORIGINATION_SERVICE_LENDING_CORE_APPLICATION_LENDINGAPPLICATIONSERVICESPL_CONFIRMSTRUCTURERESOLUTION_PERFORM_GR...	EFFECT_GRANT	PERFORM	ADMINISTRATO...	Administrators	<input type="checkbox"/>
AUTHENTICATED_ROLE_OBP_APPX_ORIGINATION_SERVICE_LENDING_CORE_APPLICATION_LENDINGAPPLICATIONSERVICESPL_CONFIRMSTRUCTURERESOLUTION_PERFOR...	EFFECT_GRANT	PERFORM	AUTHENTICATE...	authenticated-role	<input type="checkbox"/>

Figure 2–32 Action of the Policy



- To add conditions, select the check box in the conditions column. This displays a collapsible section arrow in the first column. Click the arrow to display the conditions panel as shown below. Enter the conditions needed for the policy.

Figure 2–33 Conditions of the Policy

The screenshot shows the Oracle Banking Platform Policy Management interface. The page title is "Policy Management" and the resource is "com.ofss.fc.appx.OriginationService.Lending.Core.Application.LendingApplicationServiceSpi.confirmStructureSolution". The resource type is "SERVICE".

The "Policy Table" section displays a table with the following columns: NAME, EFFECT, ACTION, DESCRIPTION, ROLES, and CONDITIONS. The table contains three rows of policy conditions:

NAME	EFFECT	ACTION	DESCRIPTION	ROLES	CONDITIONS
ADMINISTRATORS_OBP_APPX_ORIGINATION_SERVICE_LENDING_CORE_APPLICATION_LENDINGAPPLICATIONSERVICESPL_CONFIRMSTRUCTURESOLUTION_PERFORM_GR...	EFFECT_GRANT	PERFORM	ADMINISTRATO...	Administrators	<input type="checkbox"/>
AUTHENTICATED_ROLE_OBP_APPX_ORIGINATION_SERVICE_LENDING_CORE_APPLICATION_LENDINGAPPLICATIONSERVICESPL_CONFIRMSTRUCTURESOLUTION_PERFOR...	EFFECT_GRANT	PERFORM	AUTHENTICATE...	authenticated-role	<input type="checkbox"/>
ADMINISTRATORS_OBP_APPX_ORIGINATION_SERVICE_LENDING_CORE_APPLICATION_LENDINGAPPLICATIONSERVICESPL_CONFIRMSTRUCTURESOLUTION_PERFORMWIT...	EFFECT_GRANT	PERFORM_WIT...	ADMINISTRATO...	Administrators	<input checked="" type="checkbox"/>

Below the table, there is a search filter: "LendingMatrix_ProductGroup" = "PRODUCT_GROUP" AND.

2.2.1.2 SOA Composer Rules Setup

For more information on setting up rules in SOA Composer, see [Section 2.1.1.2 SOA Composer Rules Setup](#).

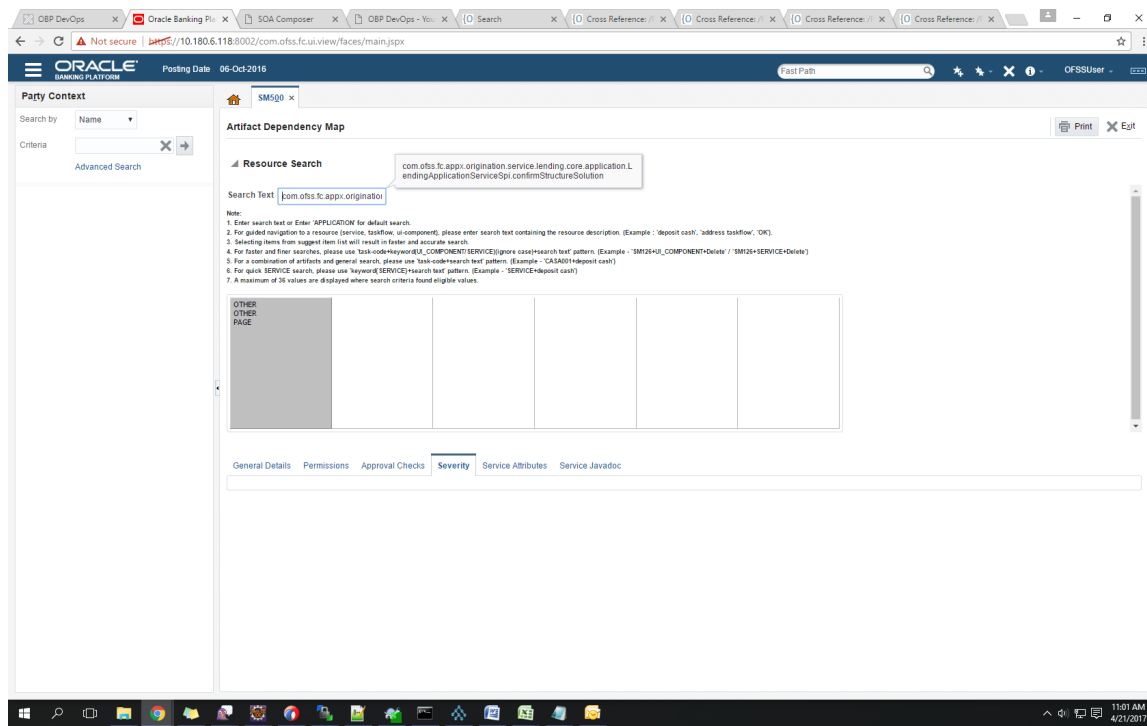
2.2.1.3 SMS Setup

The user also needs to enable matrix-based authorization for the DCA service. This can be done through the **Approval Checks** tab in **Artifact Dependency Map (Fast path: SM500)** page which is used for enabling approval.

The following procedure is performed during SMS setup:

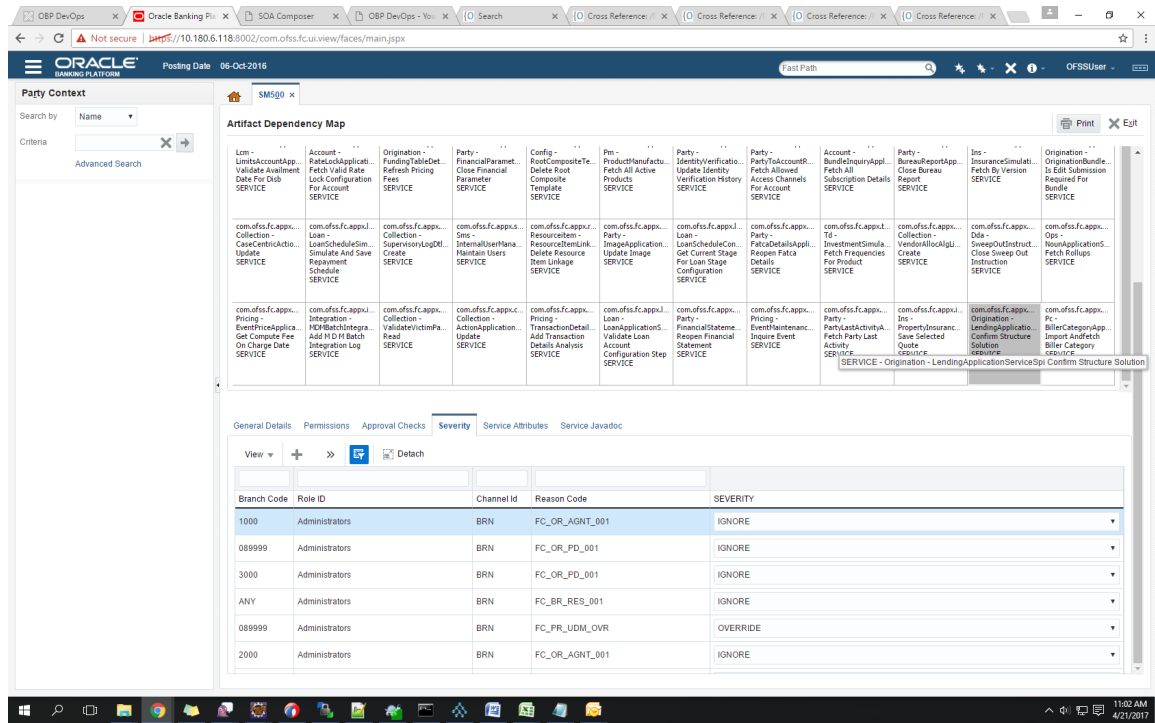
1. In the **Search Text** field, enter the service name.
com.ofss.fc.appx.OriginationService.Lending.Core.Application.LendingApplicationServiceSpi.confirmStructureSolution

Figure 2–34 Search for Service Using Entire Service Name



2. Navigate to service node by following highlighted path (in grey color) and select the service node.

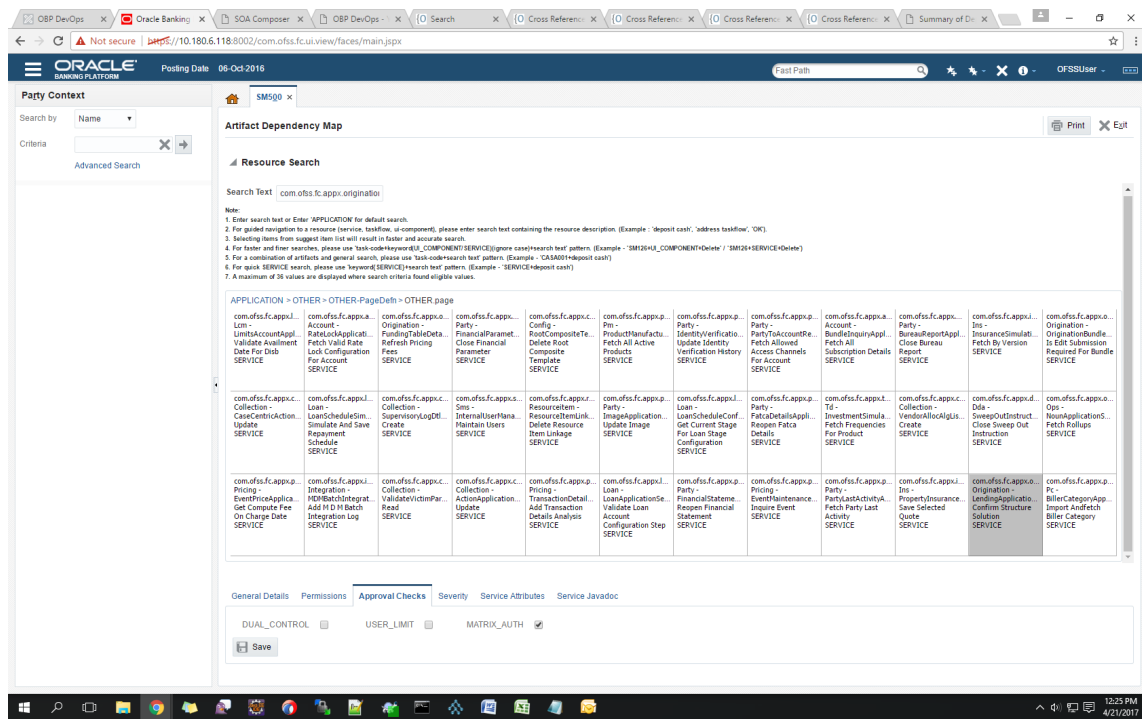
Figure 2–35 Select the Service to be Configured



3. Click the **Approval Checks** tab to maintain approval checks for the resource.

Select the **MATRIX_AUTH** checkbox and save the record.

Figure 2–36 Approval Checks tab - MATRIX_AUTH

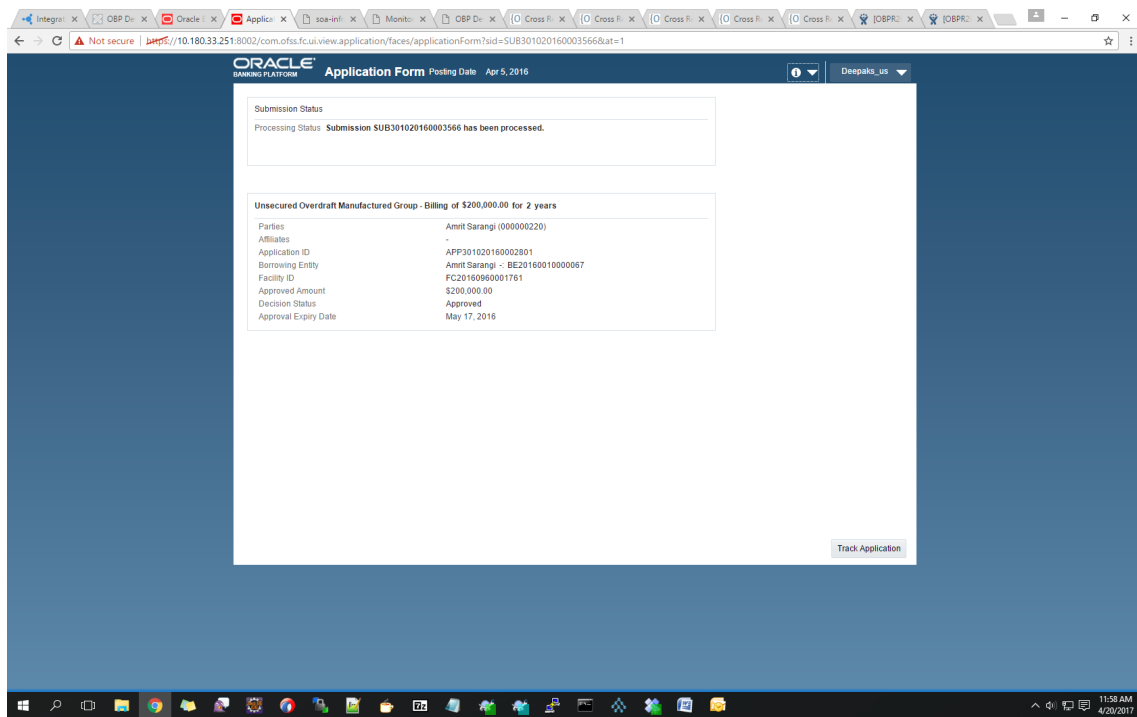


2.2.2 Performing the Transaction

Following procedure explains how to perform a transaction:

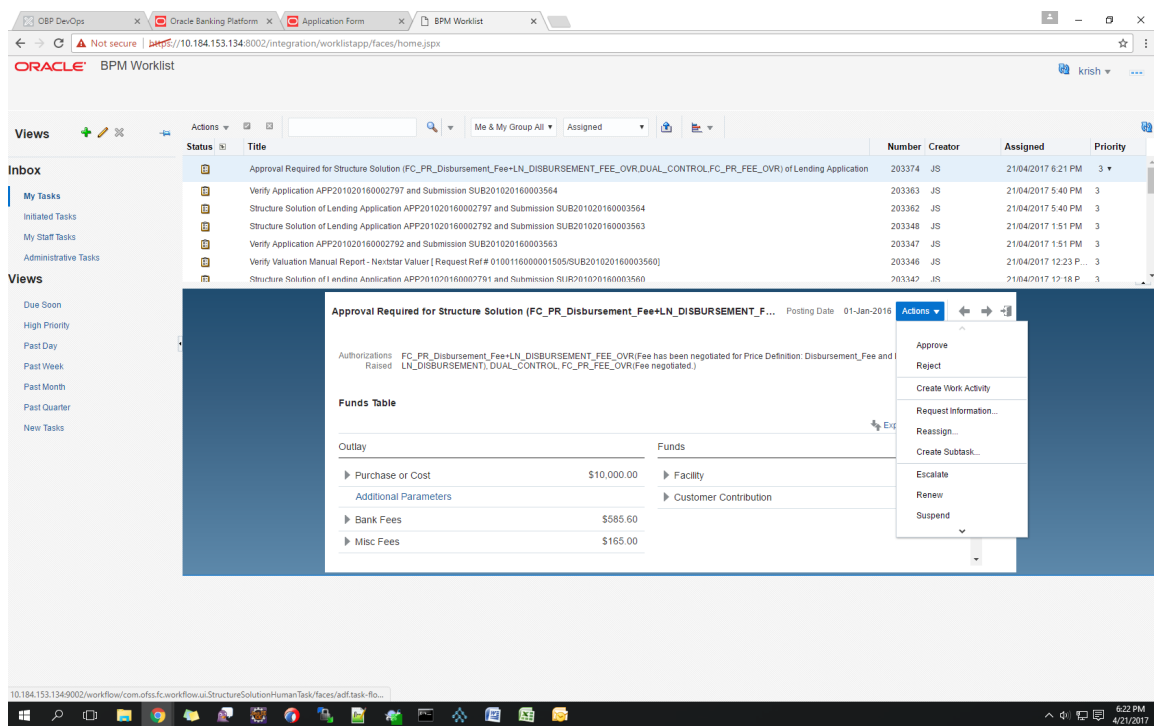
1. Log in to the application as teller user and initiate an origination task flow such that the application goes for approvals.

Figure 2–37 Initiating Origination Task Flow



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

Figure 2–38 Approving the Task



The task goes to the approver user for approval. Log in as approver user to BPM Worklist.

You can either approve or reject the work item (approved here) after adding comments for non origination flows.

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

2.3 Enabling Worklist Authorization

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

Step 1 Identify the Service Name

For example,
com.ofss.fc.appx.Origination.service.lending.core.application.LendingApplicationServiceSpi.confirmStructureSolution.

Step 2 Enable Dual Authorization or/and adding other severity

Once the services are identified, follow the below steps to enable Dual Authorization and adding other severity.

1. Log in to the application.
2. Navigate to **Artifact Dependency Map (Fast path: SM500)** page.
3. In the **Search Text** field, enter the service-name.
com.ofss.fc.appx.Origination.service.lending.core.application.LendingApplicationServiceSpi.confirmsStructureSolution

Navigate to service node by following highlighted path (in grey color) and select the service node.

Figure 2–39 Select the Service to be Configured

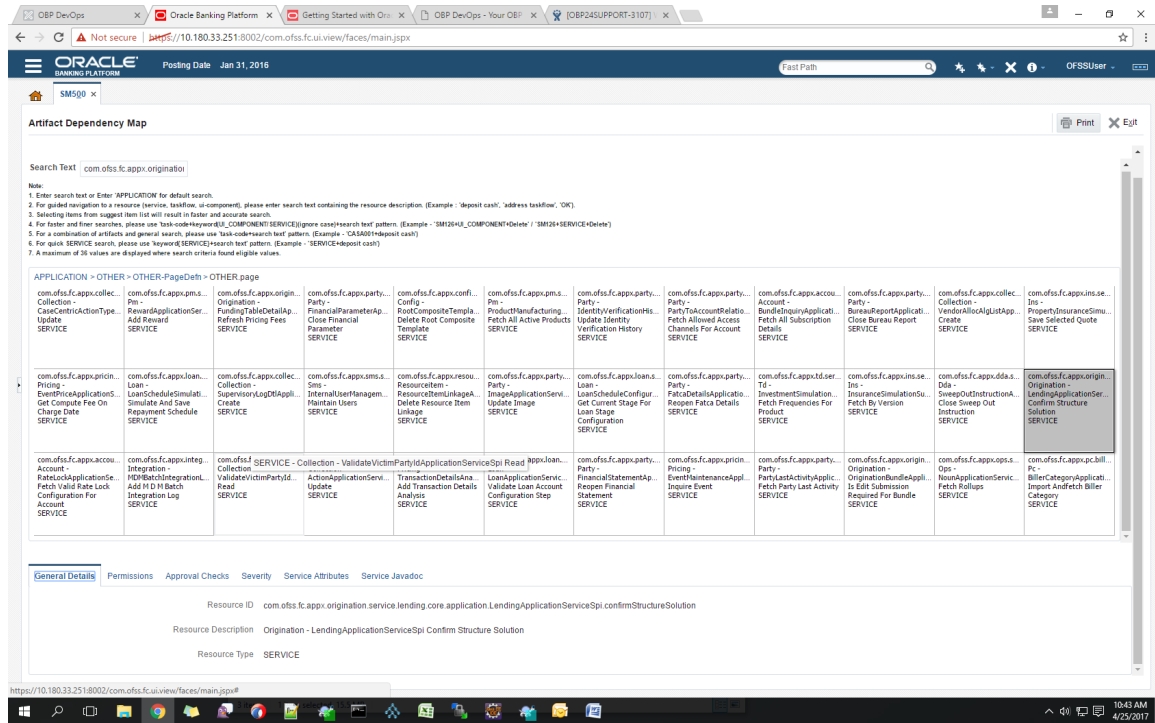
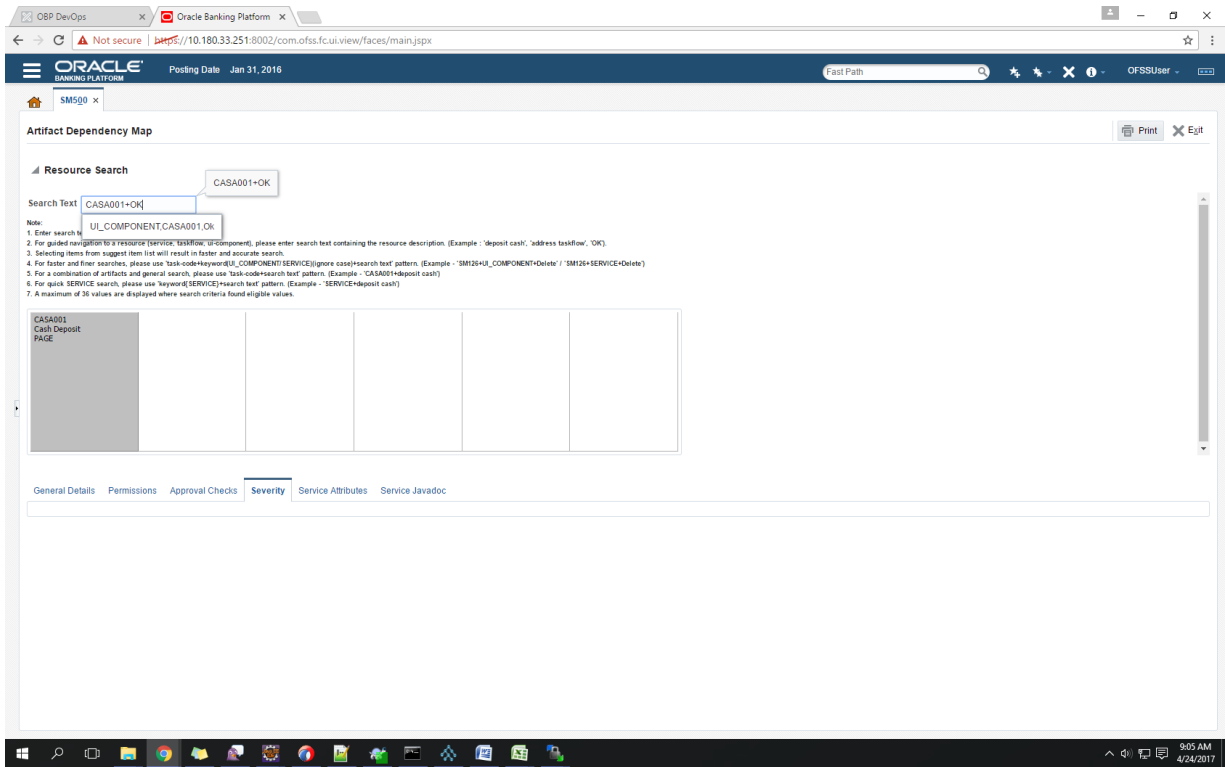


Figure 2–40 Search for Service using TASK CODE + Search text (in case of non Origination)



4. Click the **Approval Checks** tab and add approval checks.

Figure 2–41 Approval Checks tab - Add Approval Checks

The screenshot shows the Oracle Banking Platform interface. The top navigation bar includes 'Oracle Banking Platform' and 'Posting Date: 06-Oct-2016'. The main content area is titled 'Artifact Dependency Map' and 'Fast Path: SM500'. Below this, there is a 'Search Text' field containing 'com.ofss.fc.appx.Origination'. A table of services is displayed below the search field, with columns for 'APPLICATION' and 'OTHER-PageDefn'. The table lists various services such as 'com.ofss.fc.appx.Lcm', 'com.ofss.fc.appx.Account', 'com.ofss.fc.appx.RateLockApplicat', etc. At the bottom of the page, there are tabs for 'General Details', 'Permissions', 'Approval Checks', 'Severity', 'Service Attributes', and 'Service Javadoc'. The 'Approval Checks' tab is selected, and a 'Save' button is visible.

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 can be configured from the **Artifact Dependency Map (Fast path: SM500)** page.

Follow the below steps to configure severity:

1. Log in to the application.
2. Navigate to the **Artifact Dependency Map (Fast path: SM500)** page.
3. In the **Search Text** field, enter the service `com.ofss.fc.appx.Origination.service_limit.lending.core.application.LendingApplicationServiceSpi.confirmStructureSolution`.
4. Ensure approval checks are added. If not, then configure the approval checks.

2.3 Enabling Worklist Authorization

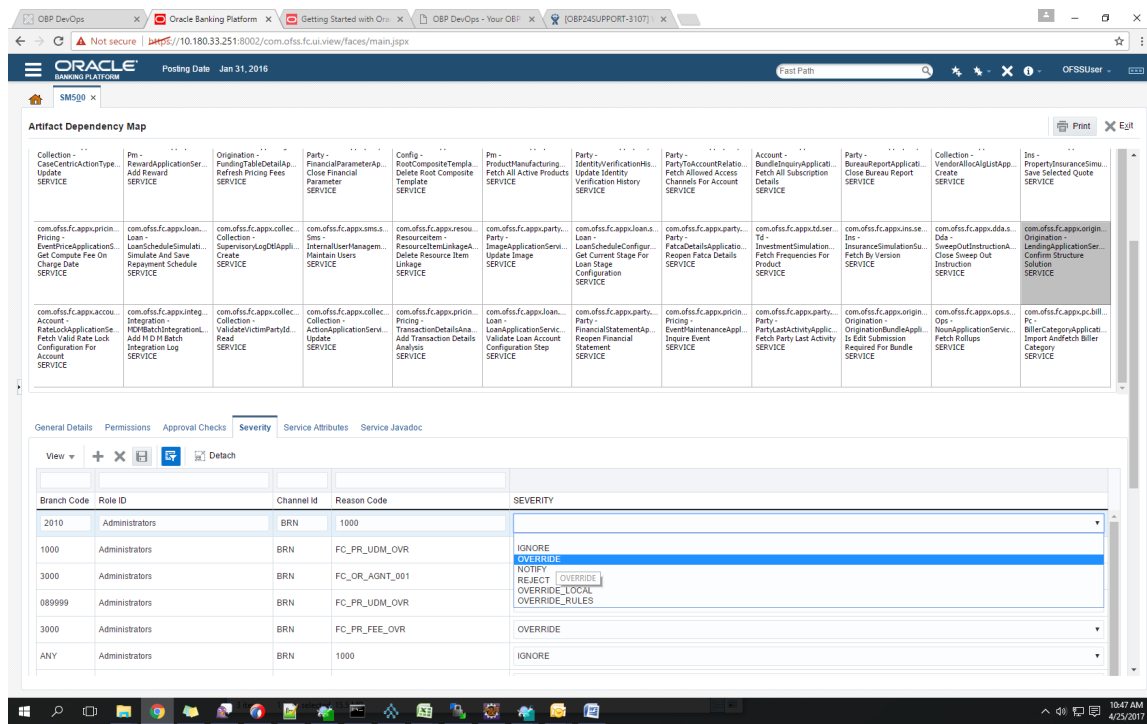
Figure 2–42 Dual Control

The screenshot shows the Oracle Banking Platform Administrator interface. The main content area is titled 'Artifact Dependency Map' and contains a 'Resource Search' section. Below the search instructions, there is a table of search results. The table has columns for service names and their dependencies. The 'Severity' tab is selected, and the 'DUAL_CONTROL' checkbox is checked. The 'Save' button is visible at the bottom of the search results area.

Service Name	Dependency	Service Name	Dependency	Service Name	Dependency	Service Name	Dependency	Service Name	Dependency	Service Name	Dependency	Service Name	Dependency	Service Name	Dependency								
com.ofss.fc.app.collec...	Collection - CaseContributionType Update SERVICE	com.ofss.fc.app.prm.se...	Prm - RewardsApplicationServ. Add Reward SERVICE	com.ofss.fc.app.origina...	Origination - FundingTableDetailApp. Refresh Pricing Fees SERVICE	com.ofss.fc.app.party...	Party - FinancialParameterApp. Close Financial Parameter SERVICE	com.ofss.fc.app.config...	Config - RootCompositeTempla. Delete Root Composite Template SERVICE	com.ofss.fc.app.prm.se...	Prm - ProductManufacturingA. Fetch All Active Products SERVICE	com.ofss.fc.app.party...	Party - IdentityVerificationHist. Update Identity Verification History SERVICE	com.ofss.fc.app.party...	Party - PartyToAccountRelatio. Fetch Allowed Access Channels For Account Details SERVICE	com.ofss.fc.app.prm.se...	Prm - BundleInquiryApplicati. Fetch All Subscription Details SERVICE	com.ofss.fc.app.party...	Party - BureauReportApplicati. Close Bureau Report SERVICE	com.ofss.fc.app.collec...	Collection - VendorAllocationApp. Create SERVICE	com.ofss.fc.app.ins.se...	Ins - ProperInsuranceSimu. Save Selected Quote SERVICE
com.ofss.fc.app.pricing...	Pricing - EventPriceApplicationS. Get Complete Fee On Charge Date SERVICE	com.ofss.fc.app.loans...	Loans - LoanScheduleSimulatio. Simulate And Save Requirement Schedule SERVICE	com.ofss.fc.app.collec...	Collection - SupervisorLogDetailApp. Create SERVICE	com.ofss.fc.app.sims...	Sims - InternalUserManagement. Maintain Users SERVICE	com.ofss.fc.app.resou...	ResourceItem - ResourceItemLinkageA. Delete Resource Item Linkage SERVICE	com.ofss.fc.app.party...	Party - ImageApplicationServ. Update Image SERVICE	com.ofss.fc.app.loans...	Loans - LoanScheduleConfigur. Get Current Stage For Loan Stage Configuration SERVICE	com.ofss.fc.app.party...	Party - PartyFatcaDetailsApplicati. Fetch Fatca Details SERVICE	com.ofss.fc.app.prm.se...	Prm - InvestmentSimulationA. Fetch Frequencies For Product SERVICE	com.ofss.fc.app.ins.se...	Ins - InsuranceSimulationSu. Fetch By Version SERVICE	com.ofss.fc.app.dda.s...	DDA - SweepOutInstructionA. Close Sweep Out Instruction SERVICE	com.ofss.fc.app.origina...	Origination - LendingApplicationServ. Confirm Structure Solution SERVICE
com.ofss.fc.app.account...	Account - RateLockApplicationSe. Fetch Valid Rate Lock Configuration For Account SERVICE	com.ofss.fc.app.integr...	Integration - MBOBatchIntegrationL. Add M D H Batch Integration Log SERVICE	com.ofss.fc.app.collec...	Collection - ValidEventItemPartyId. Read SERVICE	com.ofss.fc.app.collec...	Collection - ActionApplicationServ. Update SERVICE	com.ofss.fc.app.pricing...	Pricing - TransactionDetailsAnal. Add Transaction Details Analysis SERVICE	com.ofss.fc.app.loans...	Loans - LoanApplicationService. Validate Loan Account Configuration Step SERVICE	com.ofss.fc.app.party...	Party - FinancialStatementApp. Prepare Financial Statement SERVICE	com.ofss.fc.app.pricing...	Pricing - EventMaintenanceApplic. Inquire Event SERVICE	com.ofss.fc.app.party...	Party - PartyLastActivityApplic. Fetch Party Last Activity SERVICE	com.ofss.fc.app.origina...	Origination - OriginationBundleApplic. Is Edit Submission Required For Bundle SERVICE	com.ofss.fc.app.ops.s...	Ops - NonApplicationServic. Fetch Rollups SERVICE	com.ofss.fc.app.pc.bill...	PC - BillCategoryApplicati. Import AndFetch Biller Category SERVICE

5. Click the **Severity** tab and update the severity, if already maintained. Else, click **Add** button in the toolbar to add new row in the table.

Figure 2-43 Add New Severity



6. Enter the following details in the **Severity** tab.

Branch Code	Branch code from which the transaction is to be performed. Specify ANY to configure for all branches. Example: 082991
Role ID	Security Role to which the user belongs and initiates the transaction.
Channel ID	Channels such as BRN, ATM, IB and so on, through which the transaction is performed.
Reason Code	Select REASON CODE 1000 normal approval flow.
Severity	This field contains four values as detailed below: <ul style="list-style-type: none"> ■ 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 does not allow to proceed with transaction.

2.3 Enabling Worklist Authorization

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

Note

If the Severity Configuration is already set up do not change it.

7. Save the newly added severity using the **Save**.

Figure 2–44 Save Severity Details

The screenshot shows the Oracle Banking Platform Administrator interface. The top navigation bar includes 'ORACLE BANKING PLATFORM' and 'Posting Date: Jan 31, 2016'. The main content area displays an 'Artifact Dependency Map' with a grid of service dependencies. An 'Information' dialog box is open in the center, showing a success message: 'Operation completed successfully. Transaction reference number: 2016015004583021'. Below the map, the 'Severity' configuration is visible, showing a table with columns for Branch Code, Role ID, Channel Id, Reason Code, and SEVERITY.

Branch Code	Role ID	Channel Id	Reason Code	SEVERITY
2010	Administrators	BRN	1000	VERRIDE
1000	Administrators	BRN	FC_PR_UDM_OVR	VERRIDE
3000	Administrators	BRN	FC_OR_AGNT_001	IGNORE
089999	Administrators	BRN	FC_PR_UDM_OVR	VERRIDE
3000	Administrators	BRN	FC_PR_FEE_OVR	VERRIDE

Figure 2–45 Update Severity and Save

The screenshot shows the Oracle Banking Platform interface. The top navigation bar includes the Oracle logo, the text 'Posting Date Jan 31, 2016', and a search bar. Below the navigation bar is the 'Artifact Dependency Map' section, which displays a grid of service dependencies. The 'Severity' tab is selected, showing a table with the following data:

Branch Code	Role ID	Channel Id	Reason Code	SEVERITY
2010	Administrators	BRN	1000	IGNORE
1000	Administrators	BRN	FC_PR_UDM_OVR	OVERRIDE
3000	Administrators	BRN	FC_OR_AGNT_001	IGNORE
089999	Administrators	BRN	FC_PR_UDM_OVR	OVERRIDE
3000	Administrators	BRN	FC_FR_FEE_OVR	OVERRIDE

Step 4 Testing Approvals

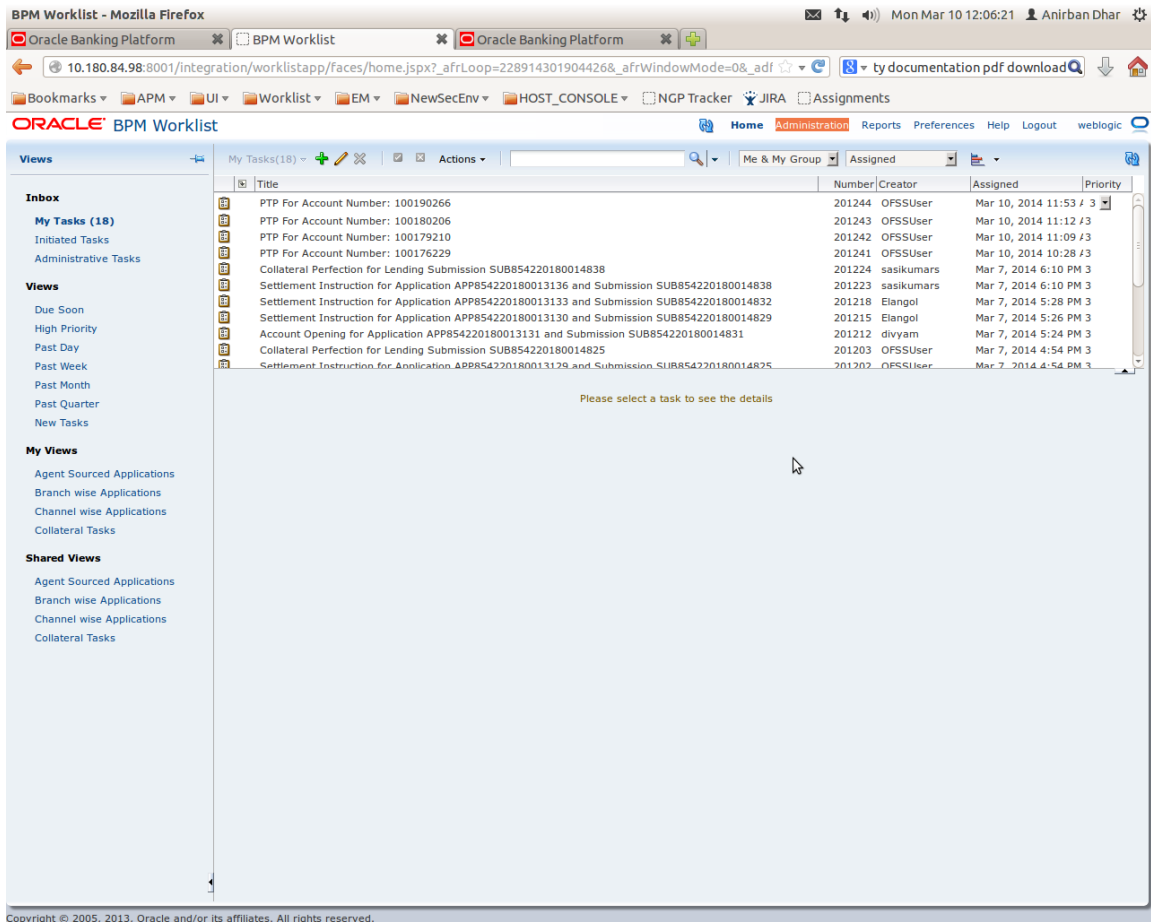
Once the configurations are done, navigate to the screen for which approval is enabled. For more information on testing approvals in case of Lending Confirm Structure Solution, see [Section 2.2.2 Performing the Transaction](#)

2.4 BPM Routing Rules Setup

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

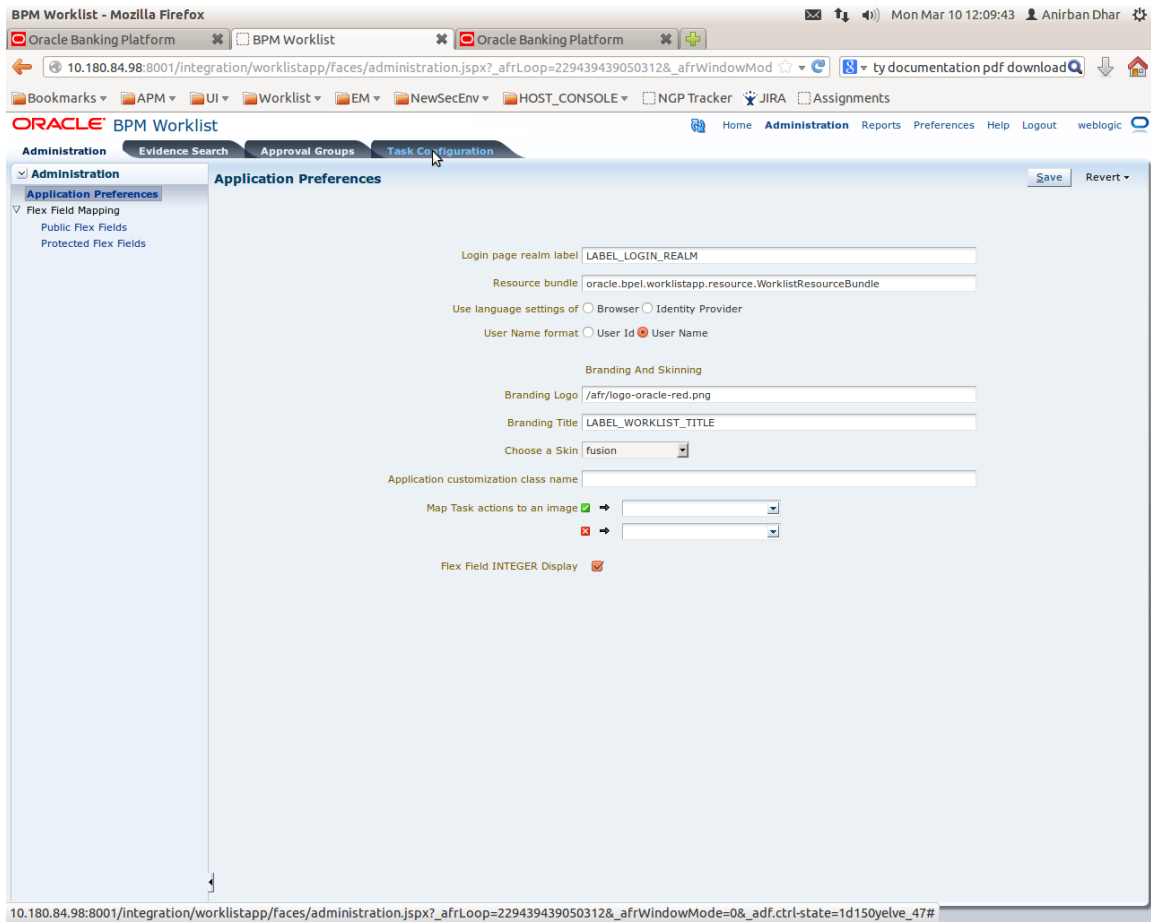
1. Log in to Worklist App using Weblogic / Weblogic1 [or any other user with administrators as role].
2. Click **Administration** link.

Figure 2–46 Worklist App - Administration Link



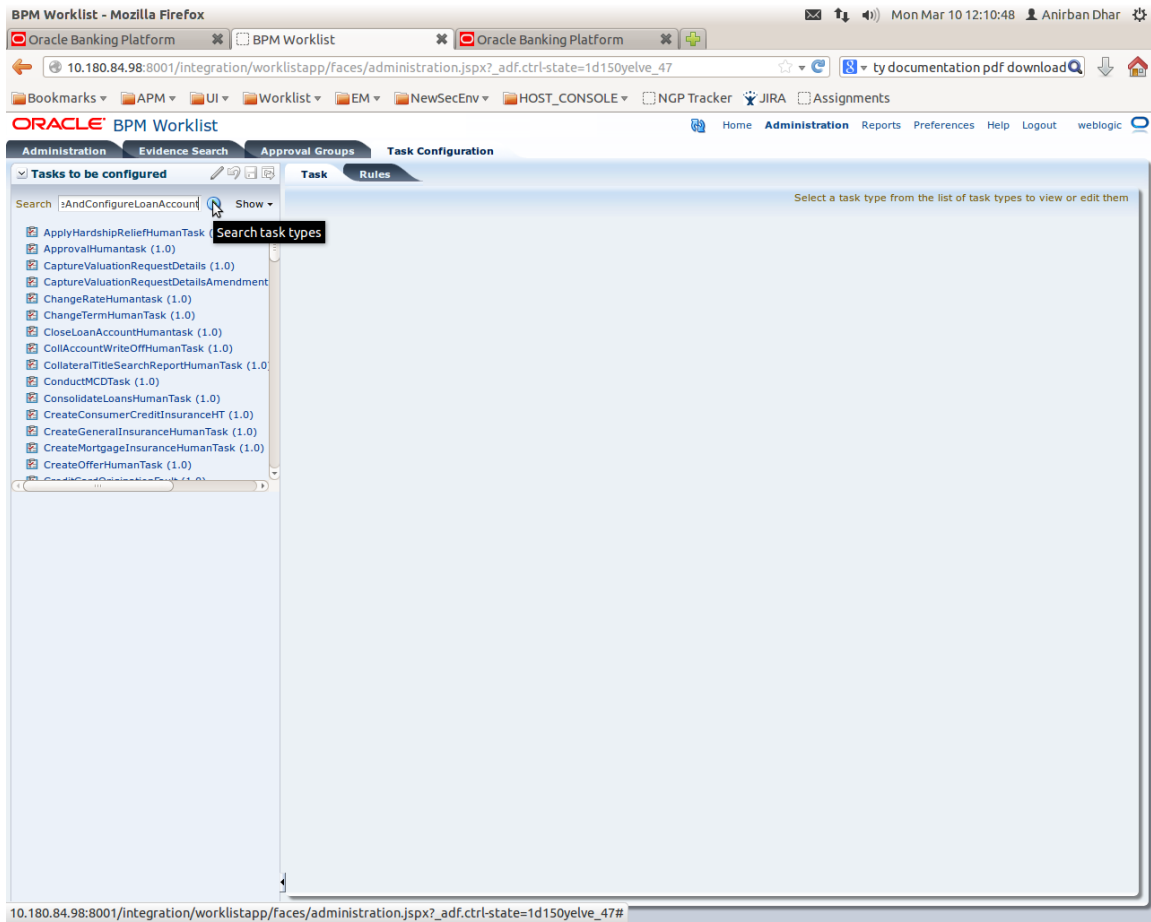
3. Navigate to **Task Configuration** tab.

Figure 2–47 Task Configuration



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

Figure 2–48 Search Task Types



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

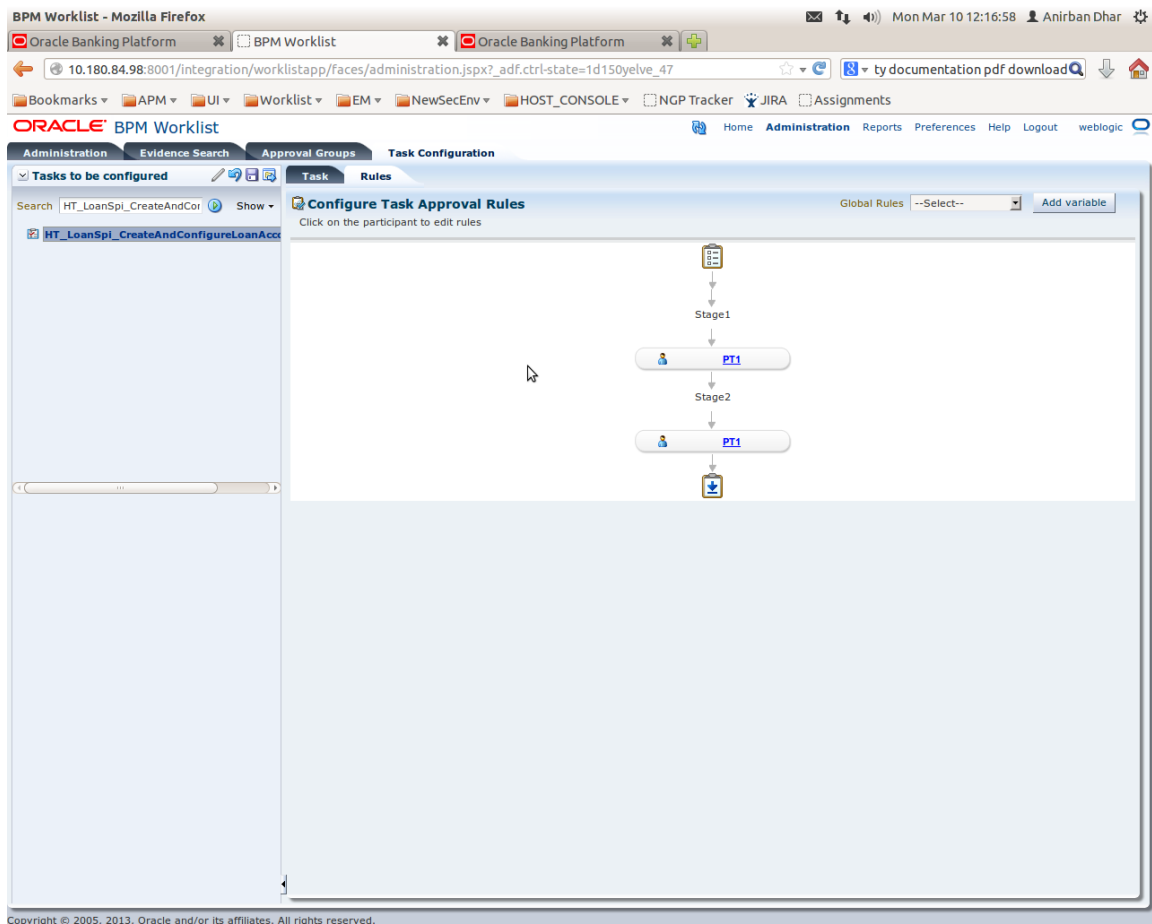
Figure 2–49 Task Editing

The screenshot shows the Oracle BPM Worklist administration interface. The browser window title is 'BPM Worklist - Mozilla Firefox'. The address bar shows the URL: '10.180.84.98:8001/integration/worklistapp/faces/administration.jspx?_adf.ctrl-state=1d150yelve_47'. The page title is 'ORACLE BPM Worklist'. The navigation menu includes 'Administration', 'Evidence Search', 'Approval Groups', and 'Task Configuration'. The 'Task Configuration' page is active, showing the configuration for the task 'HT_LoanSpi_CreateAndConfigureLoanAccount : Event Driven Configuration'. The configuration includes sections for 'Task Aggregation', 'Assignment and Routing Policy', 'Expiration and Escalation Policy', and 'Notification Settings'. The 'Notification Settings' section contains a table with columns for 'Task Status', 'Recipient', and 'Notification Header'. The table has three rows: 'Assign' with 'Assignees' as the recipient, 'Complete' with 'Initiator' as the recipient, and 'Error' with 'Owner' as the recipient. There are edit icons in the 'Notification Header' column for each row. Below the table, there are checkboxes for 'Make notifications secure (exclude details)', 'Make notification actionable', 'Send task attachments with email notifications', and 'Don't send multiple notifications for the same human task event'. The page also shows 'No reminders' and a 'Task Access' section.

Copyright © 2005. 2013. Oracle and/or its affiliates. All rights reserved.

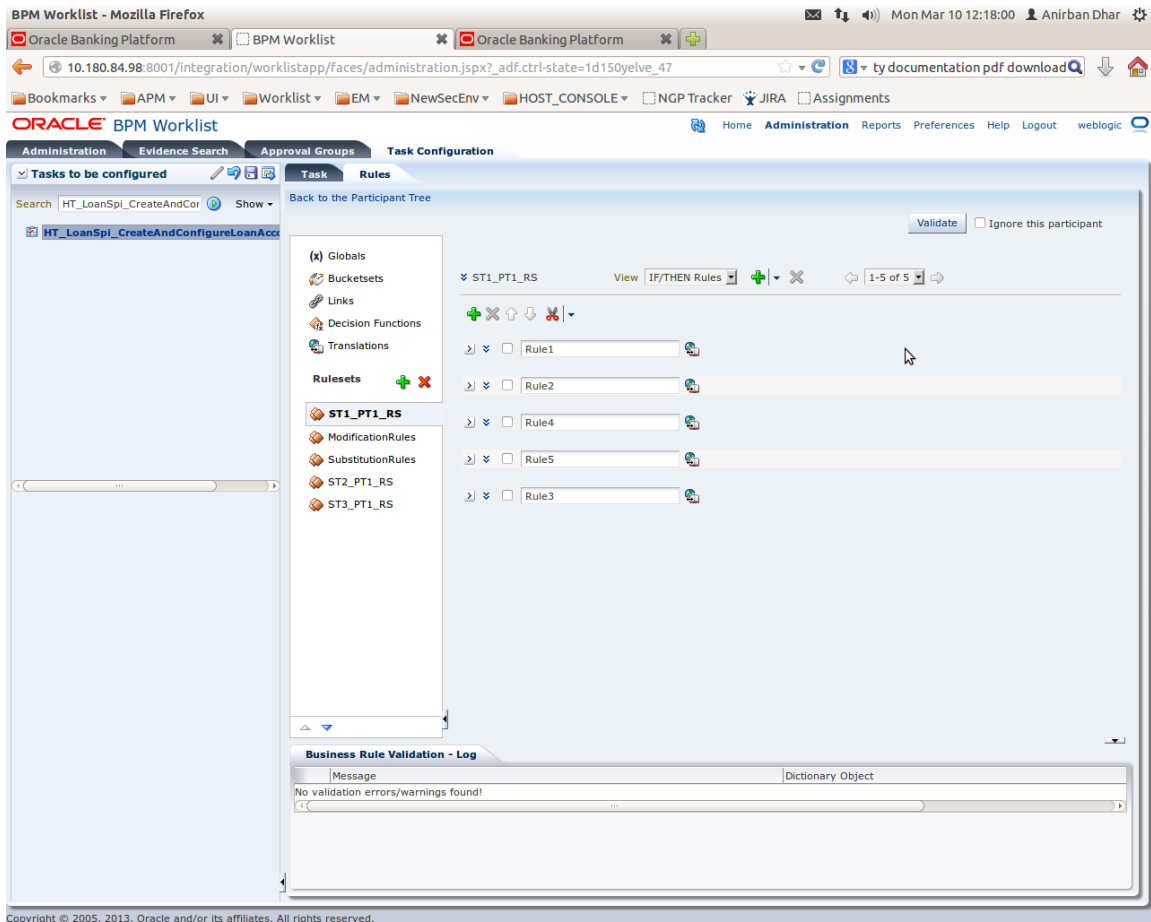
6. Click the **Edit** icon.

Figure 2–50 Configuring Rules - Edit Icon



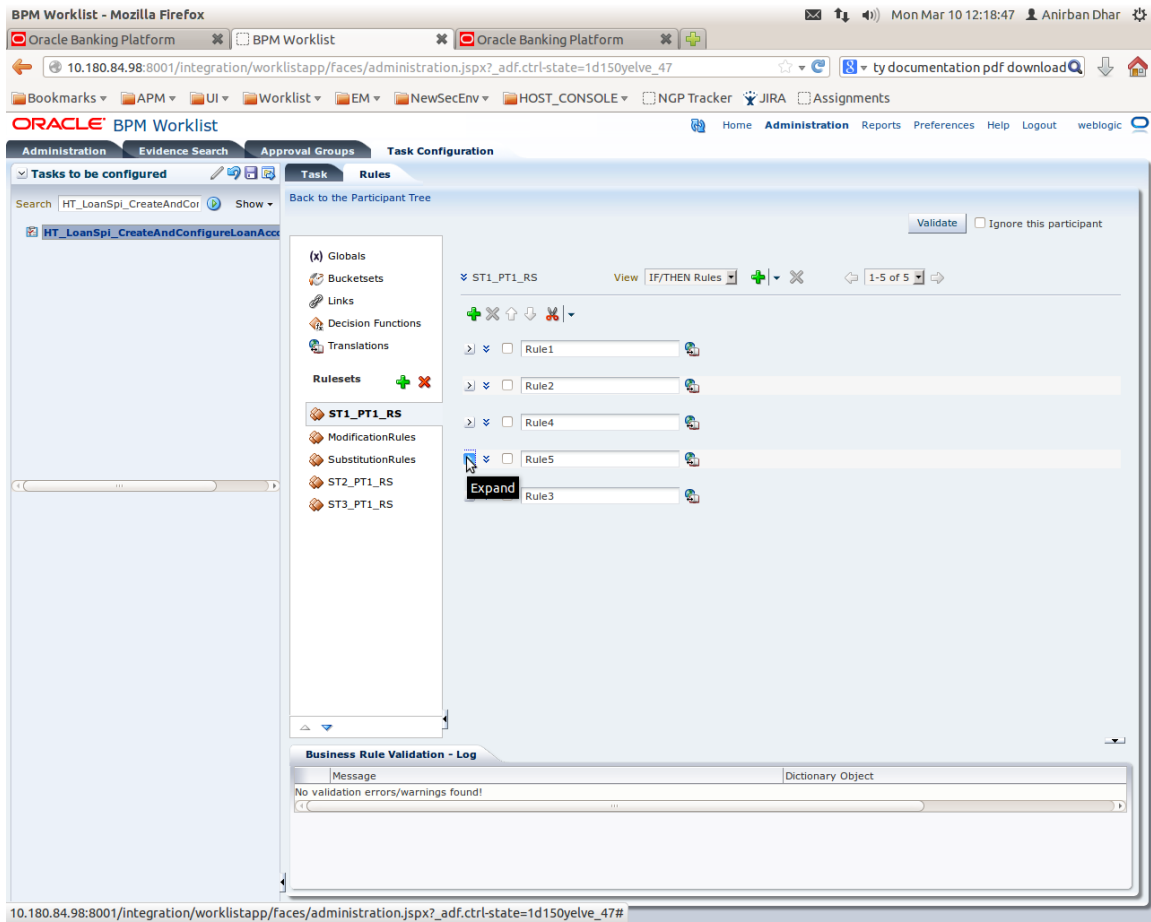
7. Navigate to **Rules** tab for configuring rules.
8. Click participant to edit rule for that stage. For example, stage 1 -> PT1)

Figure 2–51 Configuring Rules - Rules tab



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

Figure 2–52 Expand Rule



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

Figure 2–53 Advanced Settings

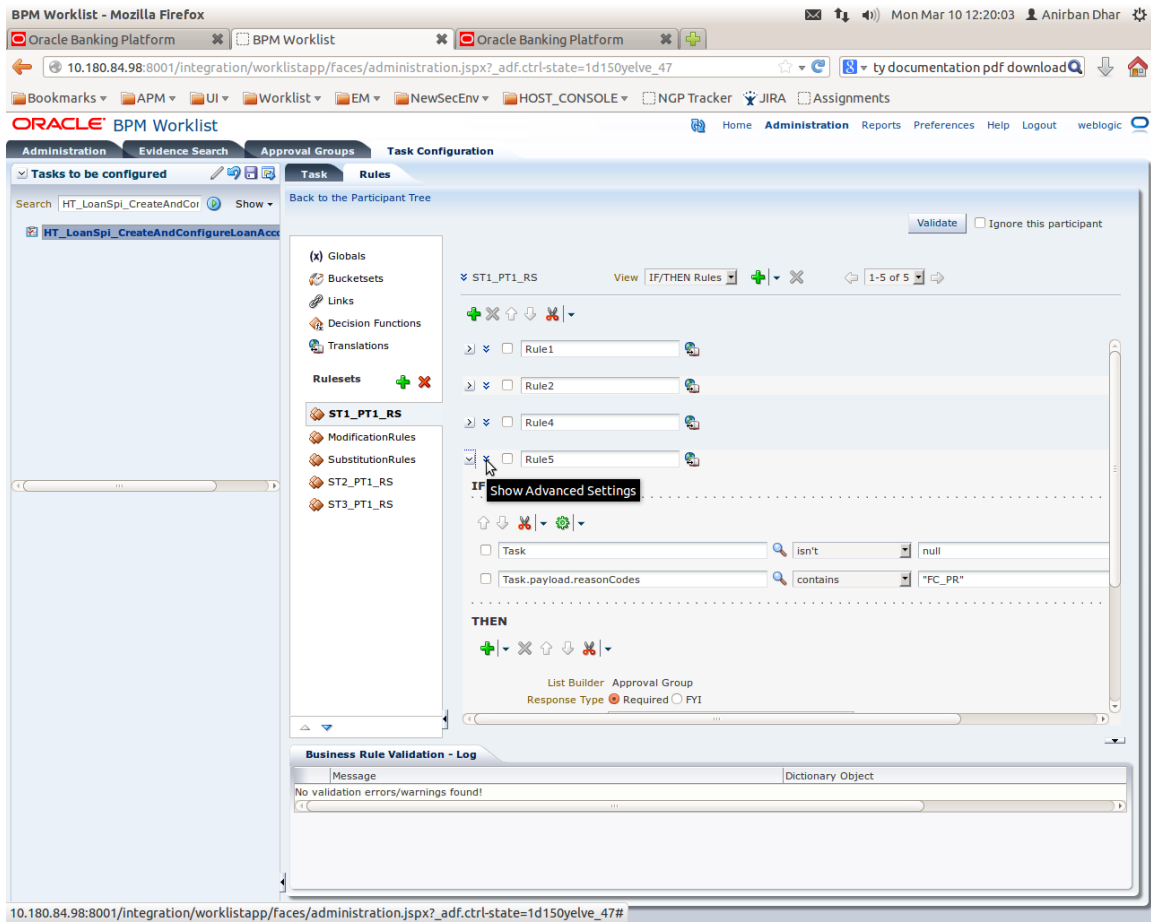
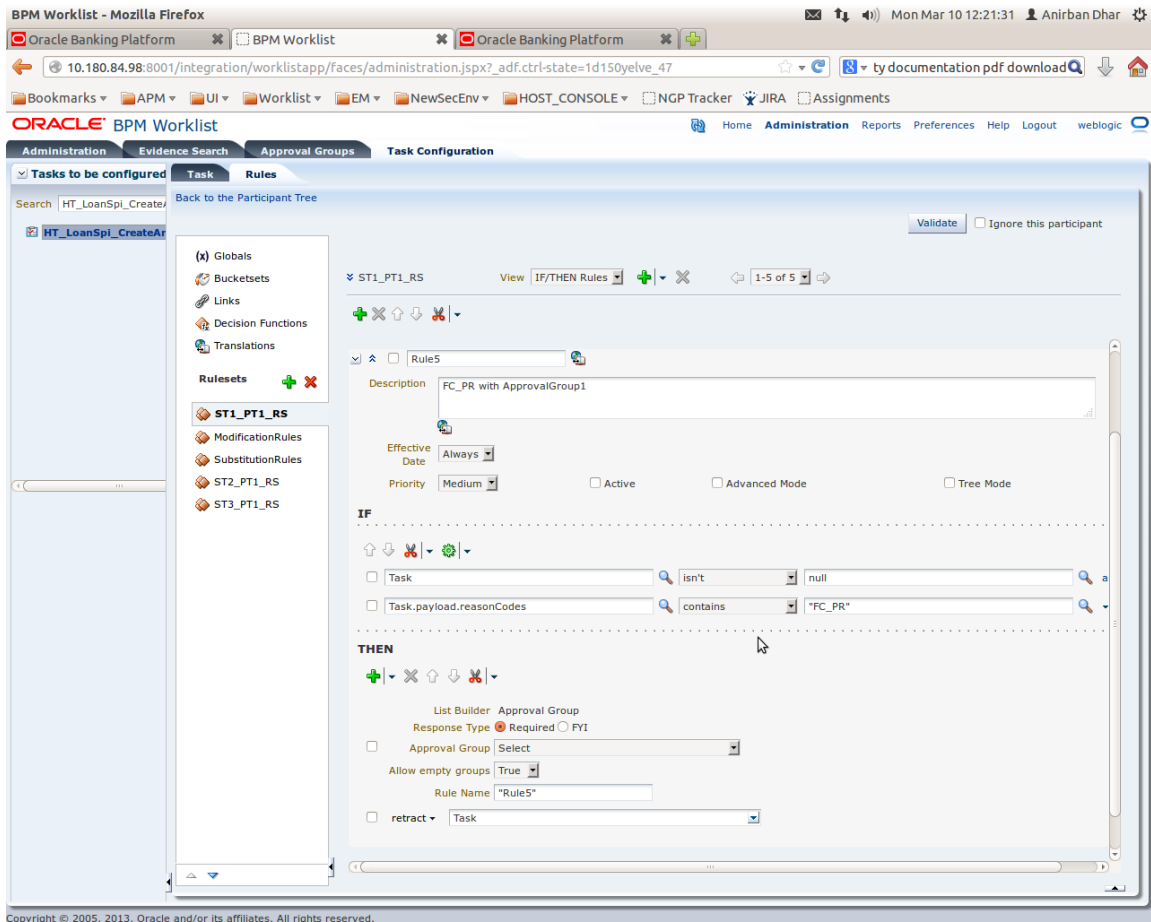
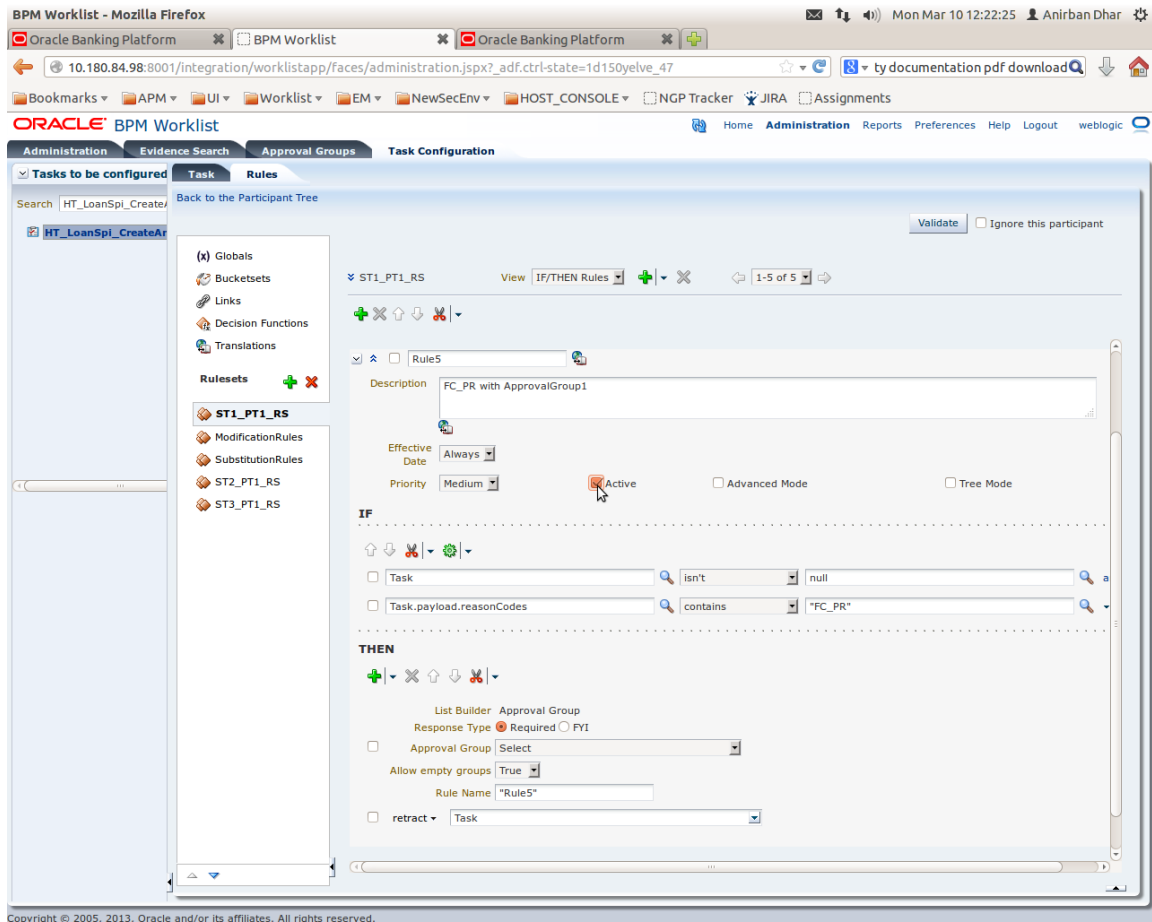


Figure 2–54 Rule Expansion



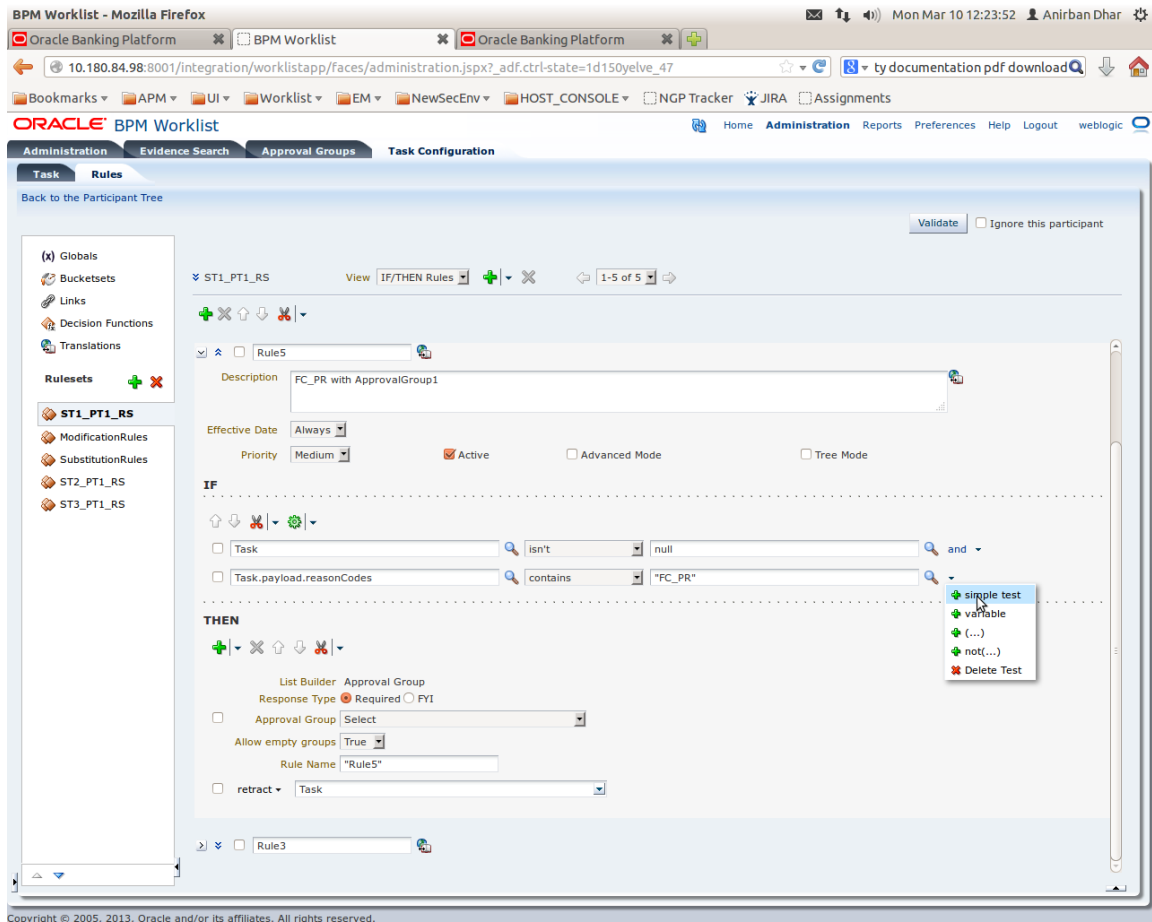
11. Click **Active** check box to enable a particular rule.

Figure 2–55 Enabling Rule



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

Figure 2–56 Adding New Row of Condition



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

Figure 2–57 Expanding Payload

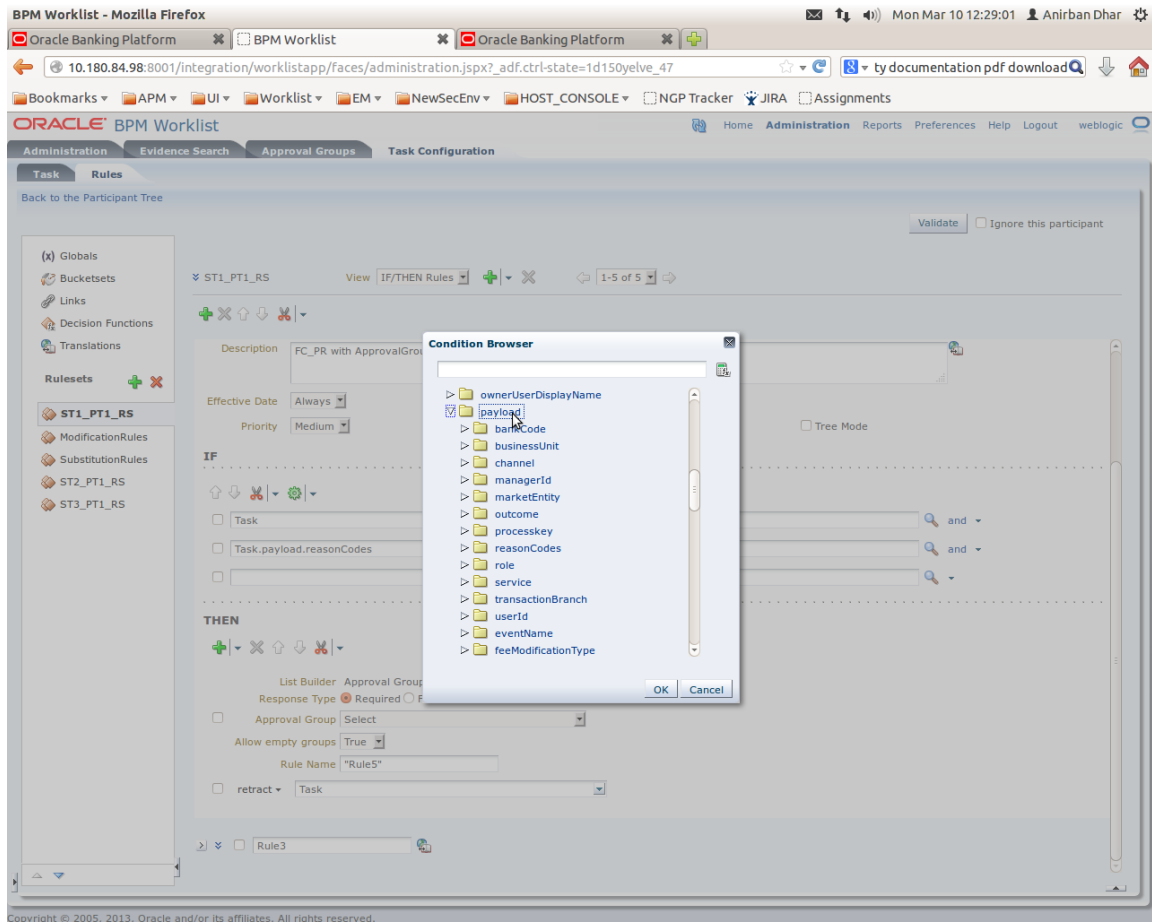
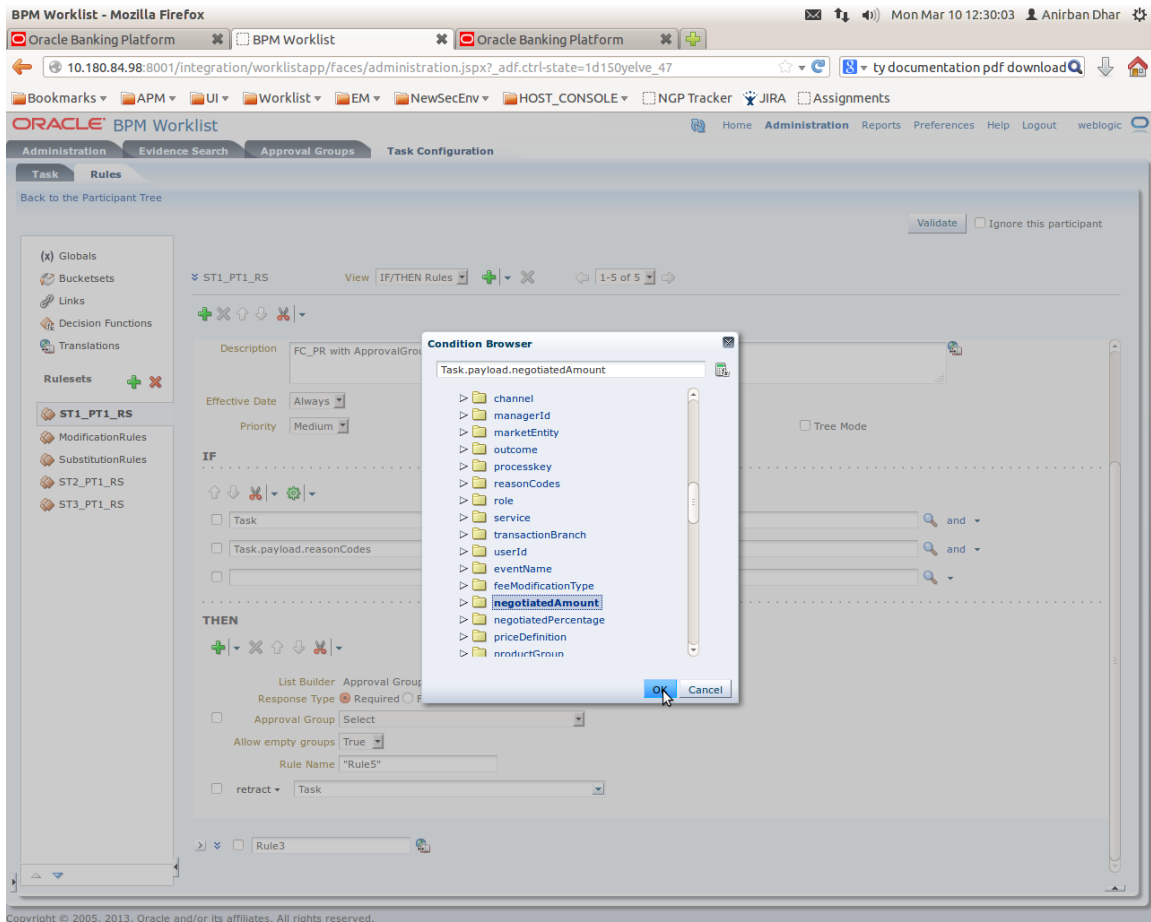
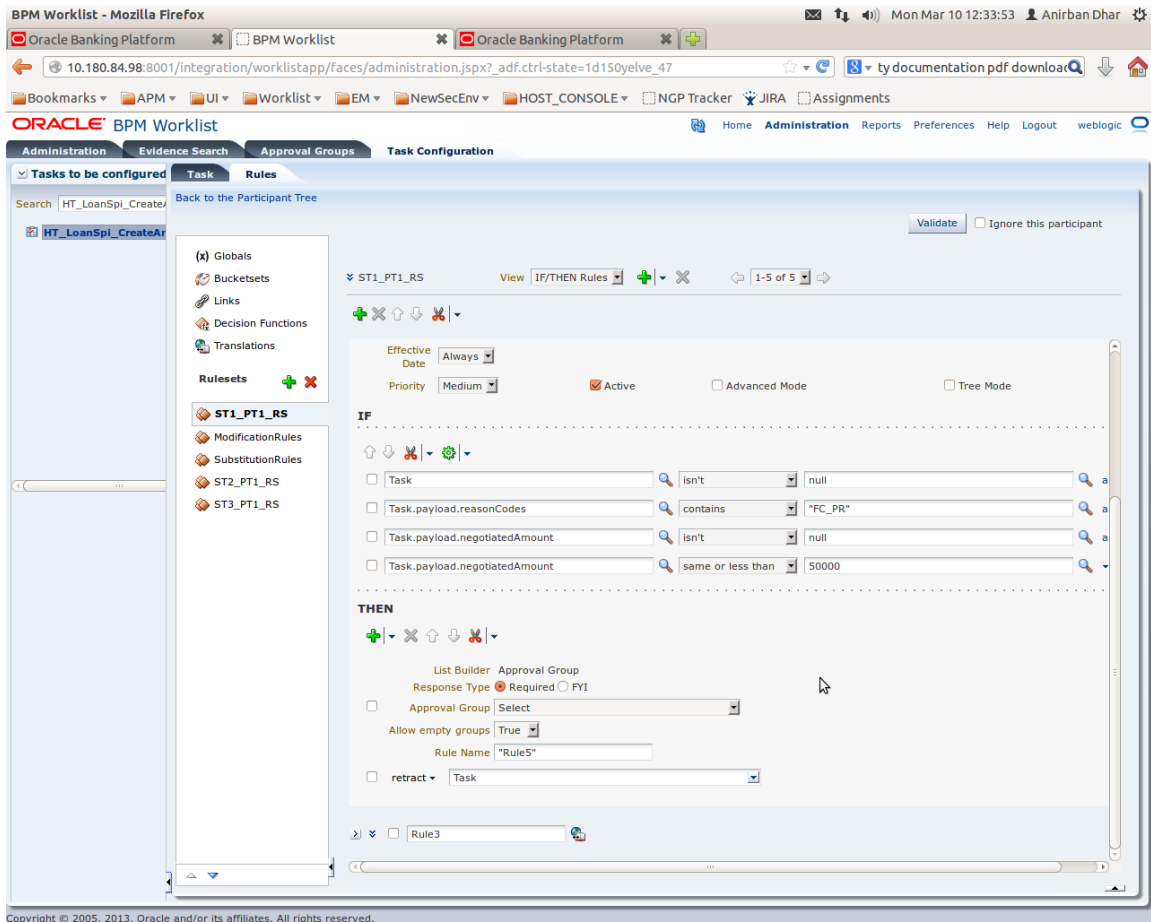


Figure 2–58 Selecting Fact



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

Figure 2–59 Updating Fact Value

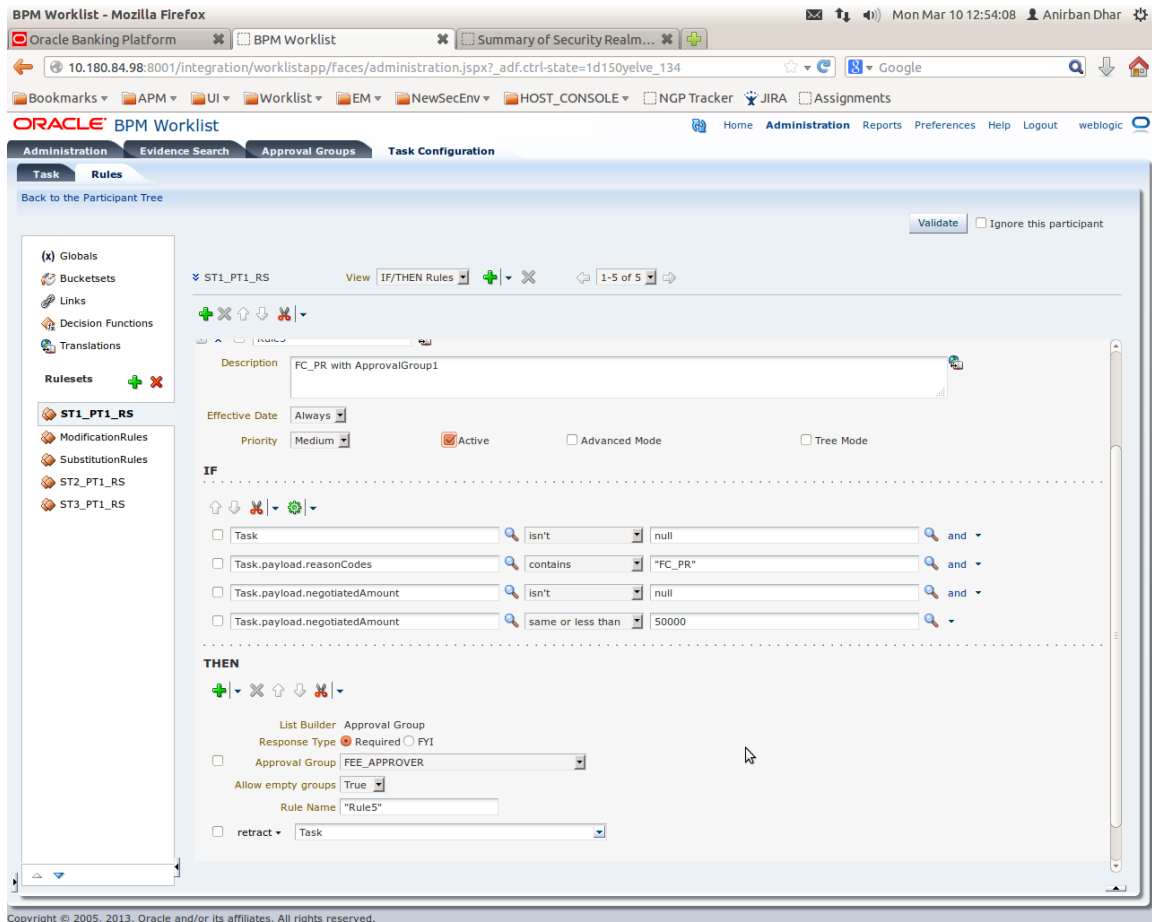


Note

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 2–60 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 Defining Task Configuration Rules

Each human task in Origination business process has business rule associated with it. This business rule can be used to set various parameters for the task like SLA period, Assignees, task priority, task owner, STP configuration, and so on. Below sections illustrate the steps to configure such business rules.

3.1 Important Rule Artifacts

This section provides information about important rule artifacts.

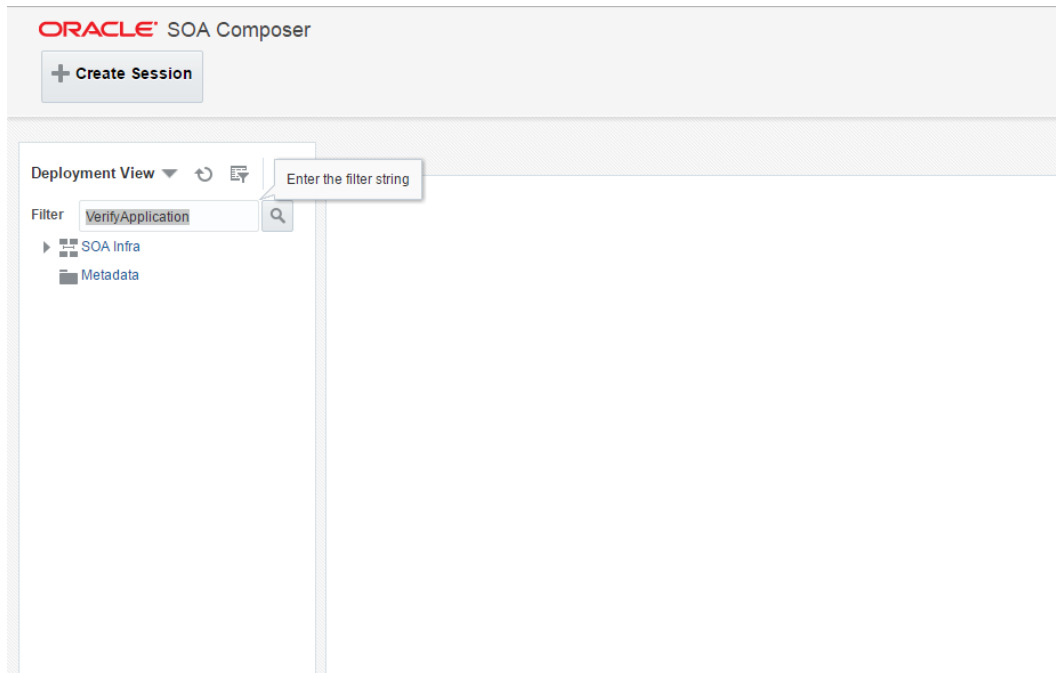
3.1.1 Rules Dictionary

For each human task, a `<name>TaskConfigRules.rules` file is provided. For example, `VerifyApplicationTaskConfigRules.rules`. These rules dictionary files have to be used to configure attributes of the respective human task.

Rules dictionaries can be viewed and edited using Oracle SOA Composer. Oracle SOA composer can be accessed using <http://<IP-of-SOA-server>:8001/soa/composer>.

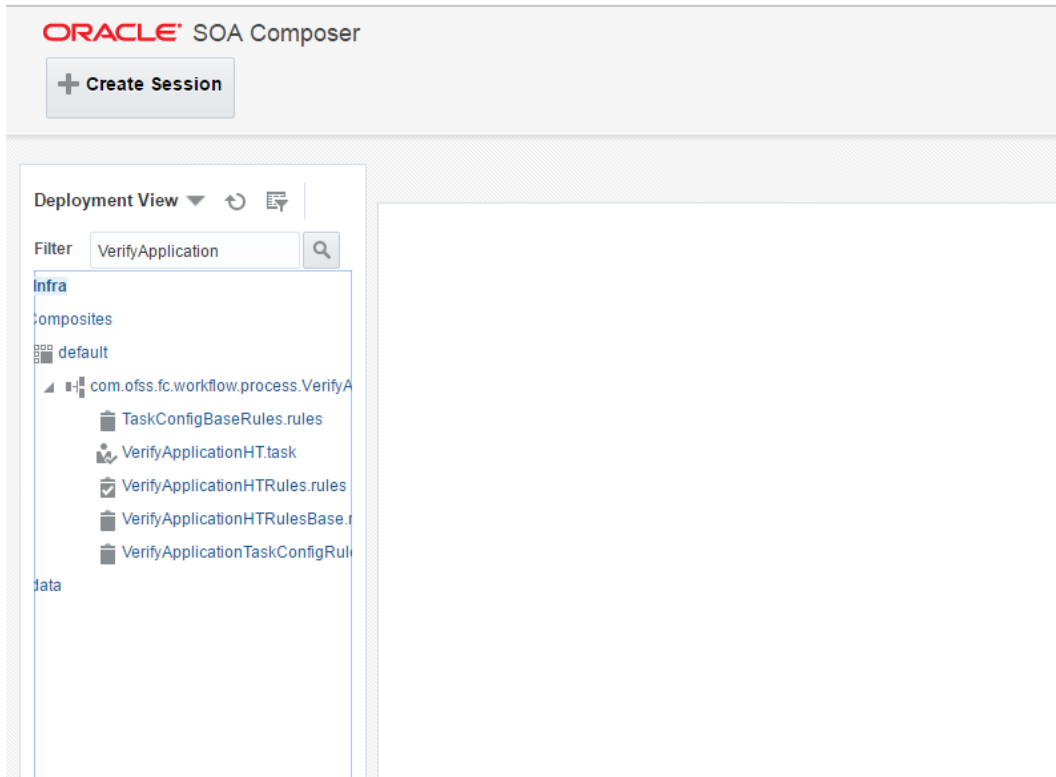
Figure 3–1 and Figure 3–2 illustrate the procedure of opening the rules dictionary for VerfiyApplicationTask.

Figure 3–1 SOA Composer - Open Rules Dictionary Browser



3.1 Important Rule Artifacts

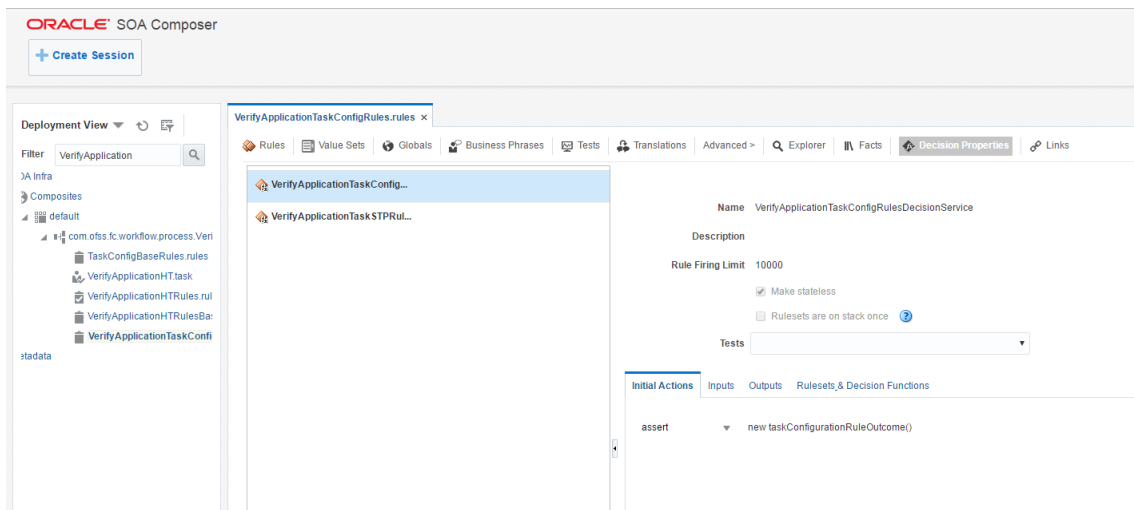
Figure 3–2 SOA Composer – Selecting Rules Dictionary



3.1.2 Decision Function

Configuration rules are written in Rulesets, which are executed through a Decision Function. In each rules dictionary a decision function is provided by the name of <name>TaskConfigurationRulesDecisionService.

Figure 3–3 SOA Composer – Selecting Decision Function



3.1.3 Rulesets

Each decision function executes one or more rulesets. This is where the rules are written. Any rulesets that are defined in the rules dictionary can be added to be used in a decision function.

Figure 3–4 illustrates addition or removal of rulesets from a decision function.

Figure 3–4 SOA Composer - Adding Rulesets to Decision Function

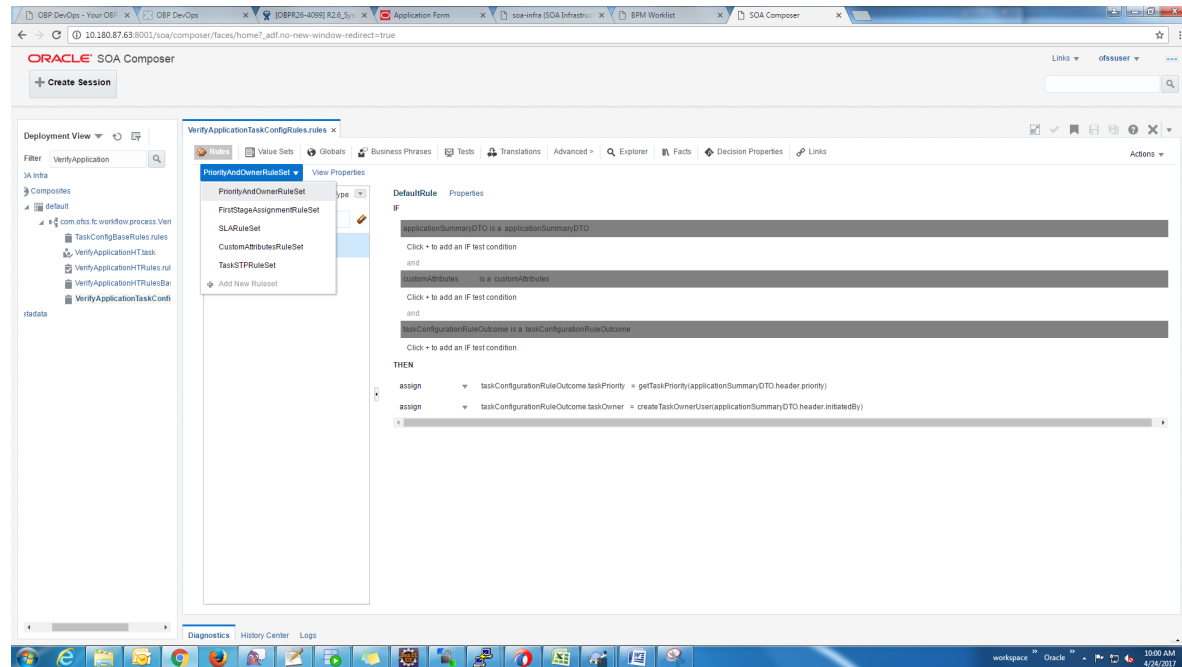
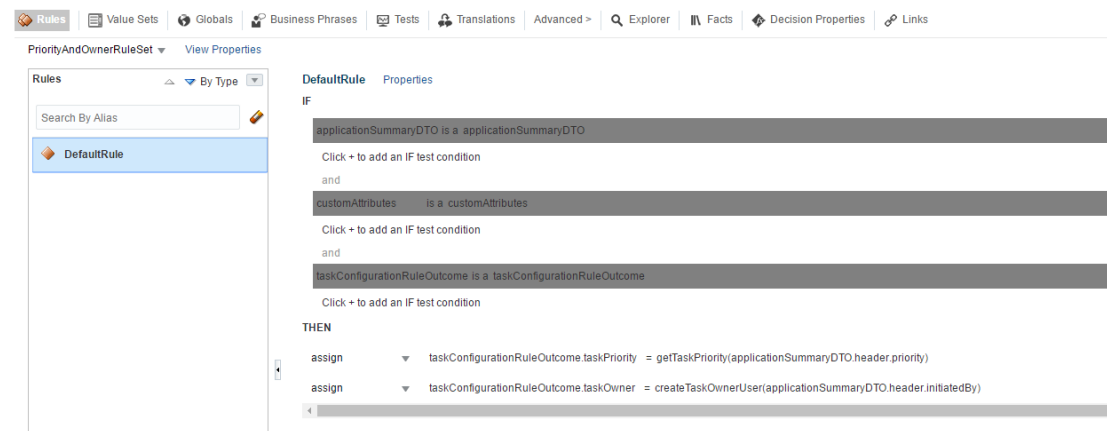


Figure 3–5 shows what a ruleset looks like.

Figure 3–5 SOA Composer - Viewing a ruleset

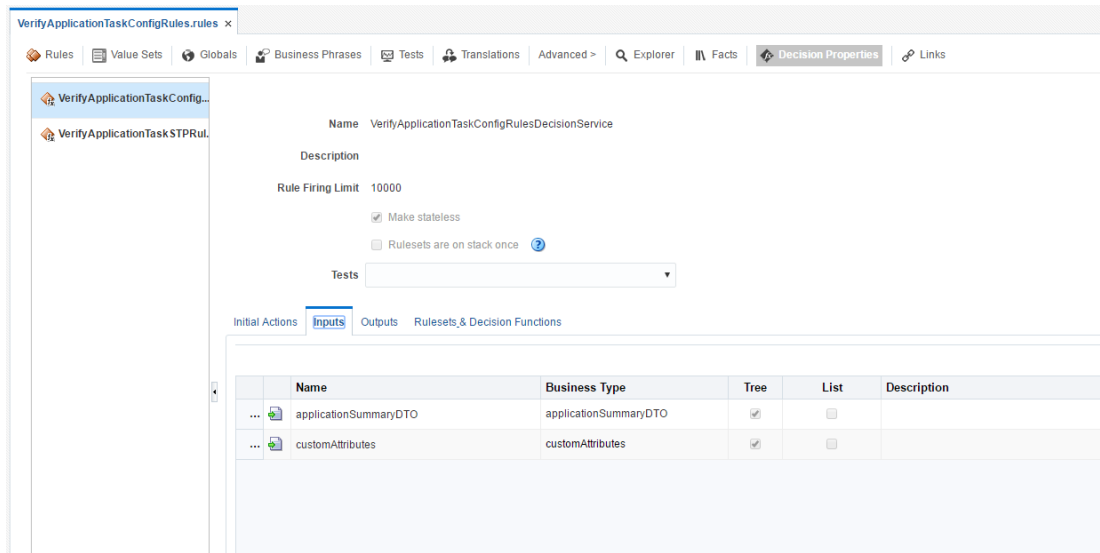


3.2 Inputs to Decision Function

A decision function can take in data objects as input. The rulesets executed by the decision function then work on those inputs to create the output.

Figure 3–6 shows the inputs to a decision function. In this example, there are two input objects - customAttributes of type CustomAttributes and applicationSummaryDTO of type ApplicationSummaryDTO.

Figure 3–6 SOA Composer - Viewing inputs to a Decision Function



In the example, the applicationSummaryDTO is used in the rules to determine the task priority and the task owner. This is illustrated in Figure 3–5.

3.2.1 Custom Input Attributes

CustomAttributes allow three types of attributes - text, number and date, for which, it has following members, respectively:

- CustomTextAttributeList
- CustomNumberAttributeList
- CustomDateAttributeList

Each of these members has a list of respective types,

- CustomTextAttribute
- CustomNumberAttribute
- CustomDateAttribute

All of these three types have a similar steps to configure rules structure and have two members:

- attributeName, of type String
- attributeValue, of type String, int or dateTime, respectively

The UML class diagram of the type CustomAttributes is shown in Figure 2-g. For details on dateTime, please refer <http://www.w3.org/TR/xmlschema-2/#dateTime>

To access the custom attributes passed as input to the decision service, following three functions are provided:

- `getCustomTextAttribute(CustomAttributes customAttributes, String attributeName)`
 - Return type - String
- `getCustomNumberAttribute(CustomAttributes customAttributes, String attributeName)`
 - Return type - int
- `getCustomDateAttribute(CustomAttributes customAttributes, String attributeName)`
 - Return type - XMLGregorianCalendar

Figure 3–7 shows example usage of custom attributes.

Figure 3–7 SOA Composer – Example Usage of Custom Attributes

```

assign new ▾ String dummyCustomInputText = getCustomTextAttribute(customAttributes, "dummyCustomInputText")
assign ▾ dummyCustomInputText = dummyCustomInputText.toUpperCase()
addCustomTextAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputText", dummyCustomInputText)
assign new ▾ int dummyCustomInputNumber = getCustomNumberAttribute(customAttributes, "dummyCustomInputNumber")
assign ▾ dummyCustomInputNumber = dummyCustomInputNumber + 7
addCustomNumberAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputNumber", dummyCustomInputNumber)
assign new ▾ XMLGregorianCalendar dummyCustomInputDate = getCustomDateAttribute(customAttributes, "dummyCustomInputDate")
assign ▾ dummyCustomInputDate = XMLDate.add days to(dummyCustomInputDate, 1)
addCustomDateAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputDate", dummyCustomInputDate)

```

3.3 Output from Decision Function

The output for all task configuration decision functions is of the type `TaskConfigurationRuleOutcome`. This object holds as its attributes, the parameters needed for task configuration. The values for its attributes are set using the rules in the rulesets.

Figure 3–8 shows output definition of decision function.

Figure 3–8 SOA Composer – Viewing Output of a Decision Function

The screenshot shows the SOA Composer interface for configuring a decision function. The main window displays the configuration for the decision function `VerifyApplicationTaskConfigRulesDecisionService`. The `Outputs` tab is selected, showing a table with the following data:

Name	Business Type	Tree	List	Description
taskConfigurationRuleOutcome	taskConfigurationRuleOutcome	<input type="checkbox"/>	<input type="checkbox"/>	

The interface also shows various configuration options such as `Rule Firing Limit` (10000), `Make stateless` (checked), and `Rulesets are on stack once` (unchecked).

3.3.1 List of Configurable Attributes in Rule Outcome

The following human task attributes can be set in TaskConfigurationRuleOutcome object:

1. Task Priority

The task priority can be set by assigning an integer value to the taskPriority attribute of the TaskConfigurationRuleOutcome object. For example, assign `taskConfigurationRuleOutcome.taskPriority = 3`

2. Service Level Agreement (SLA)

SLA consists of taskExpirationDuration and taskDueDuration.

To set SLA for the human task:

- create a new Sla object using `createSLA(String expirationDuration, String dueDuration)`
- assign it to `taskConfigurationRuleOutcome.sla`

For example:

```
assign taskConfigurationRuleOutcome.sla = createSLA("P5D", "P1D")
```

expirationDuration and dueDuration are of the type `xsd:duration` encoded in String. The format of `xsd:duration` is *PnYnMnDnHnMnS*.

P is a literal value that starts the expression

nY represents n years

nM represents n months

nD represents n days

T is a literal value that separates date and time

nH represents n hours

nM represents n minutes

nS represents n seconds

In the example, we have an expiration duration of 5 days and due duration of 1 day. As another example, duration of 1 Month 15 days are represented by "P1M15D"

For more details on the Duration type, please refer <http://www.w3.org/TR/xmlschema-2/#duration>

3. Task Owner

Task owner can be set via following steps:

- create a new ParticipantSet object using `createTaskOwnerUser(String ownerUser)` or `createTaskOwnerGroup(String ownerGroup)`
- assign the new ParticipantSet object to taskOwner attribute of TaskConfigurationRuleOutcome

For example:

```
assign taskConfigurationRuleOutcome.taskOwner = createTaskOwnerUser("user1")
```

4. Stage Participant

Stage participant attribute, stageParticipant is of the type StageParticipant and it consists of following members:

- participant: A ParticipantSet object that holds the participant users and groups.
- filter: A UserFilterCriteria object that holds one or more than one UserAttributeFilterCriteria on which the users will be filtered. A criteria has an attribute name, attributeName, the value of which would determine the filter outcome, the reference value, attributeValue and one operator, out of equals, greater than, less than and in, which decided the type of comparison to be made between the actual value of the attribute and the reference value.

Following steps are supposed to be followed for assigning the stageParticipant:

- Create a ParticipantSet using one of the following functions.
 - createParticipant(String groups, String users): ParticipantSet
 - createParticipantFromUsers(String users): ParticipantSet
 - createParticipantFromGroups(String groups): ParticipantSet
- Create a new UserFilterCriteria.
- Create UserAttributeFilterCriteria objects using CreateUserAttributeFilterCriteria(String attributeName, String attributeValue, UserAttributeFilterOperator operator) and add them to UserFilterCriteria using addUserAttributeFilterCriteria(UserFilterCriteria filter, UserAttributeFilterCriteria attributeFilter)
- Create a new StageParticipant from the ParticipantSet and the UserFilterCriteria using the function createStageParticipant(ParticipantSet participant, UserFilterCriteria filter)
- Add StageParticipant to the TaskConfigurationRuleOutcome using addStageParticipant or addStageParticipantWithStageName

Figure 3–9 shows an example for assigning a StageParticipant.

Figure 3–9 SOA Composer – Example for adding Stage Participant

```
assign new ▾ ParticipantSet participant = createParticipantFromGroups("Administrators")
assign new ▾ UserFilterCriteria filter = new UserFilterCriteria()
assign new ▾ UserAttributeFilterCriteria attributeFilter1 = createUserAttributeFilterCriteria("dummyAttributeName1","dummyAttributeValue1",UserAttrib
addUserAttributeFilterCriteria(filter,attributeFilter1)
assign new ▾ UserAttributeFilterCriteria attributeFilter2 = createUserAttributeFilterCriteria("dummyAttributeName2","dummyAttributeValue2",UserAttrib
addUserAttributeFilterCriteria(filter,attributeFilter2)
assign new ▾ StageParticipant stageParticipant = createStageParticipant(participant, filter)
addStageParticipant(taskConfigurationRuleOutcome, stageParticipant)
```

5. Custom Output Attributes

Following methods may be used in order to add custom text, number or date attributes, respectively to the TaskConfigurationRuleOutcome:

- addCustomTextAttribute
- addCustomNumberAttribute
- addCustomDateAttribute

Figure 3–10 shows example usage of custom attributes.

Figure 3–10 SOA Composer – Example usage of custom attributes

```
assign new ▾ String dummyCustomInputText = getCustomTextAttribute(customAttributes, "dummyCustomInputText")
assign ▾ dummyCustomInputText = dummyCustomInputText.toUpperCase()
addCustomTextAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputText", dummyCustomInputText)
assign new ▾ int dummyCustomInputNumber = getCustomNumberAttribute(customAttributes, "dummyCustomInputNumber")
assign ▾ dummyCustomInputNumber = dummyCustomInputNumber + 7
addCustomNumberAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputNumber", dummyCustomInputNumber)
assign new ▾ XMLGregorianCalendar dummyCustomInputDate = getCustomDateAttribute(customAttributes, "dummyCustomInputDate")
assign ▾ dummyCustomInputDate = XMLDate.add days to(dummyCustomInputDate, 1)
addCustomDateAttribute(taskConfigurationRuleOutcome, "dummyCustomOutputDate", dummyCustomInputDate)
```

3.3.2 List of Functions to Set Rule Outcome

Following functions are available to configure the TaskConfigurationRuleOutcome:

1. **getTaskPriority(String priority): int**

The input to this function is a number as a String and it returns the number as int type. For example, variable1.priority has a value of 2
getTaskPriority(variable1.priority) will return 2.

This method can be used when assigning the TaskConfigurationRuleOutcome.taskPriority

2. **createSLA(String expirationDuration, String dueDuration): Sla** This method takes as inputs the expiration duration and due duration as Strings written in xsd:duration format. Please refer to the description of xsd:duration in section 3.2.1 - 2 Service Level Agreement (SLA) for more details.

3. **createParticipantFromUsers(String users): ParticipantSet**

Use this method to create ParticipantSet from a string containing user names separated with commas, that is, user1,user2,user3. The ParticipantSet can then be used as an input parameter to createStageParticipant function.

4. **createParticipantFromGroups(String groups): ParticipantSet**

Use this method to create ParticipantSet from a string containing group names separated with commas, that is, group1,group2. The ParticipantSet can then be used as an input parameter to createStageParticipant function.

5. **createParticipant(String groups, String users): ParticipantSet**

Use this method to create ParticipantSet containing users as well as groups. The first argument, groups, is a string containing group names separated with commas, that is, group1,group2, and the second argument, users, is a string containing user names separated with commas, that is,

user1,user2,user3. The ParticipantSet can then be used as an input parameter to createStageParticipant function.

6. **getCustomTextAttribute(CustomAttributes customAttributes,String attributeName): String**

This method is used to fetch a text attribute from a CustomAttributes object. Its inputs are:

- customAttributes: the CustomAttributes object from which attribute is to be fetched.
- attributeName: string containing the name of attribute that is to be fetched.

It returns the value for the specified attribute as a String.

7. **getCustomNumberAttribute (CustomAttributes customAttributes, String attributeName): int**

This method is used to fetch a number attribute from a CustomAttributes object. Its inputs are:

- customAttributes: the CustomAttributes object from which attribute is to be fetched
- attributeName: string containing the name of attribute that is to be fetched

It returns the value for the specified attribute as a int.

8. **getCustomDateAttribute (CustomAttributes customAttributes,String attributeName): XMLGregorianCalendar**

This method is used to fetch a date attribute from a CustomAttributes object. Its inputs are:

- customAttributes: the CustomAttributes object from which attribute is to be fetched.
- attributeName: string containing the name of attribute that is to be fetched.

It returns the value for the specified attribute as an XMLGregorianCalendar.

XMLGregorianCalendar is the java representation for xml dateTime. For more information see, <http://docs.oracle.com/javase/1.5.0/docs/api/javax/xml/datatype/XMLGregorianCalendar.html>

9. **createUserAttributeFilterCriteria (String attributeName,String attributeValue,UserAttributeFilterOperator operator): UserAttributeFilterCriteria**

This method constructs a new UserAttributeFilterCriteria object using the given parameters. Its inputs are:

- attributeName: string containing name of the attribute on which the filter criteria is based on.
- attributeValue: string containing the reference value with which the actual value of the attribute is compared to.
- operator: userAttributeFilterOperator object specifying the operator to be used for comparison. The allowed values are GREATER_THAN, LESS_THAN, EQUALS and IN. Refer Figure 3-7 for the UML class diagram of UserAttributeFilterOperator and the related types.

10. **addUserAttributeFilterCriteria (UserFilterCriteriauserFilterCriteria, UserAttributeFilterCriteria UserAttributeFilterCriteria): UserFilterCriteria**

This method adds a UserAttributeFilterCriteria object to the given UserFilterCriteria object. Its input are:

- `userFilterCriteria`: `userFilterCriteria` object to which the attribute filter criteria needs to be added.
- `userAttributeFilterCriteria`: `userAttributeFilterCriteria` object which needs to be added to `userFilterCriteria`. `UserAttributeFilterCriteria` can be created using the function `createUserAttributeFilterCriteria`.

11. **`createStageParticipant (ParticipantSet participant, UserFilterCriteria userFilter): StageParticipant`**

This method is used to create a `StageParticipant` from `ParticipantSet` and a `UserFilterCriteria`, which are passed in as following parameters.

- `participant`: `participantSet` object which can be created using any of the three functions `createParticipant`, `createParticipantFromUsers` or `createParticipantFromGroups`. `userFilter`:
- `userFilterCriteria` object.

12. **`addStageParticipant (TaskConfigurationRuleOutcome taskConfigurationRuleOutcome, StageParticipant stageParticipant): TaskConfigurationRuleOutcome`**

This method is used to add a `StageParticipant` to a `TaskConfigurationRuleOutcome` object.

- `taskConfigurationRuleOutcome`: `taskConfigurationRuleOutcome` to which the stage participant is to be added.
- `stageParticipant`: `stageParticipant` object which is added to `taskConfigurationRuleOutcome`. A `StageParticipant` can be created using `createStageParticipant` function

The function `addStageParticipantWithStageName` can also be used to the same task, and it also has the capability of setting the stage name.

13. **`createTaskOwnerUser (String ownerUser): ParticipantSet`**

This method is used to create a `ParticipantSet`, that is suitable to be set to `TaskConfigurationRuleOutcome.taskOwner`, from a string containing the owner user name, for example, `user1`. The `ParticipantSet` can then be assigned to `TaskConfigurationRuleOutcome.taskOwner`.

14. **`createTaskOwnerGroup (String ownerGroup): ParticipantSet`**

This method is used to create a `ParticipantSet`, that is suitable to be set to `TaskConfigurationRuleOutcome.taskOwner`, from a string containing the owner group name, for example, `group1`. The `ParticipantSet` can then be assigned to `TaskConfigurationRuleOutcome.taskOwner`.

15. **`addCustomTextAttribute (TaskConfigurationRuleOutcome ruleOutcome, String attrName, String attrValue): void`**

This method is used to add a custom text attribute to a `TaskConfigurationRuleOutcome` object. Its inputs are:

- `ruleOutcome`: the `TaskConfigurationRuleOutcome` object to which attribute is to be added.
- `attrName`: string containing the name of attribute that is to be added.

- attrValue: string containing the value of attribute that is to be added.

16. **addCustomNumberAttribute(TaskConfigurationRuleOutcome ruleOutcome,String attrName,int attrValue): void**

This method is used to add a custom number attribute to a TaskConfigurationRuleOutcome object. Its inputs are:

- ruleOutcome: the TaskConfigurationRuleOutcome object to which attribute is to be added.
- attrName: string containing the name of attribute that is to be added.
- attrValue: int containing the value of attribute that is to be added.

17. **addCustomDateAttribute (TaskConfigurationRuleOutcome ruleOutcome, String attrName,XMLGregorianCalendar attrValue): void**

This method is used to add a custom date attribute to a TaskConfigurationRuleOutcome object. Its inputs are:

- ruleOutcome: the TaskConfigurationRuleOutcome object to which attribute is to be added.
- attrName: string containing the name of attribute that is to be added.
- attrValue: XMLGregorianCalendar object containing the value of attribute that is to be added.

XMLGregorianCalendar is the java representation for xml dateTime. For more information see, <http://docs.oracle.com/javase/1.5.0/docs/api/javax/xml/datatype/XMLGregorianCalendar.html>

18. **addStageParticipantWithStageName (TaskConfigurationRuleOutcometaskConfigurationRuleOutcome,StageParticipant stageParticipant,StringstageName): TaskConfigurationRuleOutcome**

This method is has a similar function as that of addStageParticipant and is used to add a StageParticipant to a TaskConfigurationRuleOutcome object and also specify a stage name.

- taskConfigurationRuleOutcome: TaskConfigurationRuleOutcome to which the stage participant is to be added.
- stageParticipant: StageParticipant object which is added to taskConfigurationRuleOutcome. A StageParticipant can be created using createStageParticipant function.
- stageName: String containing the desired stage name.

4 Data Management

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

4.1 Batch Execution

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 Batch Execution.

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 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.

4.1.1 Database Backup

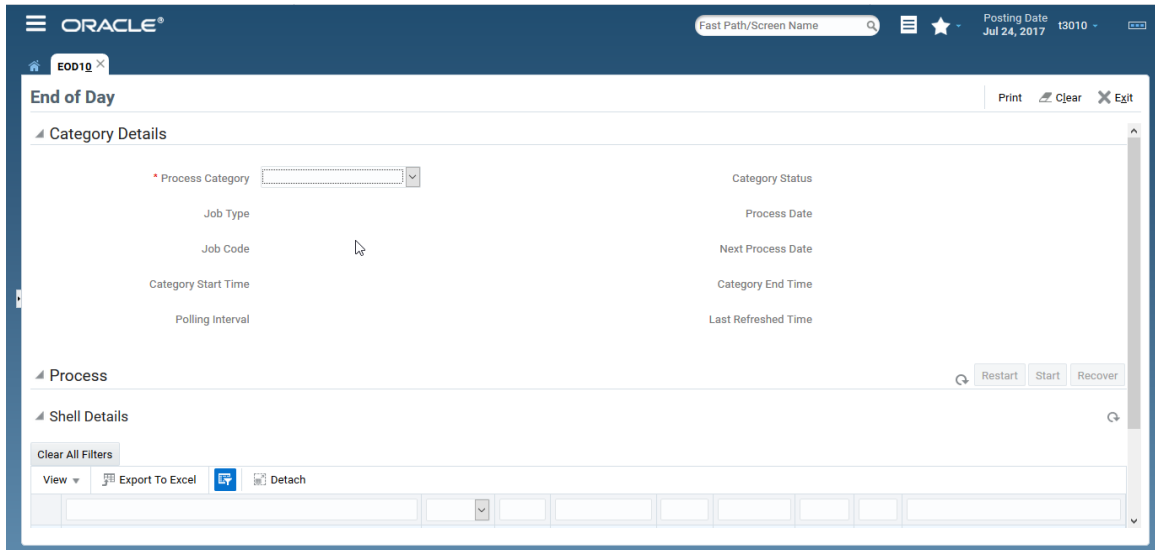
Perform Database Backup before starting with the Batch Execution.

4.1.2 Navigate to End of Day Page

To navigate to the End of Day page:

1. Log in to the Admin Application.
2. Navigate to *End of Day* page either by entering the Fast path **EOD10** or through the menu **Administration > End of Day**.

Figure 4–1 End of Day (Fast path:EOD10)



4.1.3 Cutoff Category Execution

This category marks the logical closure of business in the system to ensure that all online transactions during batch run get processed with the next process date.

To execute the Cutoff category:

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

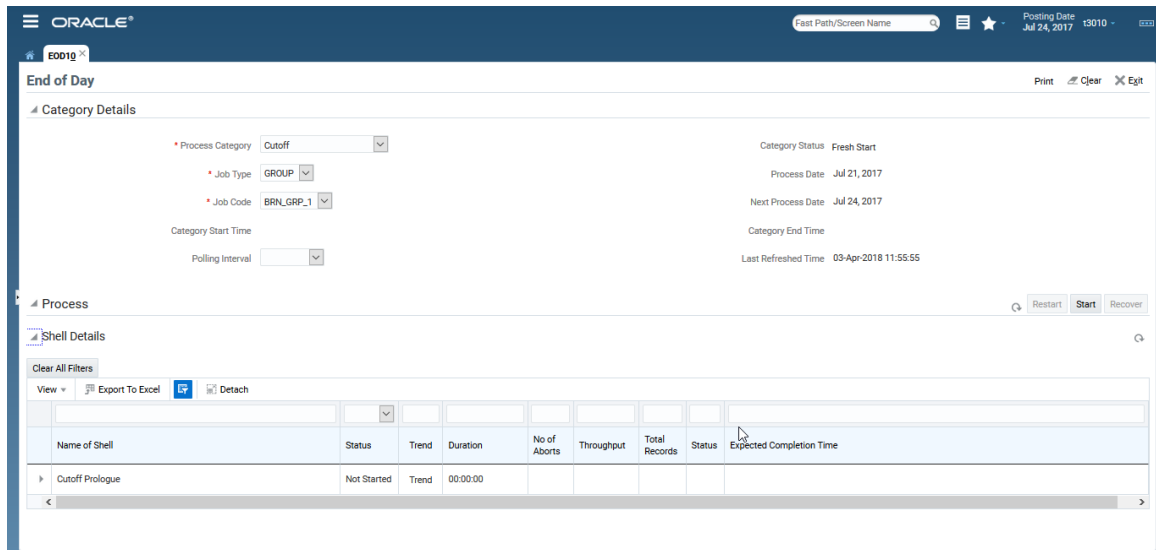
Process Category	Cutoff
Job Type	GROUP
Job Code	BRN_GRP_1

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*.

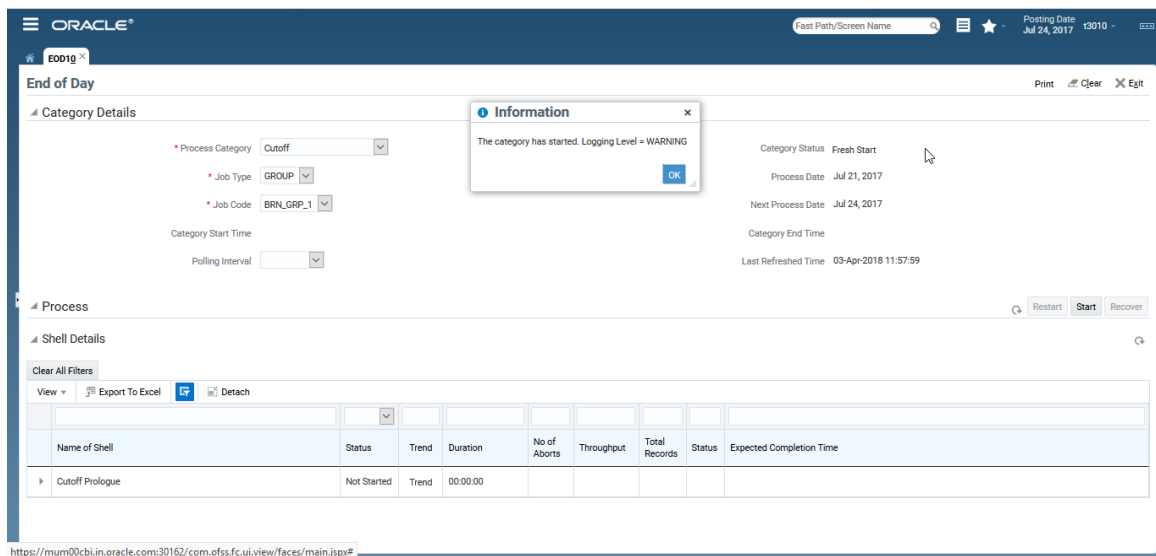
Figure 4–2 Cutoff Category - Not Started



3. Verify the **Process Date** and the **Next Process Date**.
4. Click the **Start** button to begin the execution.

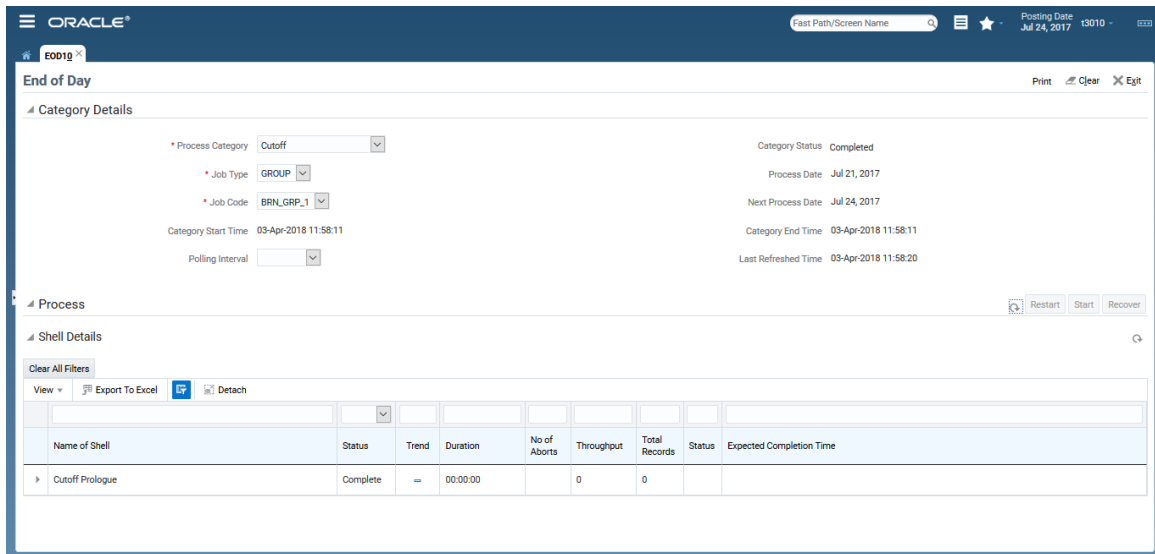
Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress*.

Figure 4–3 Cutoff Category - Start



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

Figure 4–4 Cutoff Category - Complete



4.1.4 End of Day (EOD) Category Execution

This category performs the tasks required to mark closure of a business day in a bank. For example, value date cleaning, instruction expiry, auto disbursement instruction execution, bundle expiry, report generation and so on. Each task or transaction is performed by a shell in a predefined dependency and sequence.

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*.

Figure 4–5 EOD Category - Not Started

The screenshot shows the Oracle EOD19 'End of Day' configuration page. The 'Category Details' section includes:

- Process Category: End of Day
- Job Type: GROUP
- Job Code: BRN_GRP_1
- Category Status: Fresh Start
- Process Date: Jul 21, 2017
- Next Process Date: Jul 24, 2017
- Last Refreshed Time: 03-Apr-2018 11:59:36

 The 'Process' section has buttons for Restart, Start, and Recover. The 'Shell Details' section contains a table with the following data:

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Reg CC Schedule Generation EOD Shell	Not Started	Trend	00:00:00					
Bundle Reports	Not Started	Trend	00:00:00					

3. Verify the **Process Date** and the **Next Process Date**.
4. Click the **Start** button to begin the execution.

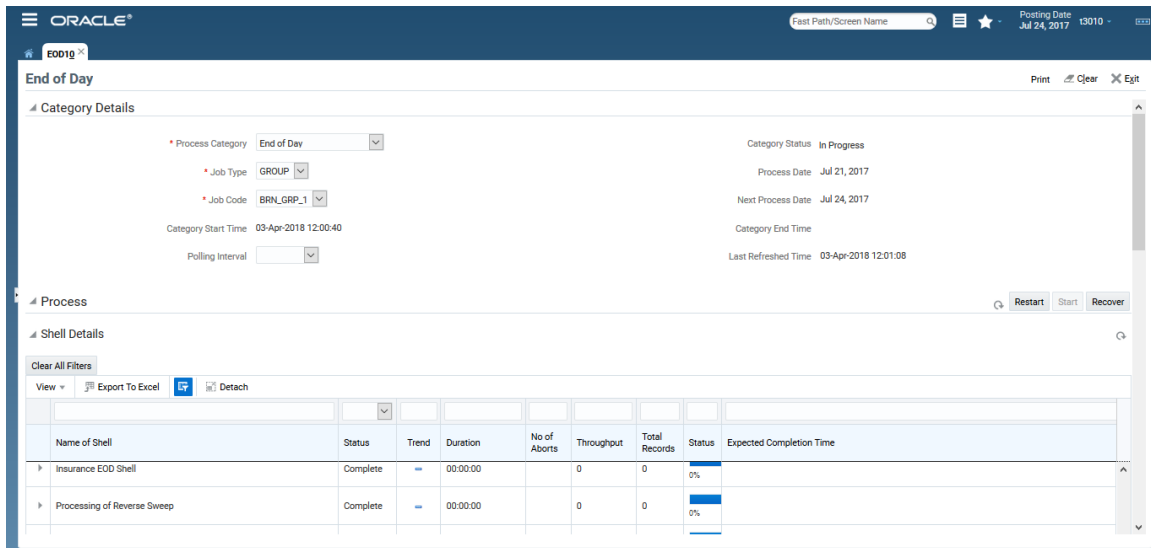
Once the process starts the **Category Status** and the **Shell State** of currently running process display *In Progress*.

Figure 4–6 EOD Category - Start

The screenshot shows the Oracle EOD19 'End of Day' configuration page after the process has started. An 'Information' dialog box is displayed with the message: "The category has started. Logging Level = WARNING". The 'Category Status' is now 'Fresh Start' and the 'Last Refreshed Time' is 03-Apr-2018 12:00:42. The 'Process' section has the 'Start' button highlighted. The 'Shell Details' section contains a table with the following data:

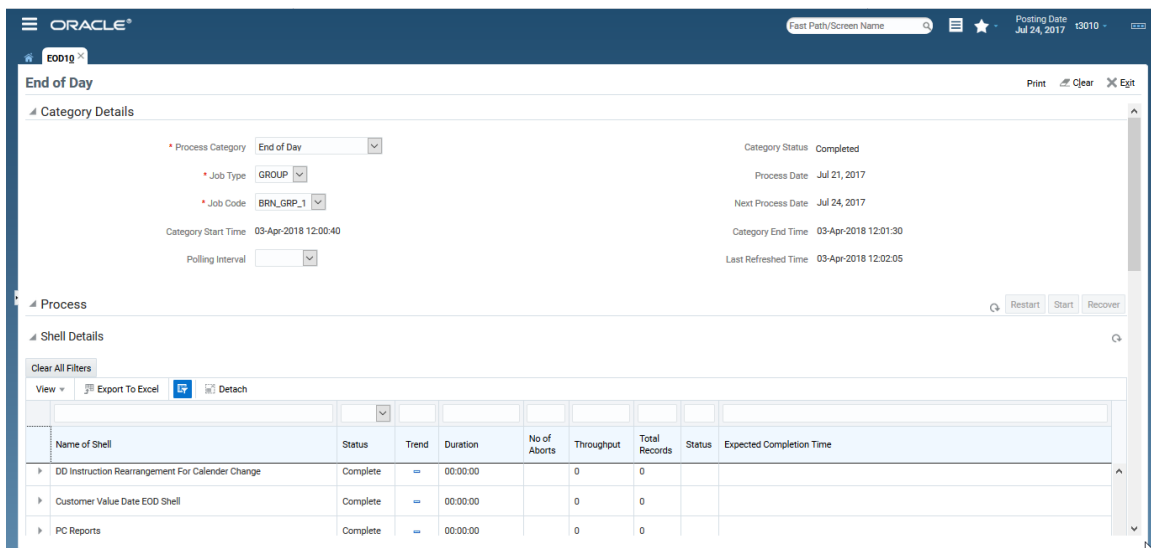
Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Submission EOD Expiry Shell	In Progress		00:00:00	0	0	0	0%	03-Apr-2018 08:00:41
Value Date EOD Shell	Not Started		00:00:00	0	0	0	0%	

Figure 4–7 EOD Category - In Progress



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

Figure 4–8 EOD Category - Complete



4.1.5 Internal System EOD Category Execution

This category performs interest accrual, interest capitalisation, interest compounding, accounting balance verification, ledger balance verification and update and related reporting.

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*.

Figure 4–9 Internal System EOD Category - Not Started

The screenshot displays the Oracle EOD19 interface. The top navigation bar includes the Oracle logo, a search field for 'Fast Path/Screen Name', and the posting date 'Jul 24, 2017 13:010'. The main content area is titled 'End of Day' and contains several sections:

- Category Details:** This section shows configuration options for the process. The 'Process Category' is set to 'Internal System EOD', 'Job Type' is 'GROUP', and 'Job Code' is 'BRN_GRP_1'. The 'Category Status' is 'Fresh Start', 'Process Date' is 'Jul 21, 2017', and 'Next Process Date' is 'Jul 24, 2017'. Other fields include 'Category Start Time', 'Polling Interval', 'Category End Time', and 'Last Refreshed Time' (03-Apr-2018 12:03:09).
- Process:** This section contains buttons for 'Restart', 'Start', and 'Recover'.
- Shell Details:** This section includes a 'Clear All Filters' button and a table with columns for 'Name of Shell', 'Status', 'Trend', 'Duration', 'No of Aborts', 'Throughput', 'Total Records', 'Status', and 'Expected Completion Time'. The table lists two shells: 'Lending Account Statistics Shell' and 'Account Action Internal EOD Shell', both with a status of 'Not Started'.

3. Verify the **Process Date** and the **Next Process Date**.
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 4–10 Internal System EOD Category - Start

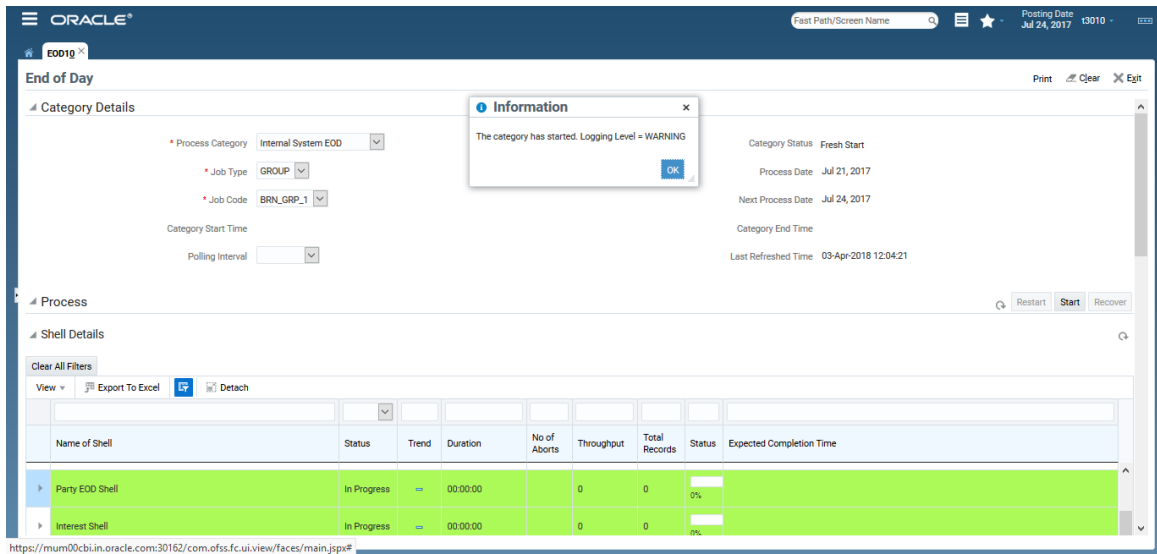
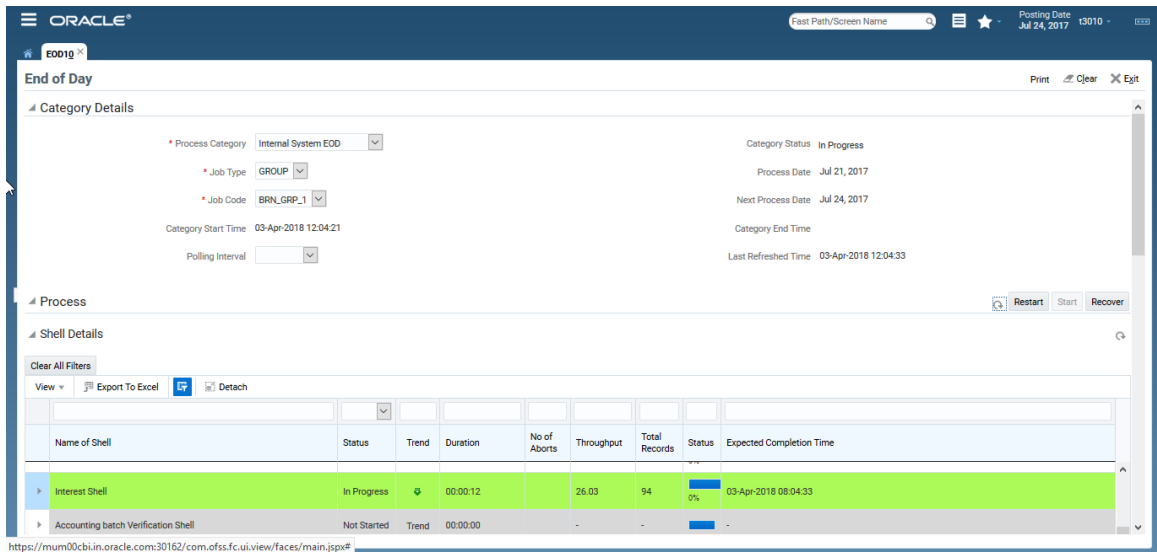


Figure 4–11 Internal System EOD Category - In Progress



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

Figure 4–12 Internal System EOD Category - Complete

4.1.6 Beginning of Day (BOD) Category Execution

This category performs the tasks required for opening a business day in a bank. For example, standing instruction, sweepout instruction, loan account charging, periodic repayment instruction execution, period fee charging, and report generation. Each task or transaction is performed by a shell in a predefined dependency and sequence.

To execute the Beginning of Day category:

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

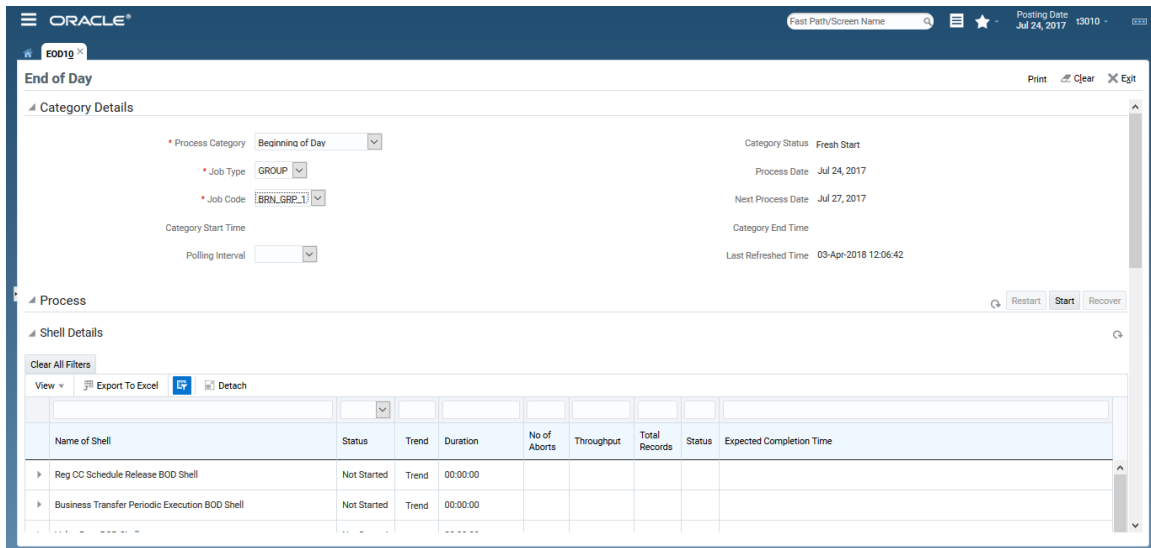
Process Category	Beginning of Day
Job Type	GROUP
Job Code	BRN_GRP_1

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*.

Figure 4–13 BOD Category - Not Started



3. Verify the **Process Date** and the **Next Process Date**.
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 4–14 BOD Category - Started

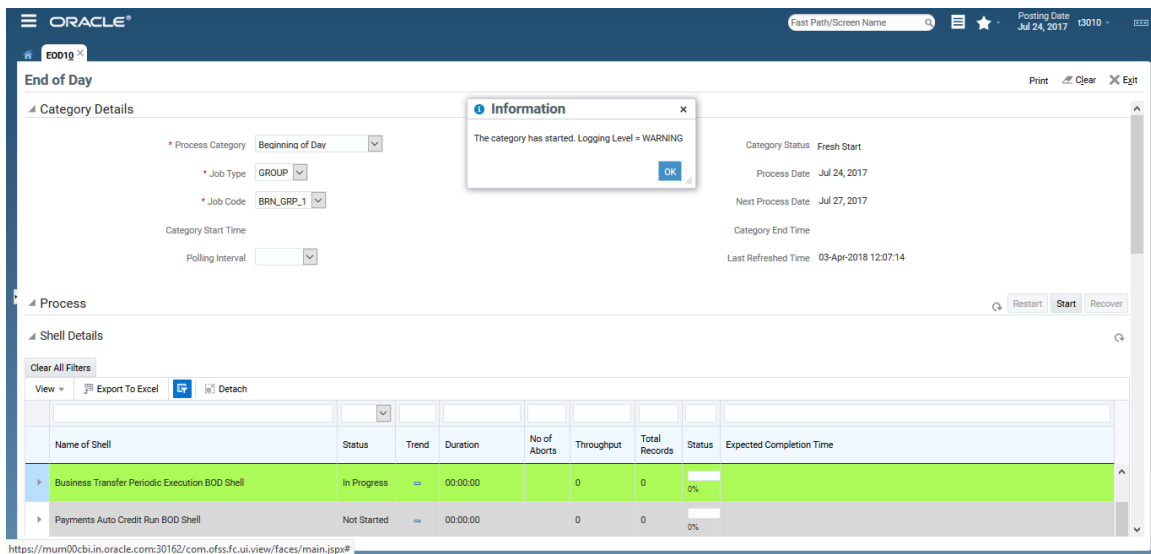


Figure 4–15 BOD Category - In Progress

The screenshot shows the Oracle EOD19 interface for the 'End of Day' category. The 'Category Details' section shows the following information:

- Process Category: Beginning of Day
- Job Type: GROUP
- Job Code: BRN_GRP_1
- Category Start Time: 03-Apr-2018 12:07:14
- Category Status: In Progress
- Process Date: Jul 24, 2017
- Next Process Date: Jul 27, 2017
- Category End Time: (blank)
- Last Refreshed Time: 03-Apr-2018 12:07:26

The 'Process' section shows a table of shells:

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Untanking Adjustment Posting	In Progress	▲	00:00:12		14.5	4	2%	03-Apr-2018 08:07:26
Customer Value Date BOD Shell	Complete	▬	00:00:00		0	0	0%	

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

Figure 4–16 BOD Category - Completed

The screenshot shows the Oracle EOD19 interface for the 'End of Day' category. The 'Category Details' section shows the following information:

- Process Category: Beginning of Day
- Job Type: GROUP
- Job Code: BRN_GRP_1
- Category Start Time: 03-Apr-2018 12:07:14
- Category Status: Completed
- Process Date: Jul 24, 2017
- Next Process Date: Jul 27, 2017
- Category End Time: 03-Apr-2018 12:07:35
- Last Refreshed Time: 03-Apr-2018 12:08:29

The 'Process' section shows a table of shells:

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Reg CC Schedule Release BOD Shell	Complete	▬	00:00:00		0	0		
Customer Value Date BOD Shell	Complete	▬	00:00:00		0	0		
Human Task EOD Resume Shell	Complete	▬	00:00:00		0	0		

4.1.7 Housekeeping Category Execution

This category performs the tasks such as statement generation, alert generation, exposure tracking, offset benefit calculation, and facility closure.

To execute the Housekeeping category:

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

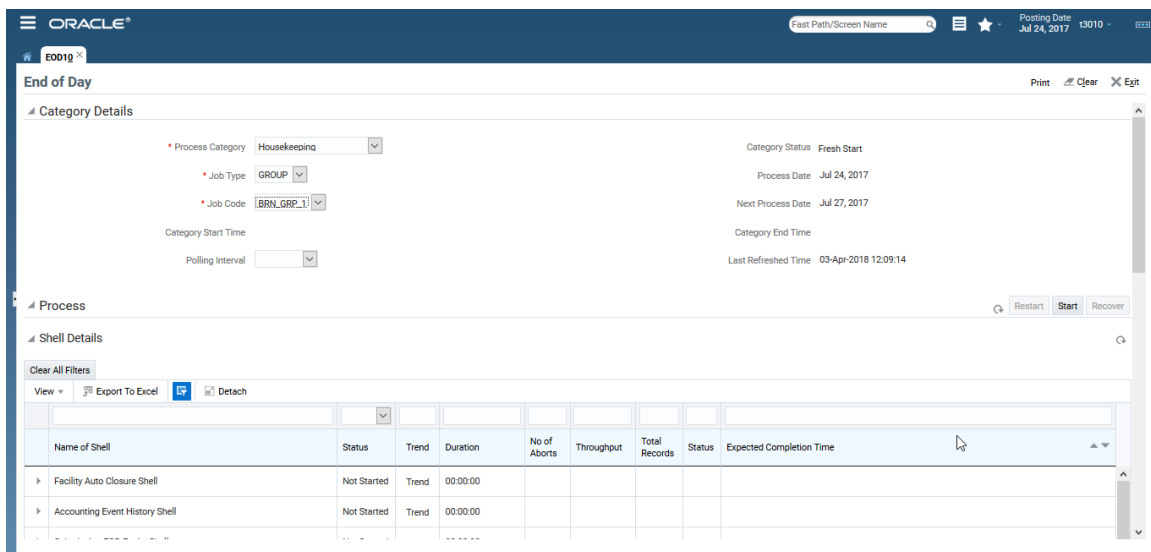
Process Category	Housekeeping
Job Type	GROUP
Job Code	BRN_GRP_1

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*.

Figure 4–17 Housekeeping Category - Not Started



3. Verify the **Process Date** and the **Next Process Date**.
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 4–18 Housekeeping Category - In Progress

The screenshot displays the Oracle EOD19 interface for the 'End of Day' process. The 'Category Details' section shows the following information:

- Process Category: Housekeeping
- Job Type: GROUP
- Job Code: BRN_GRP_1
- Category Start Time: 03-Apr-2018 12:10:03
- Category Status: In Progress
- Process Date: Jul 24, 2017
- Next Process Date: Jul 27, 2017
- Category End Time: (blank)
- Last Refreshed Time: 03-Apr-2018 12:10:16

The 'Process' section includes buttons for Restart, Start, and Recover. The 'Shell Details' section shows a table with the following data:

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Move driver table data to History table	Complete	Trend	00:00:00	-	-	-	0%	-
Accounting Event History Shell	Complete	Trend	00:00:00	-	-	-	0%	-

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

Figure 4–19 Housekeeping Category - Completed

The screenshot displays the Oracle EOD19 interface for the 'End of Day' process, now completed. The 'Category Details' section shows the following information:

- Process Category: Housekeeping
- Job Type: GROUP
- Job Code: BRN_GRP_1
- Category Start Time: 03-Apr-2018 12:10:03
- Category Status: Completed
- Process Date: Jul 24, 2017
- Next Process Date: Jul 27, 2017
- Category End Time: 03-Apr-2018 12:22:49
- Last Refreshed Time: 03-Apr-2018 12:22:49

The 'Process' section includes buttons for Restart, Start, and Recover. The 'Shell Details' section shows a table with the following data:

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Move driver table data to History table	Complete	==	00:00:00	0	0	0		
Accounting Event History Shell	Complete	==	00:00:00	0	0	0		

4.1.8 Alert Generation Category Execution

This category is used to generate previously logged alerts.

To execute the Alert Generation category:

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

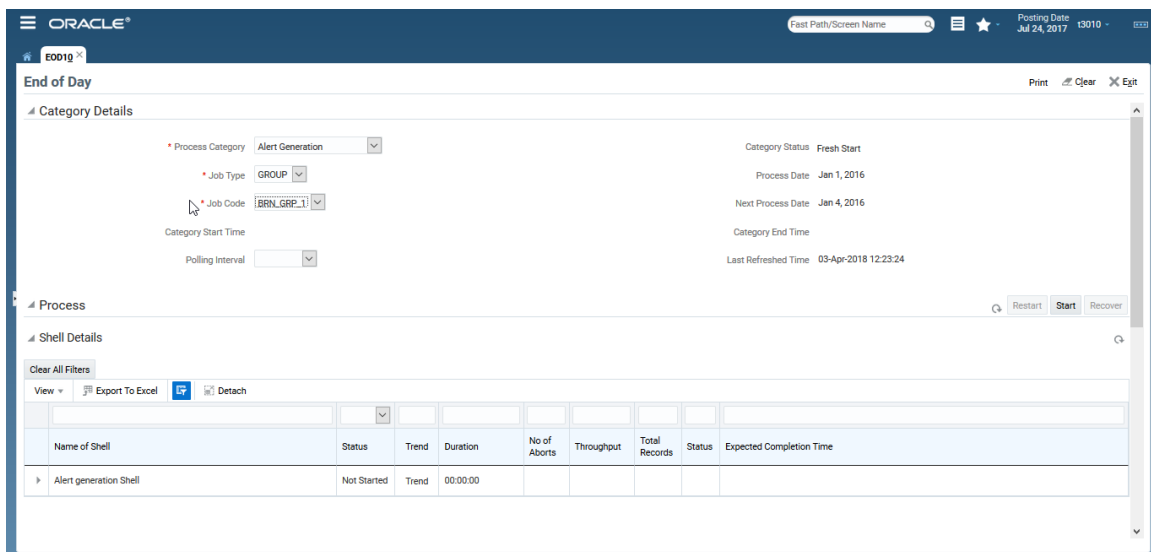
Process Category	Alerts Generation
Job Type	GROUP
Job Code	BRN_GRP_1

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*.

Figure 4–20 Alert Generation Category - Not Started



3. Verify the **Process Date** and the **Next Process Date**.
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 4–21 Alert Generation Category - In Progress

The screenshot shows the Oracle EOD19 interface for the 'Alert Generation' category. The 'Category Status' is 'In Progress'. The 'Shell Details' table shows the 'Alert generation Shell' with a status of 'In Progress'.

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Alert generation Shell	In Progress	-	00:00:09		276.32	84	0%	03-Apr-2018 08:24:00

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

The screenshot shows the Oracle EOD12 interface for the 'Alert Generation' category. The 'Category Status' is 'Completed'. The 'Shell Details' table shows the 'Alert generation Shell' with a status of 'Complete'.

Name of Shell	Status	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Alert generation Shell	Complete	=	00:00:20		276	84		

4.2 Batch Exception Recovery

Batch Exception Recovery refers to mechanism to allow support and business users perform actions on the records that were skipped during batch execution. During batch execution, if the number of failures due to business exception is less than pre-configured threshold, such records are skipped for future processing.

The batch exception recovery can be done using the Batch Exception Recovery (Fast Path: OPA007) page. It is recommended that user in support or operations role, checks this page after every batch processing is completed for any PENDING records.

This section explains the steps involved in Batch Exception Recovery.

4.2 Batch Exception Recovery

Batch exception recovery actions can be broadly classified in two categories:

- Actions for Support/Operations user (Performed on Batch Exception Recovery page)
- Actions for Business user (Viewed in worklist application and actioned using OBP screens, data patches.)

In its entire life cycle, the batch exception record will go through the above mentioned actions starting with PENDING and ending with either IGNORED or REPROCESSED. Support or Operations user acts on exception record using the Batch Exception Recovery page.

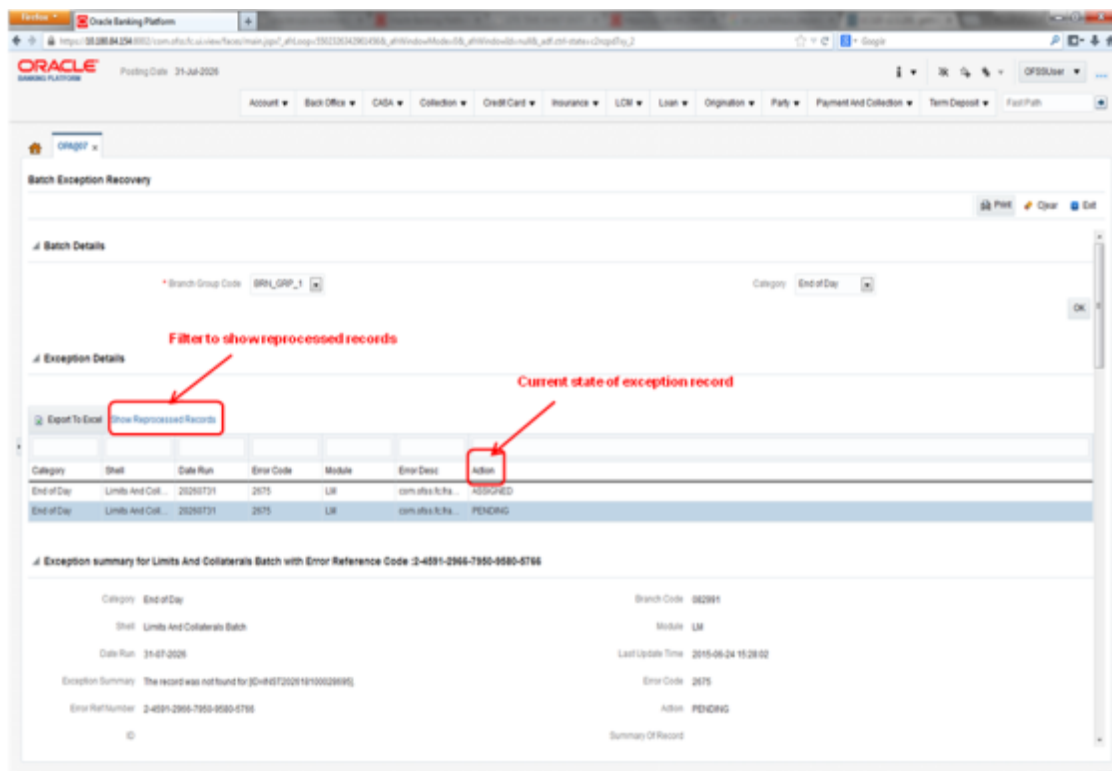
To navigate to the Batch Exception Recovery page:

1. Log in to the Admin application.
2. Navigate to Batch Exception Recovery page either by entering the Fast Path **OPA007** or through the menu **Administration > Batch Exception Recovery**.
3. Select the relevant Category Details as shown in the following table:

Branch Group Code	BRN_GRP_1
Category	End of Day

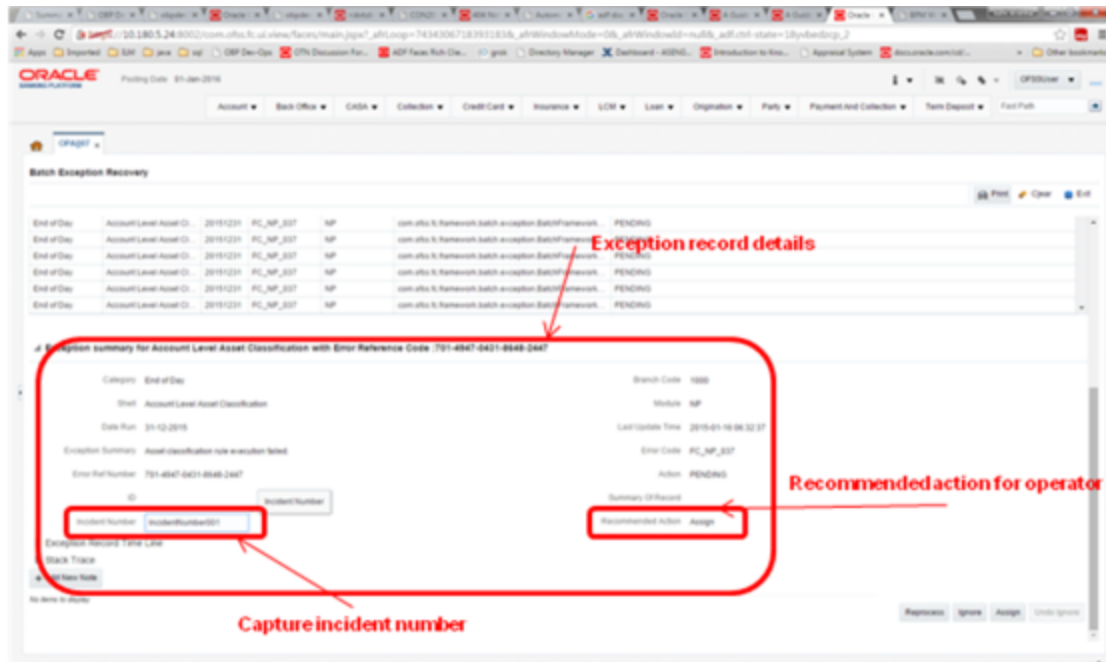
4. Click OK. The Exception Details appear.

Figure 4–22 Exception Details



5. Select an exception record. The additional details such as Stack Trace and Comments appear. One of the important attributes is Recommended Action for the operator.

Figure 4–23 Exception Record Details



6. Check the Stack Trace and Comments. It is recommended that if the current action on an exception record is PENDING and there are no Comments, click Assign to create a task for business user to take appropriate corrective actions on the exception record. Also, look for the Recommended Action. By looking at the Stack Trace, if the support or operator users find a similar previous incident, they can capture the same in the Incident Number field as shown in the above figure.

Note

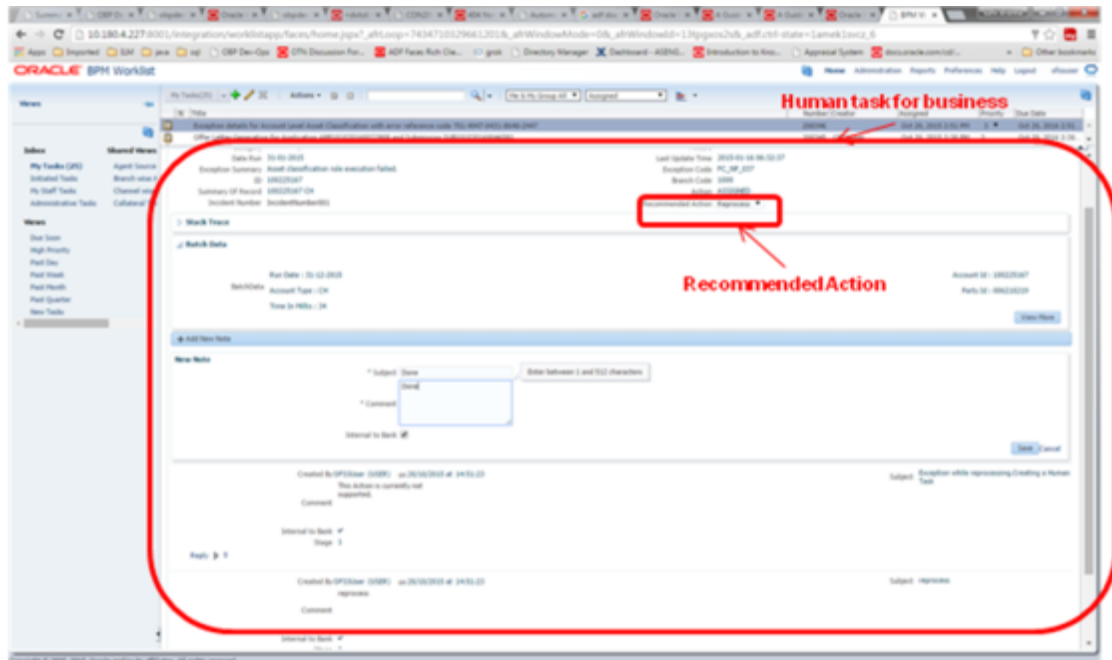
It is mandatory to capture valid meaningful Comments while performing any action on the exception record.

Business user acts on exception record using the Worklist Application

To navigate to the Worklist page:

1. Log in to Worklist application.
2. All the exception records with action as ASSIGNED will appear as a task in the worklist. Select the task to act on.

Figure 4–24 Exception record in Worklist application



3. Click Claim to claim the task.
4. Check the Stack Trace, Batch Data, Incident Number and Comments.
5. Perform appropriate actions using application screens, data patches.
6. Capture comments in Comments section. These comments will be used by the support user to further act on the exception record.
7. Select the mandatory Recommended Action for the support or operator user.
8. Click Done to complete the task. The exception record moves into PENDING state and will be visible to the support user to take further action.

Note

It is mandatory to capture valid meaningful Comments while performing any action on the exception record.

5 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, which the administrator can use to perform a bank and a branch setup.

5.1 Common Services Day 0 Setup

The Common Services setup includes the following sections.

5.1.1 Core Maintenances

Core Entity Services seek 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)
- Business Group (Fast path: CS02)
- Bank Parameters (Fast path: CS03)
- Branch Parameters (Fast path: CS06)
- Other Financial Institution Details (Fast path: CS07)
- Country Codes (Fast path: CS09)
- Financial Cycle (Fast path: CS10)
- Define Payment Calender 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)
- Risk Category Maintenance (Fast path: CS41)
- Risk Code Maintenance (Fast path: CS42)
- Mitigant Code Maintenance (Fast path: CS43)
- Transaction Code Maintenance (Fast path: CS44)

- Define Non-Financial Event Transaction Code Mapping (Fast path: CS45)
- Data Security Configuration (Fast path: CS50)
- Questionnaire Maintenance (Fast path: CS103)
- Section Maintenance (Fast path: 104)

Note

To view the detailed procedure for each application page, see its context sensitive help in the application.

5.1.1.1 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 for each application page, see its context-sensitive help in the application.

5.1.2 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 for each application page, see its context-sensitive help in the application.

5.1.3 Calendar Maintenances

The calendar services are embedded in the common services 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:

- Holiday Rule Maintenance (Fast Path: CAL01)
- Calendar Type Maintenance (Fast Path: CAL02)
- Adhoc Calendar Maintenance (Fast path: CAL03)

Note

To view the detailed procedure for each application page, see its context-sensitive help in the application.

5.2 Accounting Day 0 Setup

The Accounting 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 Accounting maintenances must be completed as a part of bank and branch setup:

- Define System Defined Elements (Fast path: AS013)
- Define Accounting Configuration (Fast path: AS001)
- GAAP Summary (Fast path: AS005)
- Define Bank Parameter (Fast path: AS002)
- Define Branch Parameter (Fast path: AS003)
- Define SDE Range (Fast path: AS012)
- System Defined Elements Class Summary (Fast path: AS011)
- Define Accounting Ledger (Fast path: AS009)
- Define Accounting Ledger (Additional) Details (Fast path: AS010)

- Define Accounting Ledger Group (Fast path: AS008)
- Define Inter Branch Parameters (Fast path: AS006)
- Define Product Group Accounting Entry Template (Fast path: AS014)
- Define Product Accounting Template (Fast path: AS015)
- Define Domain Category Accounting Template (Fast path: AS016)
- Define Product Group Role Mapping (Fast path: AS017)
- Define Product Account Role Mapping (Fast path: AS018)
- Define Domain Role Mapping (Fast path: AS019)
- Define Event Accounting Class (Fast path: AS020)

Note

To view the detailed procedure for each application page, see its context-sensitive help in the application.

5.3 Product Manufacturing Day 0 Setup

Product Manufacturing 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. Product Manufacturing is complete if the respective mandatory tasks are complete.

Prerequisites

Following are the prerequisites for Product Manufacturing Day 0 Maintenances:

- Asset Classification (AC) Plans Maintenance (Fast path: NP002)
- Common Services: Purpose Code, Currency Code, Calendar Maintenance, Bank Policy
- Accounting Template Maintenance
- Facility Category (Fast path: LCMS53)
- Collateral Category
- DMS maintenance: Document Type Definition (Fast path: CNM01), Document Category Definition (Fast path: CNM02), Document Policy Definition (Fast path: CNM03)
- Risk Indicators Impacts Cross-Reference (Fast path: ACCT010)
- Global Parameter (Fast path: LCM48)
- LTV Matrix (Fast path: LCM52)
- Rate Chart Maintenance (Fast path: PR004)
- Index/Margin Index Code Definition (Fast path: PR005)
- Price Policy Chart Maintenance (Fast path: PR007)
- Price Definition (Fast path: PR006)
- Charge Attribute Definition (Fast path: PR008)

-
- Tier Criteria Definition (Fast path: PR009)
 - Price Benefit Chart (Fast path: PR015)

Day 0 Maintenances

The following Product Manufacturing Maintenances must be completed as part of bank and branch set up:

- Define Party Bank Policy (Fast path: PI314)
- 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
- Define Interest Rule (Fast path: PM011)
- Define TD Interest Payout Plan (Fast path: PM012)
- Define CASA Product
- Define Loan Product
- Define TD Product
- Define Credit Card Product
- Define Investment Product
- Define Insurance Product
- Define Product Interest Linkage (Fast path: PM017)
- Link Dependent Offers (Fast path: PM018)
- Define Bonus Interest Parameters (Fast path: PM019)
- Define CASA Offer
- Define TD Offer
- Define Loan Offer
- Define Credit Card Offer
- Define Investment Offer
- Define Insurance Offer
- Installment Rule Details (Fast path: PM058)
- Loan Schedule Type (Fast path: PM059)
- Define Campaigns (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 Affinity (Fast path: PM042)
- Define Question Sets (Fast path: PM046)

Note

To view the detailed procedure for each application page, see its context-sensitive help in the application.

6 Setting Up Sales Offer Handoff

This chapter describes the sales offer handoff related activities to be performed as an administrator.

6.1 Day Zero Setup (Configuration)

This section explain the day zero setup required for setting up the sales offer handoff.

6.1.1 Changes to be done in OBP

Following are the steps:

1. ODI has two schemas, namely, one for source and another for target. Source schema is the schema of the DB to which the OBP HOST is connected. The following tables are to be created in target schema:

FLX_PM_SALES_OFFER_STG
FLX_PM_SALES_OFFER_STG_H
FLX_PM_CC_SALES_OFFER_STG
FLX_PM_CC_SALES_OFFER_STG_H
FLX_PM_CS_SALES_OFFER_STG
FLX_PM_CS_SALES_OFFER_STG_H
FLX_PM_TD_SALES_OFFER_STG
FLX_PM_TD_SALES_OFFER_STG_H
FLX_PM_LN_SALES_OFFER_STG
FLX_PM_LN_SALES_OFFER_STG_H

2. Following configuration is needed in the source schema tables:
 - a. UPDATE FLX_FW_CONFIG_VAR_B SET PROP_VALUE = '10.180.4.60' WHERE PROP_ID = 'odi.server.name'; (where '10.180.4.60' is the HOST IP of ODI Agent i.e. ip of the server on which odi is installed).
 - b. UPDATE FLX_FW_CONFIG_VAR_B SET PROP_VALUE = '15101' WHERE PROP_ID = 'odi.server.port'; (where '15101' is the port number of ODI Agent).
 - c. UPDATE FLX_FW_CONFIG_ALL_B SET PROP_VALUE = 'WORKREP' WHERE PROP_ID = 'WORK_REPOSITORY'; (where 'WORKREP' is the name of work repository).
 - d. INSERT INTO FLX_FW_ENUM_REPRESENTATIONS (ENUM_FQN,ENUM_VALUE,USER_LOCALE,ENUM_NAME,ENUM_REPRESENTATION,ORDINAL_NUMBER,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_DATE,OBJECT_STATUS_FLAG,OBJECT_VERSION_NUMBER) VALUES ('com.ofss.fc.enumeration.ProcessCategoryType','17','en_us','PMU_SALES_OFFER','Sales Offer Handoff Shell',0,'OFSSUSER',TO_TIMESTAMP('SYSDATE','DD-MON-RR HH.MI.SSXFF AM'),'OFSSUSER',TO_TIMESTAMP('SYSDATE','DD-MON-RR HH.MI.SSXFF AM'),'A',1);

- e. UPDATE FLX_FW_CONFIG_ALL_B SET PROP_VALUE = 'SUPERVISOR' WHERE PROP_ID = 'ODI_USER_NAME'; (where 'SUPERVISOR' is password for master repository).
- f. UPDATE FLX_FW_CONFIG_ALL_B SET PROP_VALUE = 'SUNOPSIS' WHERE PROP_ID = 'ODI_USER_PASSWORD'; (where 'SUNOPSIS' is password for master repository).
- g. INSERT INTO FLX_DI_ETL_JOB_DEFINITION (CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, EXECUTION_UNIT, BATCH_CATEGORY, EXECUTION_UNIT_DESC, IS_ENABLED, JOB_FREQUENCY) VALUES ('Admin', TO_DATE('10-05-2013', 'dd-mm-yyyy'), 'Admin', TO_DATE('10-05-2013', 'dd-mm-yyyy'), 'PKG_PM_SALES_OFFER_STG', 17, '', 1, 1);

6.1.2 Changes to be done in ODI

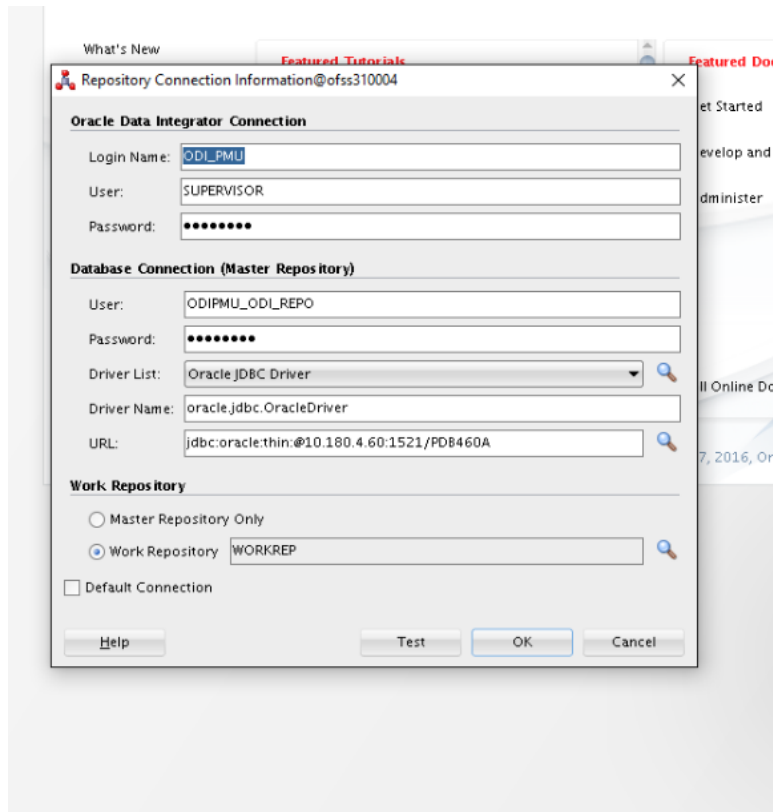
Following are the steps:

Step 1 Connect to Repository

To connect to a repository:

1. Click Connect to Repository.
2. Enter the following details:
 - Password for SUPERVISOR is SUNOPSIS. Its corresponding configuration is given in source schema in Day Zero Setup changes in OBP in source schema tables as given in step e and f. This same username and password needs to be configured on OBP HOST weblogic server console under appConnector.
 - Password for Database Connection: welcome1

Figure 6–1 Repository Connection Information

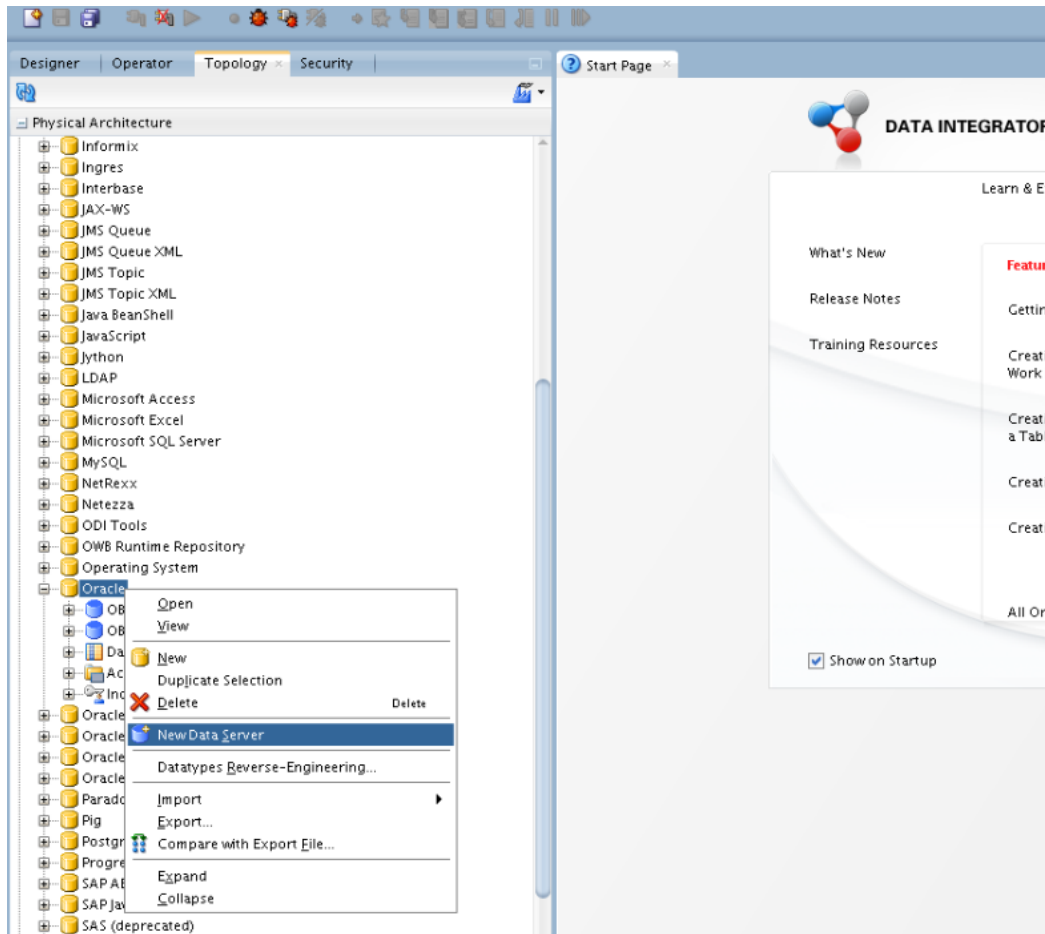


Step 2 Physical Architecture

To generate a physical architecture:

1. Navigate to Topology tab -> Physical Architecture -> Technologies -> Oracle.
2. Right-click and select the **New Data Server** option.

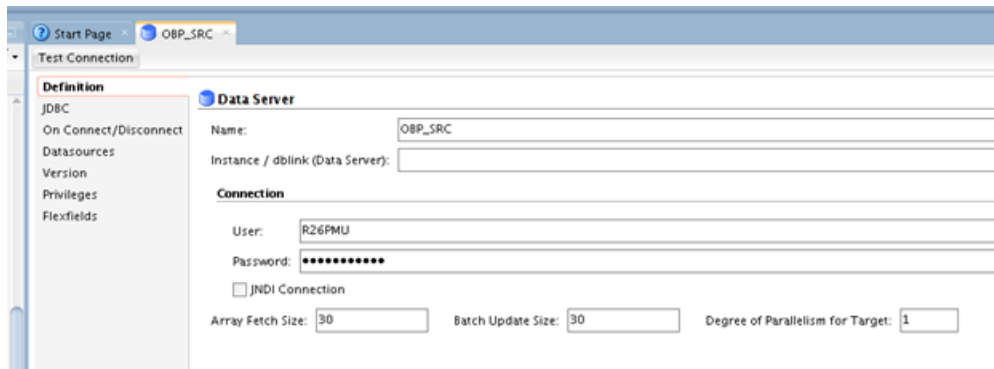
Figure 6–2 Physical Architecture - New Data Server



3. In the **Definition** tab of that data server, enter the data server name, connection username and password of the DB.

For your source data server, the name should be OBP_SRC. The target data server name can be of your choice.

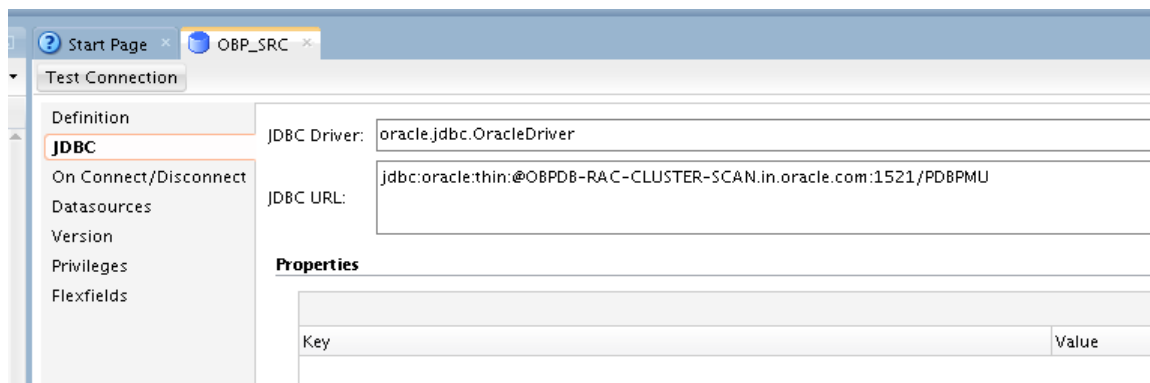
Figure 6–3 Data Server - Definition



The screenshot shows the 'Data Server' configuration window. The 'Definition' tab is active. The 'Name' field is set to 'OBP_SRC'. The 'Instance / dblink (Data Server)' field is empty. Under the 'Connection' section, the 'User' is 'R26PMU' and the 'Password' is masked with asterisks. The 'JNDI Connection' checkbox is unchecked. At the bottom, 'Array Fetch Size' is 30, 'Batch Update Size' is 30, and 'Degree of Parallelism for Target' is 1.

- In the **JDBC** tab, enter the details as shown in Figure 6–4.
Click **Test Connection**. Select Physical agent as Local and click **OK**.

Figure 6–4 Data Server - JDBC

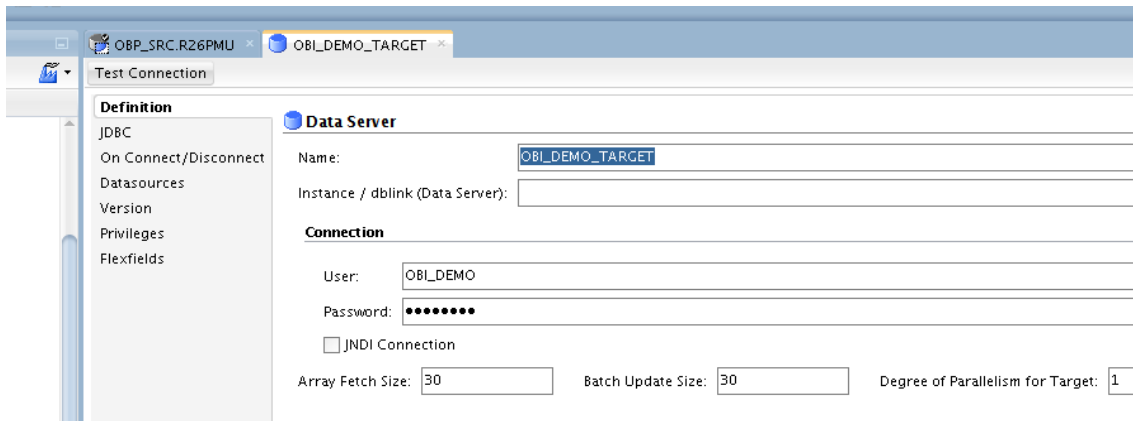


The screenshot shows the 'Data Server' configuration window with the 'JDBC' tab selected. The 'JDBC Driver' is 'oracle.jdbc.OracleDriver' and the 'JDBC URL' is 'jdbc:oracle:thin:@OBPDB-RAC-CLUSTER-SCAN.in.oracle.com:1521/PDBPMU'. Below these fields is a 'Properties' section with a table structure.

Key	Value

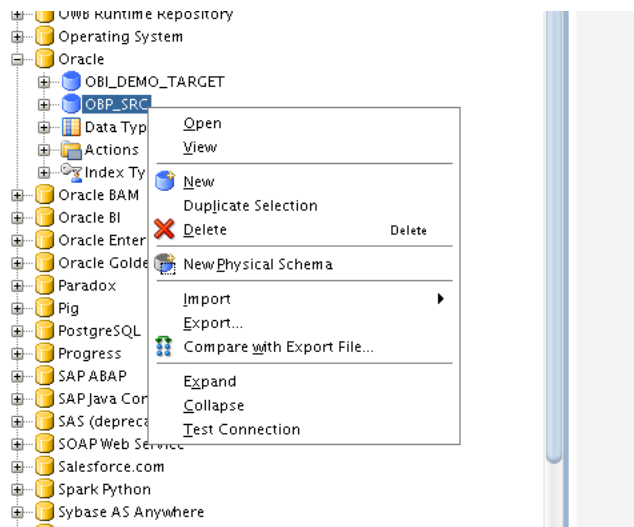
- Repeat the above steps to make a new target data server.

Figure 6–5 Create New Target Data Server



6. Right-click source data server and select **New Physical Schema**.

Figure 6–6 New Physical Schema



7. Select the schema and work schema of the data server which you just created.

Figure 6–7 Physical Schema - Definition

OBP_SRC.R26PMU

Definition

Physical Schema [Data Server: OBP_SRC]

Name: OBP_SRC.R26PMU

Schema (Schema): R26PMU

Schema (Work Schema): R26PMU

Default

Work Tables Prefix

Errors: E\$_ Loading: C\$_ Integration: I\$_ Temporary Indexes: IX\$_

Journalizing elements prefixes

Datastores: J\$_ Views: JV\$_ Triggers: T\$_

Naming Rules

Character Encoding: BASE38

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %SCHEMA.%OBJECT@%DSERVER

Partition Mask: %SCHEMA.%OBJECT PARTITION(%PARTITION)

Sub-Partition Mask: %SCHEMA.%OBJECT SUBPARTITION(%PARTITION)

Local Sequence Mask: %SCHEMA.%OBJECT.nextval

Remote Sequence Mask: %SCHEMA.%OBJECT.nextval@%DSERVER

Local Sequence Current Value Mask: %SCHEMA.%OBJECT.currval

Remote Sequence Current Value Mask: %SCHEMA.%OBJECT.currval@%DSERVER

8. Perform the same steps for target source and select the schema of target data server.

Figure 6–8 Physical Schema - Target Data Server

Definition

Context: **Physical Schema [Data Server: OBI_DEMO_TARGET]**

Version: Name: OBI_DEMO_TARGET.OBI_DEMO

Privileges: Schema (Schema): OBI_DEMO

Flexfields: Schema (Work Schema): OBI_DEMO

Default

Work Tables Prefix

Errors: E\$_ Loading: C\$_ Integration: I\$_ Temporary Indexes: IX\$_

Journalizing elements prefixes

Datastores: J\$ Views: JV\$ Triggers: T\$

Naming Rules

Character Encoding: BASE8

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %SCHEMA.%OBJECT@%DSERVER

Partition Mask: %SCHEMA.%OBJECT PARTITION(%PARTITION)

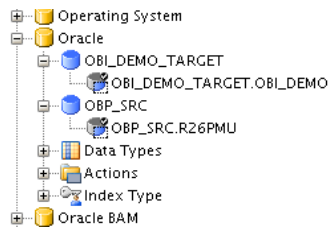
Sub-Partition Mask: %SCHEMA.%OBJECT SUBPARTITION(%PARTITION)

Local Sequence Mask: %SCHEMA.%OBJECT.nextval

Remote Sequence Mask: %SCHEMA.%OBJECT.nextval@%DSERVER

Local Sequence Current Value Mask: %SCHEMA.%OBJECT.currval

Remote Sequence Current Value Mask: %SCHEMA.%OBJECT.currval@%DSERVER

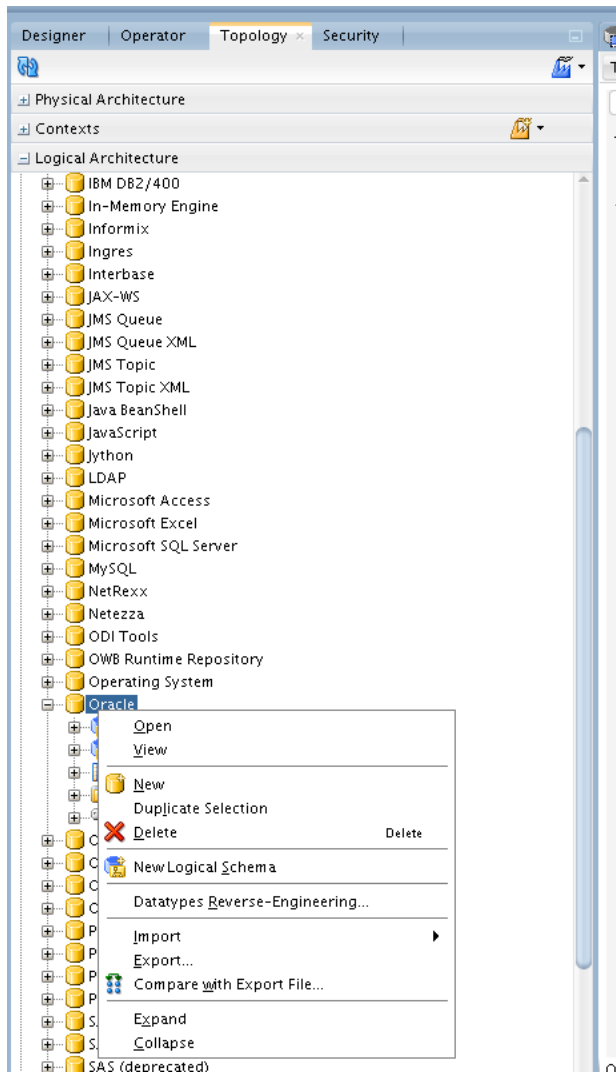
Figure 6–9 Physical Schema List

Step 3 Logical Architecture

To generate a logical architecture:

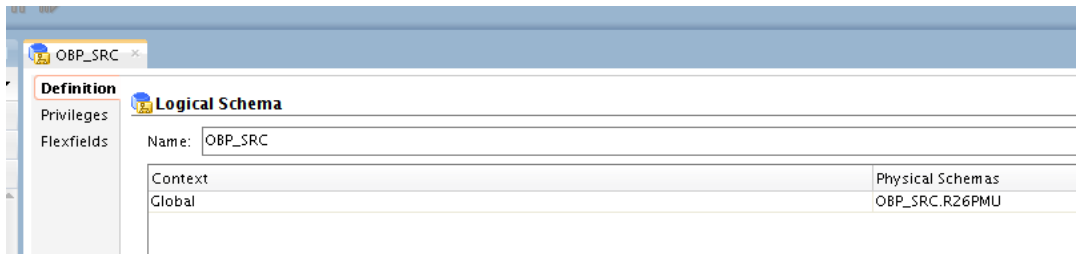
1. Navigate to Topology tab -> Logical Architecture -> Technologies -> Oracle.
2. Right-click and select the **New Logical Schema** option.

Figure 6–10 Logical Architecture - New Logical Schema



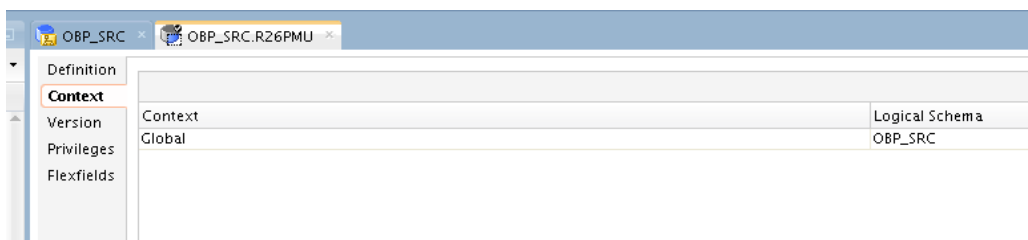
3. In the **Definition** tab of that logical schema, enter the name.
For your source logical schema, the name should be OBP_SRC.
4. Set its context to its corresponding physical schema.

Figure 6–11 Logical Schema - Definition



5. Open the physical schema. Under its context tab, set its context to this logical schema.

Figure 6–12 Physical Schema - Set Context



The architecture is now complete.

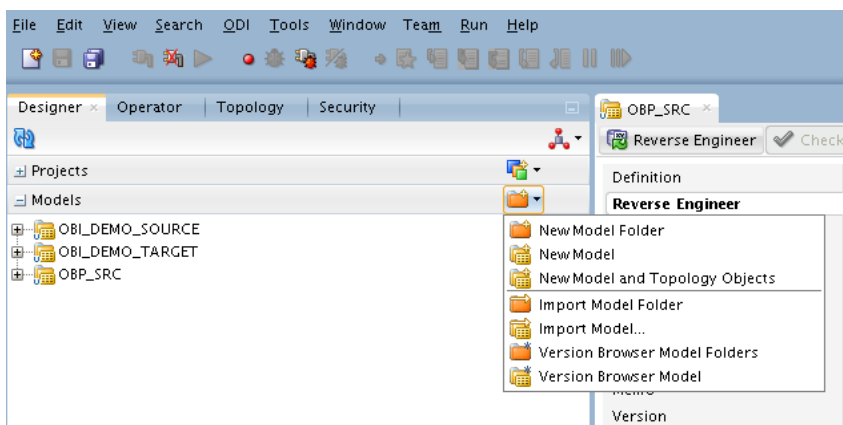
Step 4 Creating Model

To create a model:

1. Navigate to Designer tab -> Models.

Click the symbol on its upper right corner and select **New Model**.

Figure 6–13 New Model



2. In the **Definition** tab, name the source model as OBP_SRC. Target model name can be of your choice.

3. Select **Technology** as Oracle and select the corresponding **Logical Schema**.

Figure 6–14 OBP_SRC Model - Definition

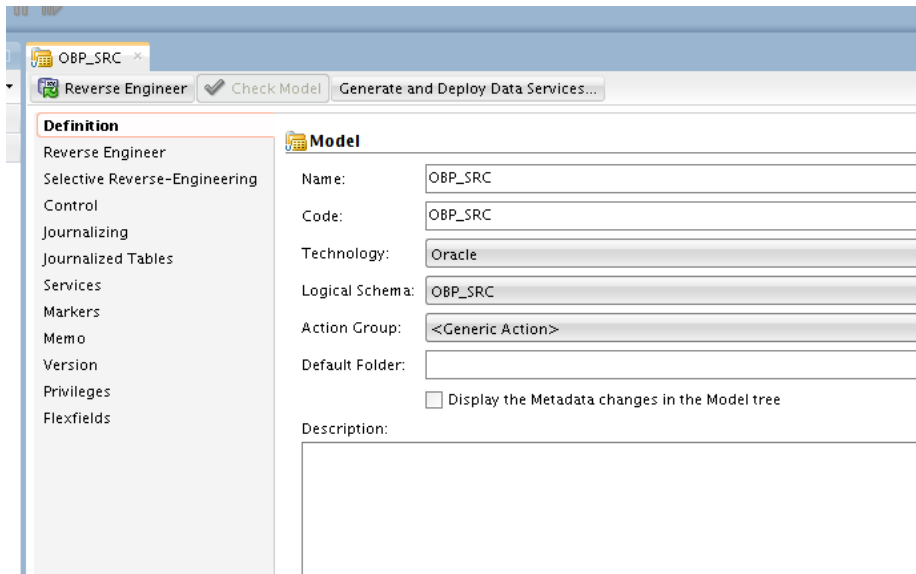
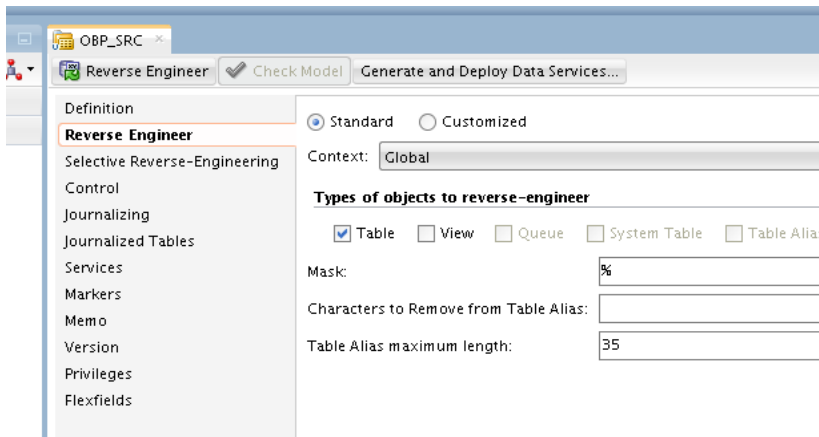


Figure 6–15 OBP_SRC Model - Reverse Engineer



4. Repeat the steps for the other model.

Figure 6–16 OBI_DEMO_TARGET Model - Definition

The screenshot shows the 'Definition' tab of the 'OBI_DEMO_TARGET' model configuration. The left sidebar lists various configuration options, with 'Reverse Engineer' selected. The main area displays the following fields:

- Name:** OBI_DEMO_TARGET
- Code:** OBI_DEMO_TARGET
- Technology:** Oracle
- Logical Schema:** L_OBI_DEMO_TARGET
- Action Group:** <Generic Action>
- Default Folder:** (empty)
- Display the Metadata changes in the Model tree
- Description:** (empty text area)

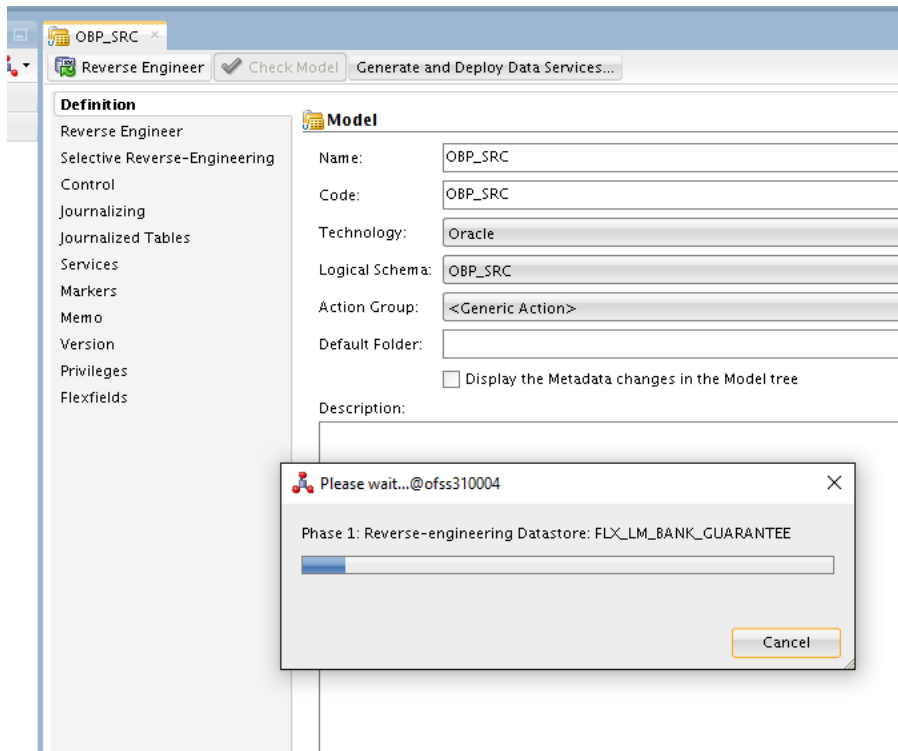
Figure 6–17 OBI_DEMO_TARGET Model - Reverse Engineer

The screenshot shows the 'Reverse Engineer' configuration window. The left sidebar lists various configuration options, with 'Reverse Engineer' selected. The main area displays the following fields:

- Standard** (selected) Customized
- Context:** Global
- Types of objects to reverse-engineer:**
 - Table View Queue System Table Table
- Mask:** %
- Characters to Remove from Table Alias:** (empty)
- Table Alias maximum length:** 35

5. In the **Reverse Engineer** tab, select the Standard option.
6. From the **Context** list select the value as Global.
7. In the Types of objects to reverse-engineer section, select the **Table** check box.
8. Click **Reverse Engineer**. This will take a few minutes.

Figure 6–18 Click Reverse Engineer



- Expand the model that you have generated. You can see all the tables of your DB under it.

Figure 6–19 OBP_SRC DB Tables

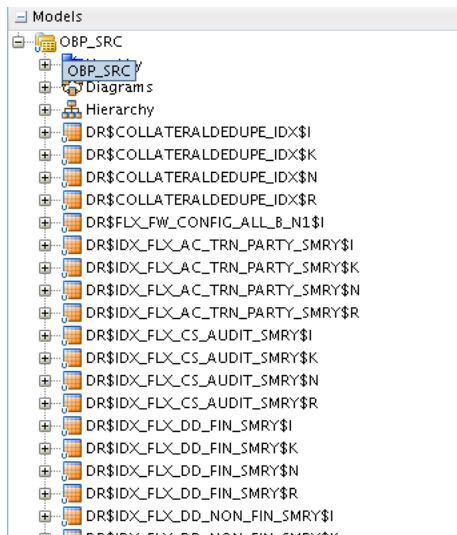
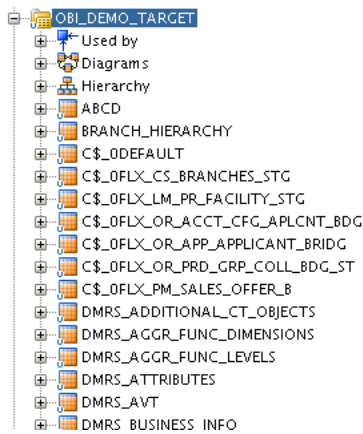


Figure 6–20 OBI_DEMO_TARGET DB Tables

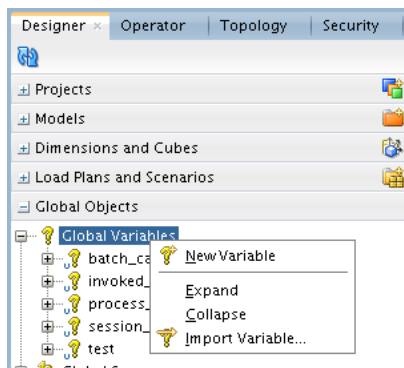


Step 5 Creating Global Variables

To create global variables:

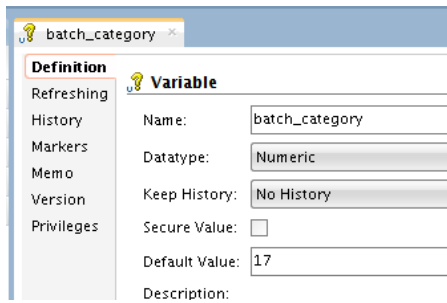
1. Navigate to Designer tab -> Global Objects.
2. Right-click on Global Variables and select **New Variable**.

Figure 6–21 New Variable

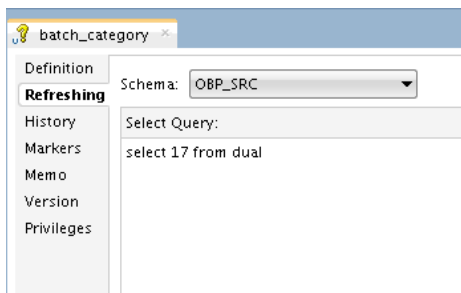


3. Enter the following values:

Name: batch_category
Datatype: Numeric
Keep History: No
Default Value: 17

Figure 6–22 Variable - Definition

- In the Refreshing tab, from the **Schema** list select the OBP_SRC schema.
In the Select Query section, enter "select 17 from dual".

Figure 6–23 Variable - Refreshing

- Create global variable invoked_scenario with the following values in the Definition tab:
Name: invoked_scenario
Datatype: Alphanumeric
Keep History: No

In the Refreshing tab, enter the following values:
Schema: OBP_SRC
Query: "SELECT '<%=odiRef.getPackage("PACKAGE_NAME")%>' from dual".
- Create global variable process_date with the following values in the Definition tab:
Name: process_date
Datatype: Alphanumeric
Keep History: No

In the Refreshing tab, enter the following values:
Schema: OBP_SRC
Query: "select '12345' from dual".
- Create global variable session_id with the following values in the Definition tab:

Name: session_id
Datatype: Numeric
Keep History: No

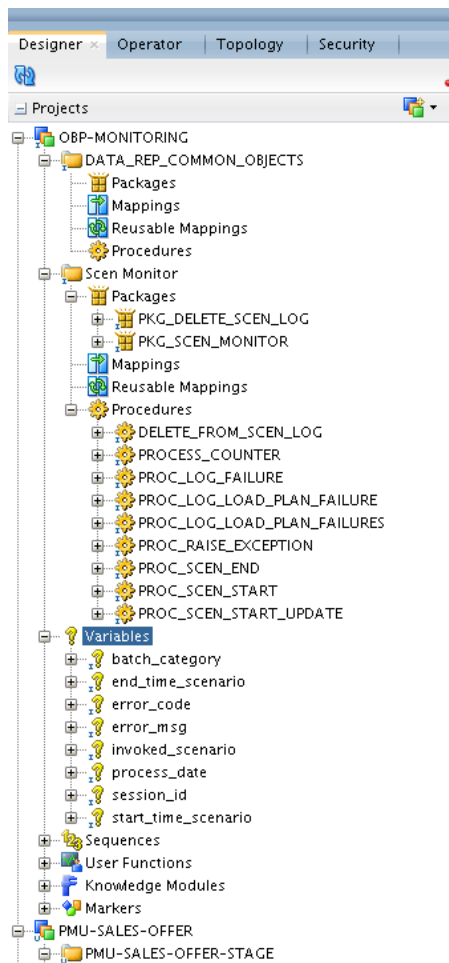
In the Refreshing tab, enter the following values:

Schema: OBP_SRC
Query: "SELECT <%=odiRef.getSession("SESS_NO")%> from dual".

Step 6 Projects Required in ODI

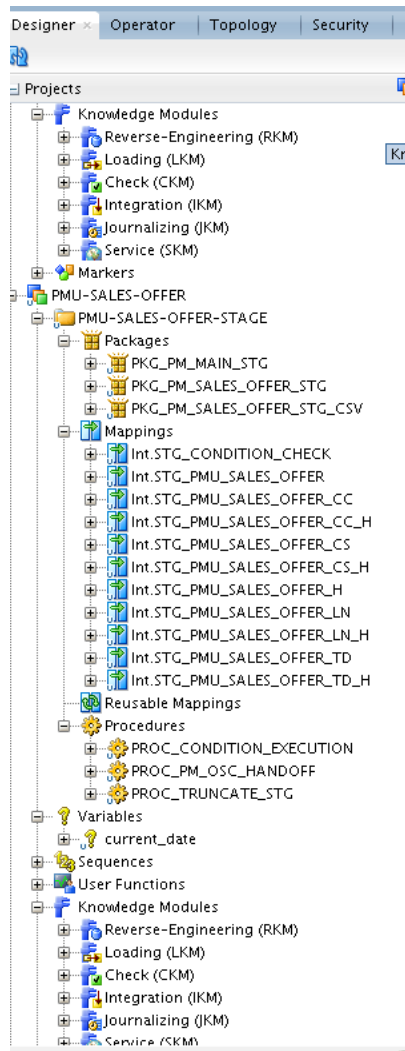
Modify the projects, scenarios, package, mappings of PMU-SALES-OFFER and OBP-MONITORING.

Figure 6–24 Project Settings



The package PKG_PM_SALES_OFFER_STAGE is the main package which is called from the OBP HOST and is configured in Day Zero Setup changes in OBP in source schema tables as given in step g.

Figure 6–25 PKG_PM_SALES_OFFER_STG



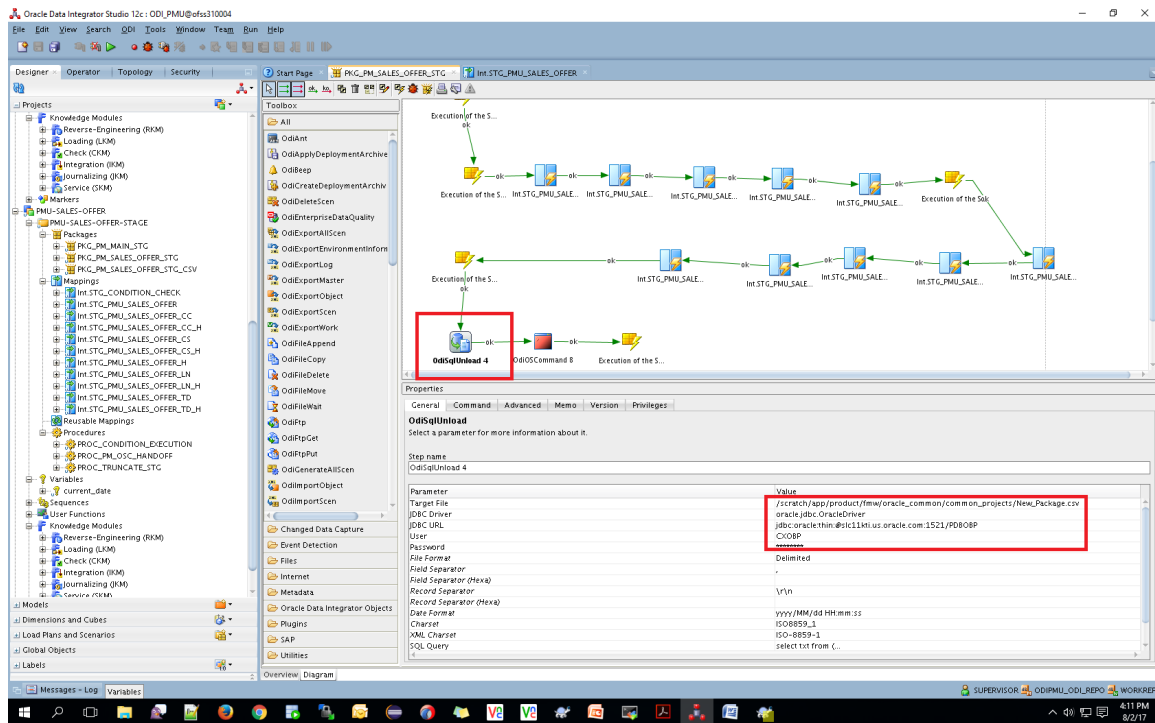
Step 7 CSV File Generation and Exporting it to Required Server

The procedure is as follows:

1. As seen in Figure 6–26, in the Properties section, in the General tab, the value mentioned in the **Target File** parameter is the path where the csv file New_Package.csv is generated.

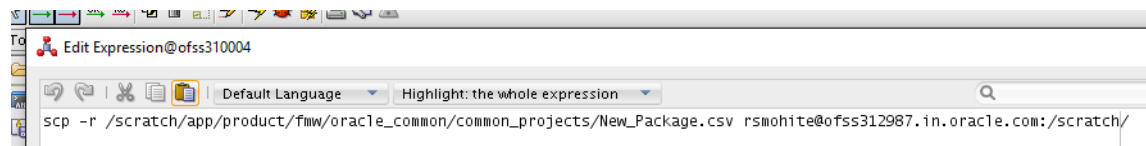
Similarly, you must also enter the jdbc and other related parameter details of your target schema.

Figure 6–26 PKG_PM_SALES_OFFER_STG Properties



2. As shown in Figure 6–27, the New_Package.csv file is being sent to ofss312987.in.oracle.com where rsmohite is the username of the machine in the director /scratch/.

Figure 6–27 Export the .csv File



7 Setting Up Product

This chapter describes the process of setting up the product. Products need to be configured manually from the Sales Offer Handoff file. The steps for manually adding Product in OCH are explained in this chapter.

7.1 Creating New Product

This section explains the procedure to set up Product for OBP as Asset or Liability.

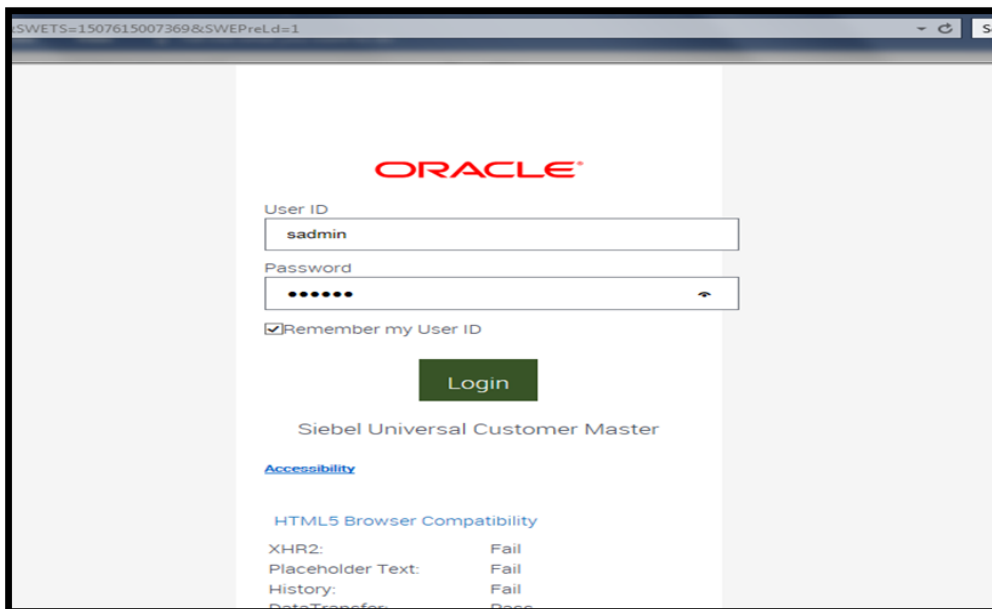
Following are the steps to configure new Product in OCH:

1. Log on to the UCM application (Siebel Universal Customer Master) with following credentials:

User ID: sadmin

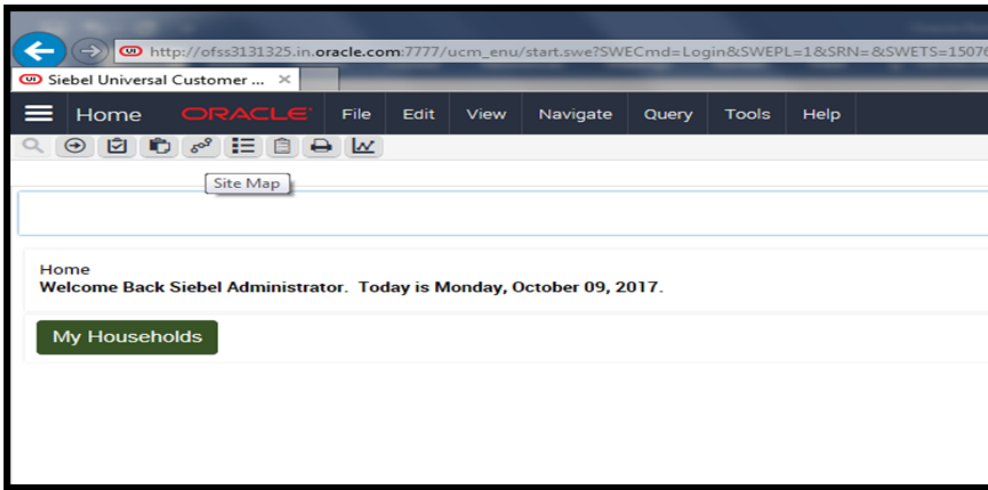
Password: sadmin

Figure 7–1 Log on to UCM



2. Navigate to Site Map.

Figure 7–2 Sitemap



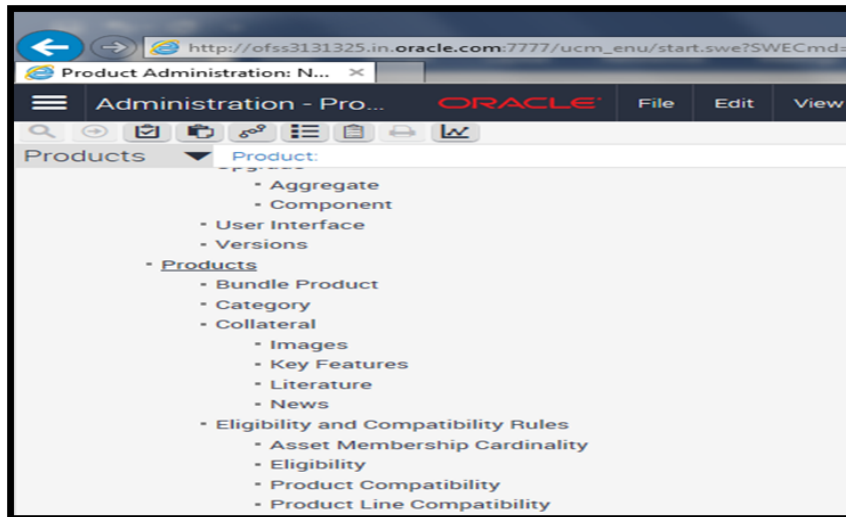
3. Click the **Administration - Product** link.

Figure 7–3 Administration - Product



4. Navigate to the Products view.

Figure 7–4 Products View

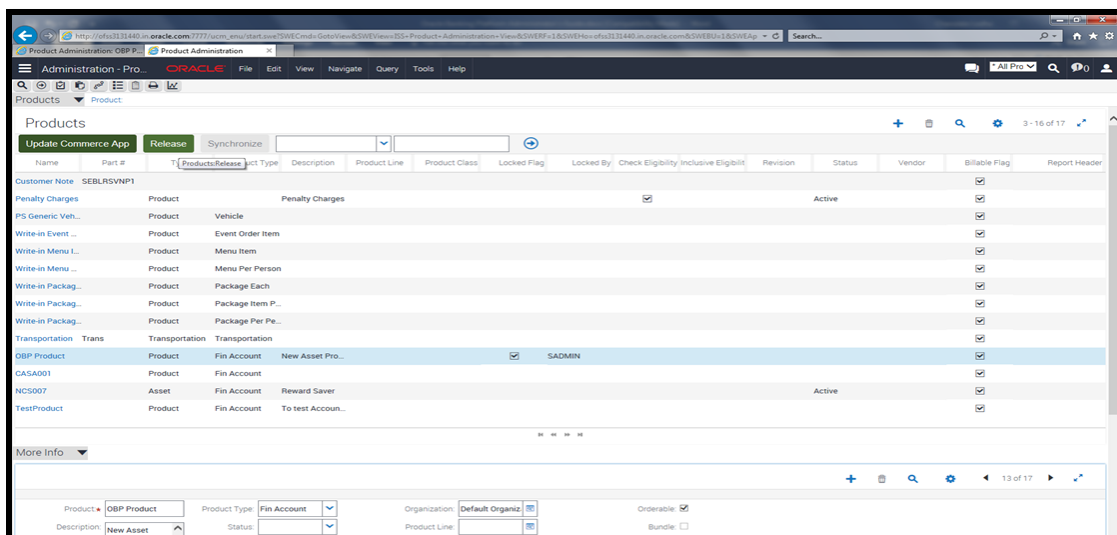


5. In the **Products** list, add a new record and complete the necessary fields.

Provide the following details:

- Name: Name of the Product. Eg: OBP Product
- Type: Product (Should be mentioned as is)
- Product Type: Fin Account (Should be mentioned as is)
- Description

Figure 7–5 Product List

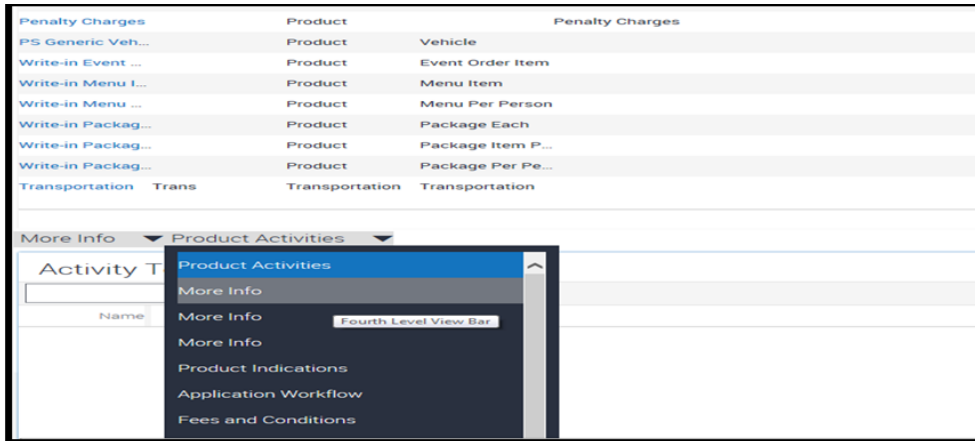


6. Step off the record. Press Ctrl + S to save the record.

7.1 Creating New Product

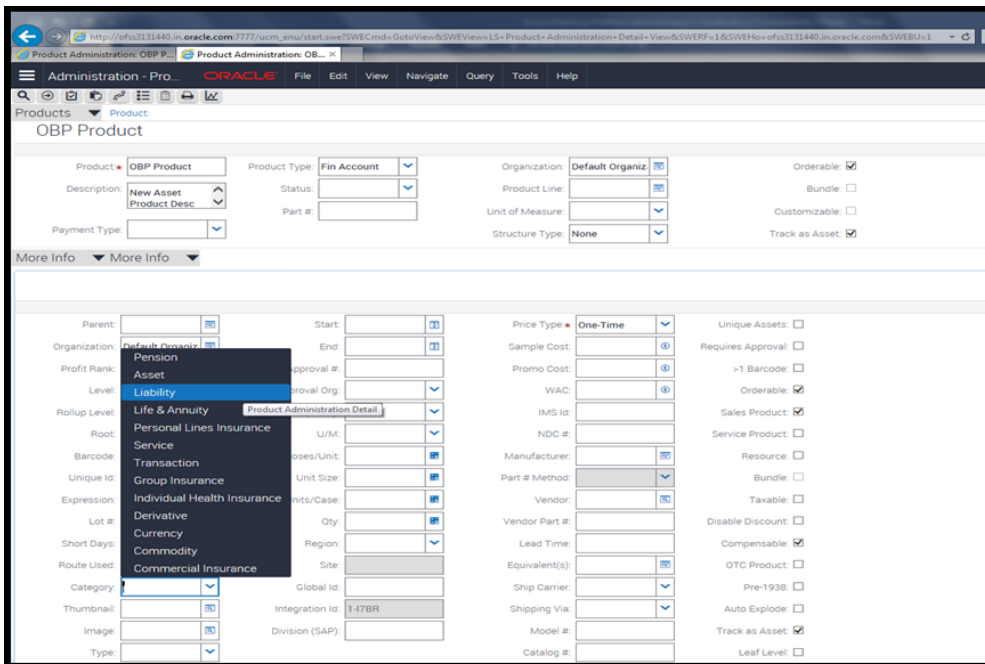
7. From the list, select the newly created product record. Click **Release**.
8. In the More Info Applet, navigate to the first occurrence of **More Info** in the list.
9. In the subsequent page, navigate to the first occurrence of **More Info** in the **Product Activities** list.

Figure 7–6 Product Activities



10. From the **Category** list select the value as Asset or Liability.

Figure 7–7 More Info - Select Category



11. From the **Type** list, select the appropriate value.

Figure 7–8 More Info - Select Type

The screenshot shows the Oracle Administration - Pro. interface for creating a new product. The main form is titled 'OBP Product' and includes the following fields and options:

- Product:** OBP Product
- Product Type:** Fin Account
- Organization:** Default Organiz
- Orderable:**
- Description:** New Asset
- Status:** [Dropdown]
- Product Line:** [Dropdown]
- Bundle:**
- Payment Type:** [Dropdown]
- Part #:** [Text]
- Unit of Measure:** [Dropdown]
- Customizable:**
- Structure Type:** None
- Track as Asset:**

The 'More Info' section is expanded, showing a list of product types on the left and various configuration options on the right:

- Product Type List (Left):** Parent, Organization, Profits Rank, Level, Rollup Level, Root, Barcode, Unique Id, Expression, Lot #, Short Days, Route Used, Category, Thumbnail, Image, Type. The 'Type' dropdown is open, showing options: Guarantee, **CASA**, Retail Deposit, DDA, Savings/MMA, CD, Investments, Credit Card, Asset Finance, Retirement, Facility, Trust Account.
- Configuration Options (Right):**
 - Start, End, Price Type (One-Time), Unique Assets
 - Approval #, Approval Org, Frequency, U/M, Doses/Unit, Unit Size, Units/Case, Qty, Region, Site, Global Id, Integration Id, Division (SAP)
 - Sample Cost, Promo Cost, WAC, IMS Id, NDC #, Manufacturer, Part # Method, Vendor, Vendor Part #, Lead Time, Equivalent(s), Ship Carrier, Shipping Via, Model #, Catalog #
 - Requires Approval, >1 Barcode, Orderable, Sales Product, Service Product, Resource, Bundle, Taxable, Disable Discount, Compensable, OTC Product, Pre-1938, Auto Exploze, Track as Asset, Leaf Level

12. Press Ctrl+S to save the record.

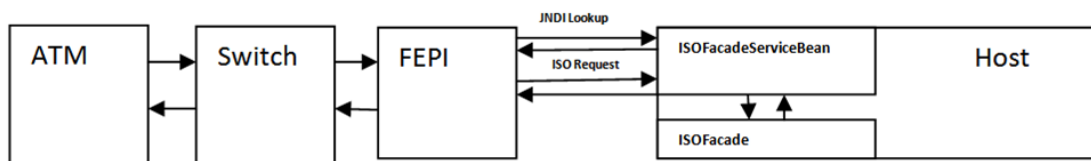
8 Setting Up Channels

This chapter describes the process of setting up the channels.

8.1 Channel Setup

This section explains the procedure to set up channels for OBP. The [Figure 8–1](#) gives an overview of the transaction message flows.

Figure 8–1 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 for **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 for **ATM or POS Terminal Setup (Fast Path: ATM002)**, see its context-sensitive help in the application.

8.2 Front End Processing Interface (FEPI)

This section contains information related to FEPI.

8.2.1 Installation

FEPI is installed in the same location as OBP (/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.

8.2.2 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.

8.2.3 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 Diagnosability details at http://docs.oracle.com/cd/E25054_01/doc.1111/e24473/diagnosability_adminuser.htm.

8.2.4 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

9 Application Monitoring Using Administration Application

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

9.1 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), and time metrics. 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.

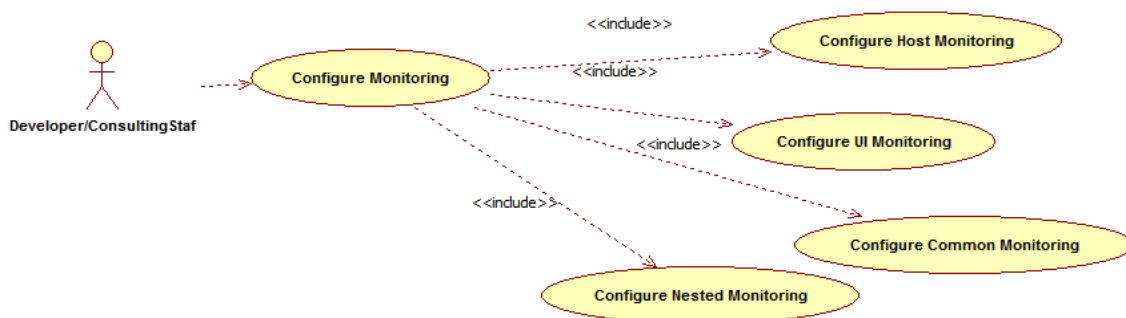
9.1.1 Usage

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.

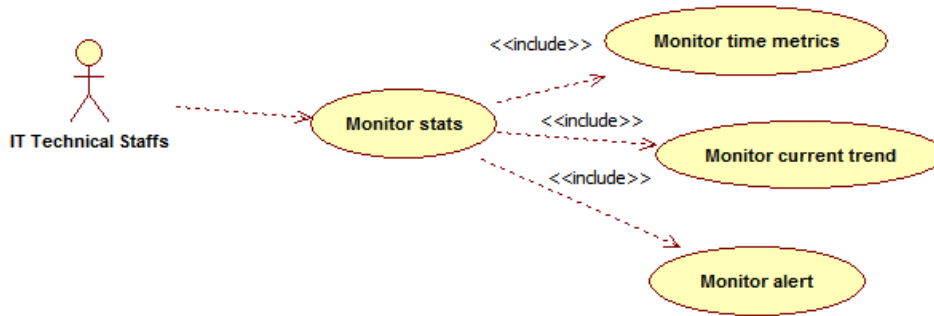
Figure 9–1 Developers



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' service could indicate need of optimization of the service.

Figure 9–2 IT Technical Staff



9.1.2 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.

9.1.2.1 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, and alert for the channel.

Figure 9–3 Monitoring Application Performance

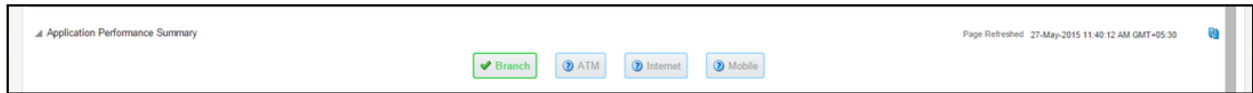
The screenshot shows the 'Monitor Application Performance' page. At the top, there are navigation buttons for 'Branch', 'ATM', 'Internet', and 'Mobile'. Below this is an 'Application Performance Summary' section with a 'Page Refreshed' timestamp. The main part of the page is a table with columns for Alert, Channel, Module, Layer, Transaction, Task Code, Trend, Alert Event Time, Trend Reference Queue, Last Alert User, Time in milliseconds (Average, Max, Min, Total), Transaction Count (Success, Failure), and Amount (Debit, Credit). The table lists several transactions with their respective metrics and alert statuses.

Alert	Channel	Module	Layer	Transaction	Task Code	Trend	Alert Event Time	Trend Reference Queue	Last Alert User	Time in milliseconds				Transaction Count		Amount	
										Average	Max	Min	Total	Success	Failure	Debit	Credit
	Branch	ORIGNATION	Spi	Perform Auto Decision	-		27-May-2015 11:39:37	4147, 5047, 3252, 3994, ...	arun	4,563	6,463	2,890	36,501	8	0	-	-
	Branch	TD	Backing Bean	Mixed Paym_ UI	TD002		27-May-2015 10:46:53	881, 936, 2143, 2616, 6816	-	4,155	19,078	861	58,167	14	0	-	-
	Branch	PARTY	Spi	Add Or Update Party Financial Profile	-		27-May-2015 11:39:21	6739, 2380, 1740, 758, 1, ...	arun	3,993	11,972	758	35,936	9	0	-	-
	Branch	CASA	Backing Bean	Alternate Accounts Save_ UI	CASA037		27-May-2015 10:39:16	465, 2720	-	1,593	2,720	465	3,185	2	0	-	-
	Branch	ACCOUNT	Spi	Recommend Bundles	VL000		27-May-2015 11:36:01	424, 901, 399, 1103, 1927	arun	1,528	10,281	306	56,535	37	0	-	-
	Branch	CONTENT	Spi	Deliver And Save Documents	OR247		27-May-2015 10:38:39	1308, 1359, 1420, 1303, ...	asavant	1,515	2,339	1,303	10,602	7	2	-	-
	Branch	ORIGNATION	Spi	Submit Create Offer	OR223		27-May-2015 10:40:22	1025, 1271, 1170, 1288, ...	asavant	1,362	2,006	972	9,537	7	0	-	-

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

9.1.2.1.1 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 and active channels. The Refresh Button on top of this section gets the latest (refreshed) metrics.

Figure 9–4 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

9.1.2.1.2 Log Level

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

Figure 9–5 Log Level

9.1.2.1.3 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, click the ALL button.

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 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 9–6 Alert State

The screenshot shows the 'Monitor Application Performance' window. At the top, there's a summary section with filters for Branch, ATM, Internet, and Mobile. Below that, the 'Application Performance' section is set to 'Transactions' for 'Branch'. The main table displays a list of alerts with columns for Alert, Channel, Module, Layer, Transaction, Task Code, Trend, Alert Event Time, Trend Reference Queue, Last Alert User, Time in milliseconds (Average, Max, Min, Total), Transaction Count (Success, Failure), Amount (Debit, Credit), Trend Reference, Nested Status, Alert EOD, and Service. The table contains 12 rows of data, each representing a different transaction type with its associated alert status and performance metrics.

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

Table 9–1 Alert State

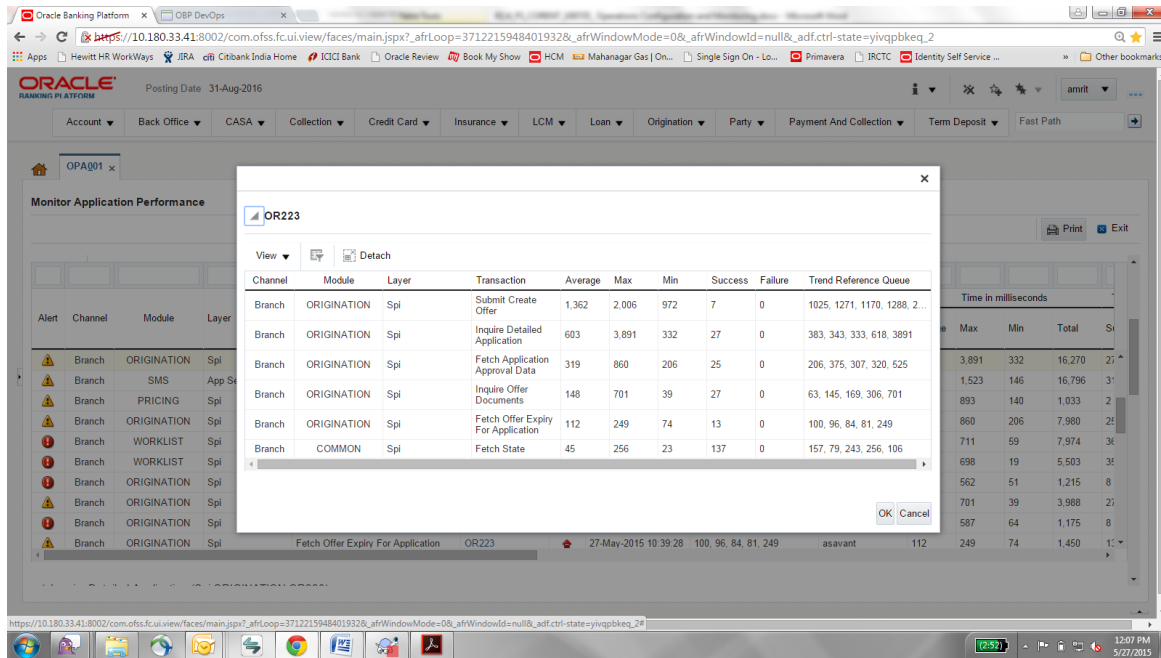
Sr. no.	Column Name	Description
1	Alert	Alert state of the transaction Valid Values: BLANK: No alert, Warning: Alert in past (default 5 minutes), Critical: Alerted Transaction
2	Channel	Channel through which the transaction occurred Valid Values: Branch, ATM, and POS.
3	Module	Application module of which transaction is a part
4	Layer	Configured Noun generation layer. Backing Bean for UI and Spi and App Service for Host.
5	Transaction	Name of the transaction
6	Task Code	Task code of the application page by which the transaction was triggered
7	Trend	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

Sr. no.	Column Name	Description
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	Nested Status	Nested Status
21	Alert ECID	ECID of the last alerted transaction
22	Service	Service name of the transaction
23	Completed Operations	Number of completed transactions
24	Active Threads	Active Threads
25	Max Active Threads	Maximum active threads
26	Host	Host name
27	Process	Process Name
28	Server Name	Server name
29	App Root Type	Root type of noun
30	Failure Security Event	Failure due to security error
31	2FA Event	Authentication error
32	Failure Database Event	Failure due to database error
33	Failure Technical Event	Failure due to technical error
34	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 task code only.

9.1 Dynamic Monitoring Service (DMS)

Figure 9–7 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 9–8 Selection of Desired Transaction

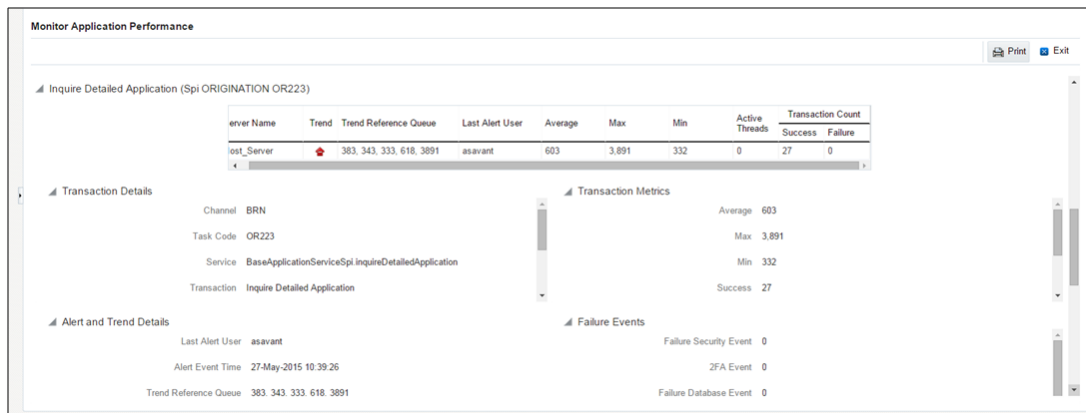


Figure 9–9 Transaction Details

Transaction Details	
Channel	BRN
Task Code	OR223
Service	BaseApplicationServiceSpi.inquireDetailedApplication
Transaction	Inquire Detailed Application
App Root Type	Transaction
Host	ofss3121059.in.oracle.com
Server Name	Host_Server
Process	obphost_server1:8001

Figure 9–10 Transaction Metrics

Transaction Metrics	
Average	603
Max	3,891
Min	332
Success	27
Success	27
Failure	0
Active Threads	0
Max Active Threads	1

Figure 9–11 Alert and Trend Details

Alert and Trend Details	
Last Alert User	asavant
Alert Event Time	27-May-2015 10:39:26
Trend Reference Queue	383, 343, 333, 618, 3891
Alert ECID	9d35654d4414a931:-6e0ab1f:14d8b6681e1:-8000-000000000000d612

Figure 9–12 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`.

9.2 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 metrics like duration, throughput, aborts, and so on.

9.2.1 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 9–13 Developers



IT Technical Staff

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

Figure 9–14 IT Technical Staff

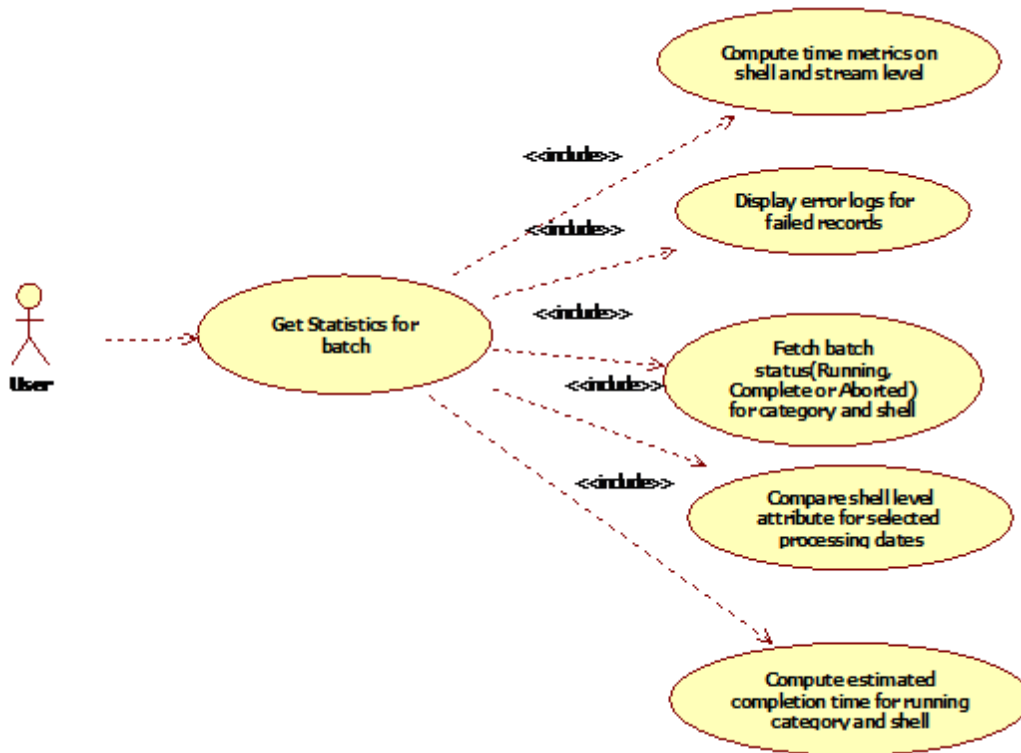


Figure 9–15 IT Technical Staff - Monitor Batch Stats



9.2.2 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.

9.2.2.1 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 9–16 Batch Performance Monitoring

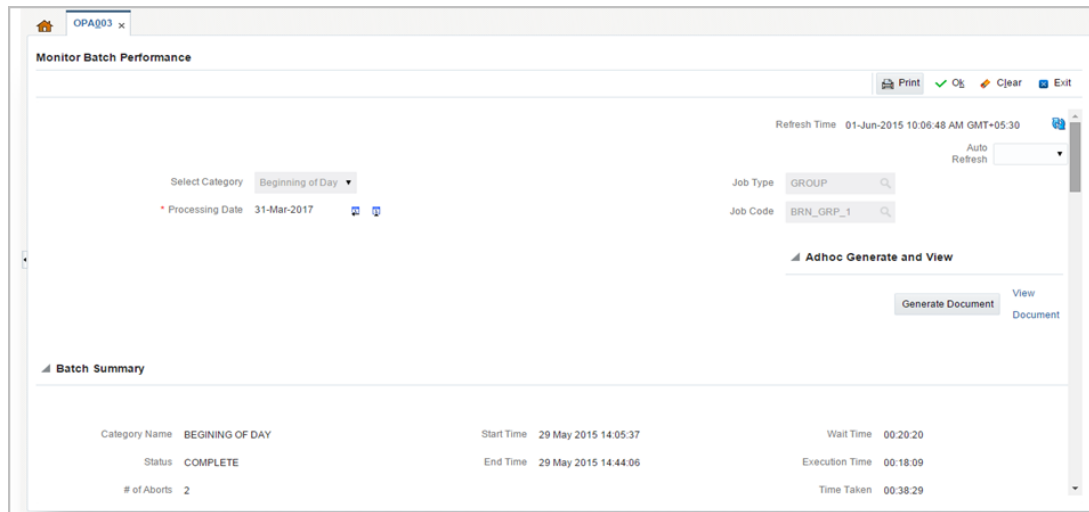
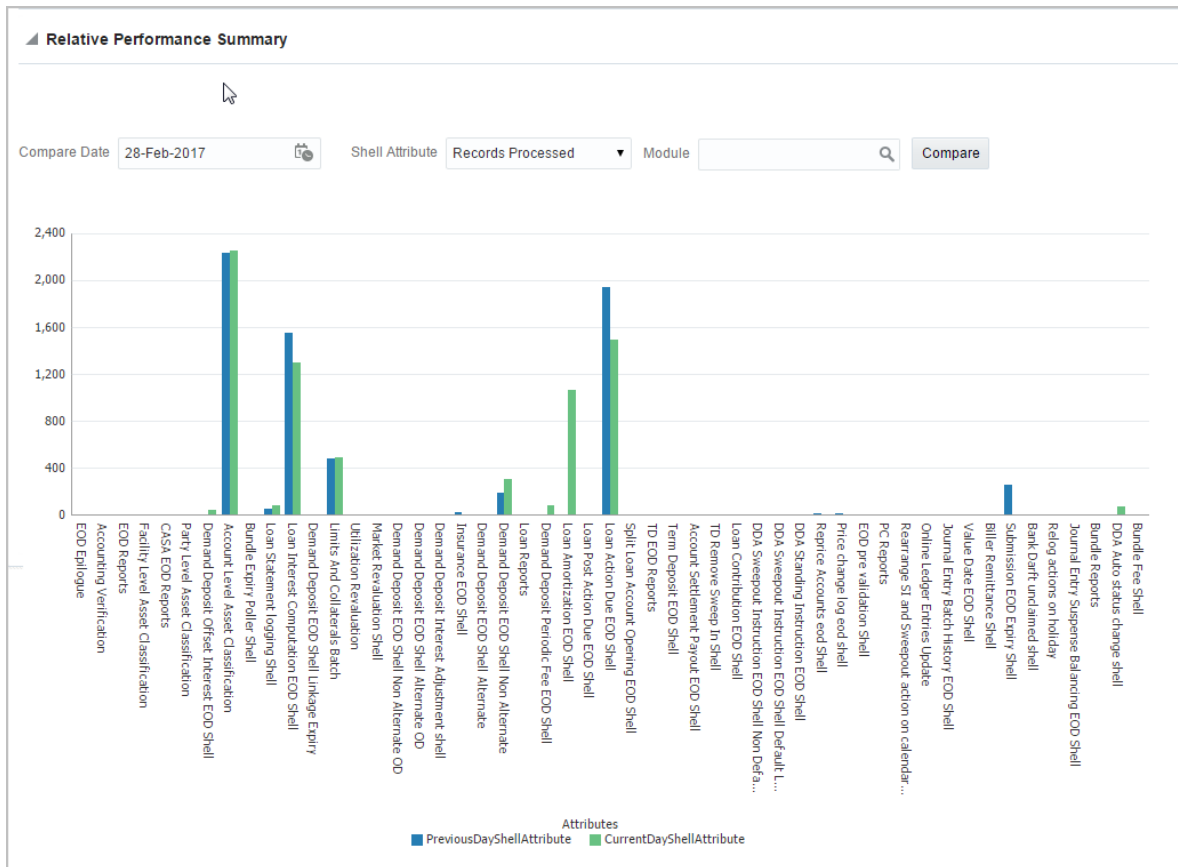


Figure 9–17 Batch Performance Monitoring - Shell Details

The screenshot shows the 'Shell Details' table with the following data:

Name of Shell	State	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
DDA Standing Instruction BOD Shell	Complete	↔	00:00:04		0	0		
DDA Sweepout Instruction BOD Shell Default L...	Complete	↔	00:00:00		0	0		
DDA Sweepout Instruction BOD Shell Non Defa...	Complete	↔	00:00:00		0	0		
Account Settlement Payout BOD Shell	Complete	↓	00:00:00		0	0		
Split Loan Account Opening BOD Shell	Complete	↓	00:00:20		7776	1		
Loan Action due BOD shell	Complete	↓	00:28:35	1	1180	508		
Loan Post Action due BOD shell	Complete	↔	00:00:02		0	0		
CASA BOD Reports	Complete	↔	00:00:00		0	0		

Figure 9–18 Batch Performance Monitoring - Relative Performance Summary



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 9–2 Category Details

Attribute Name	Description
Category Name	Name of category for example, EOD (End Of Day)
Status	Indicates status of selected category. Valid Values: COMPLETE, RUNNING, ABORTED.
Start Time	Indicates start time of category. The time is represented in DD-MM-YYYY hh:mm:ss format.
End Time	Indicates end time of category. The time is represented in DD-MM-YYYY hh:mm:ss format.

Attribute Name	Description
Wait Time	It is the time for which category is in Aborted state. Wait time for category denotes the time for which batch was halted.
Execution Time	It the time for which category is in Running state.
Number Of Aborts	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.
Estimated Completion Time	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
Time Status	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 9–3 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.
Start Time	Indicates start time of shell. The time is represented in DD-MM-YYYY hh:mm:ss format.
End Time	Indicates the time at which shell is completed. The time is represented in DD-MM-YYYY hh:mm:ss format.
Wait Time	It is the time for which shell is in aborted state.
Expected Completion Time	Indicates the estimated time for a shell to complete.
Failed Records	Number of failed records for a shell
Records Processed	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.

Attribute Name	Description
Throughput	It is the average processing time for one record. Throughput is denoted in millisecs.

Figure 9–19 Shell Details

The screenshot displays the 'Monitor Batch Performance' interface. Under the 'Shell Details' section, a table lists several shells. The 'Untanking Adjustment Posting' shell is selected, showing a throughput of 29 and 4 total records. Below the table, detailed statistics for this shell are provided:

Module Code	AS	Wait Time		Pending Time	00:00:00
Number of Streams	1	No of Aborts		CommentCount	0
Start Time	2015-05-06 18:28:01	Records Processed	4	Failed Records	0
End Time	2015-05-06 18:28:21				

Figure 9–20 Shell Details - DDA Standing Instructions

The screenshot shows the 'DDA Standing Instruction BOD Shell Details' interface. It displays the following statistics for the selected shell:

Module Code	DD	Wait Time		Pending Time	00:00:00
Number of Streams	1	No of Aborts		commentCount	0
Start Time	2015-05-06 18:28:45	Records Processed		Failed Records	0
End Time	2015-05-06 18:28:45				

Below the statistics, there is a section for 'DDA Standing Instruction BOD Shell Notes' with an 'Add New Note' button and a message 'No items to display'. At the bottom, there are options to 'Clear All Filters', 'Export To Excel', and 'Detach'.

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.

The following figure shows the view displayed during batch run. Few extra parameters like estimated completion time for shell and category are monitored during batch run.

Figure 9–21 View of Batch Run

Name of Shell	State	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
DDA Standing Instruction BOD Shell	Complete	↔	00:00:03		0	0	0% 100%	
DDA Sweepout Instruction BOD Shell Default L...	Complete	↔	00:00:00		0	0	0% 100%	
DDA Sweepout Instruction BOD Shell Non Defa...	Complete	↔	00:00:00		0	0	0% 100%	
Account Settlement Payout BOD Shell	Complete	↓	00:00:21		238	1	0% 100%	
Split Loan Account Opening BOD Shell	Complete	↑	00:00:20		11,399	1	0% 100%	
Loan Action due BOD shell	In Progress	↓	00:17:02		1,280.85	859	0% 100%	02-Jun-2015 14:38:00
Loan Post Action due BOD shell	Not Started	↔	00:00:00		0	0	0% 100%	

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 9–4 Stream Details

Attribute Name	Description
Stream Number	Indicates the number of a stream in which the record is being processed
First Row	Indicates the start sequence number of a record, processing in a particular stream.
Last Row	Indicates the end sequence number of a record, processing in a particular stream.
Duration	It is the time required for stream to complete.
Status	Valid Values: COMPLETED, RUNNING. It indicates the status of selected stream
Processed Count	Number of records processed in a stream
Server Name	Name of a server running the stream

Figure 9–22 Stream Based Shells

Loan Action due BOD shell Notes

Created By: bhaktim (USER) on 27/05/2015 at 14:38:45
 Subject: patch applied
 Comment: patch applied
 Internal to Bank:
 Stage: 2
 Reply: 0

Loan Action due BOD shell Stream Details

Clear All Filters
 View: [Dropdown] Export To Excel [Icon] Detach [Icon]

Stream Number	First Row	Last Row	Current Row	Duration	Status	Processed Count	Failed Count	Server Name
1	1	1,269	1,270	1,681	COMPLETED	1,266	0	obphost_server1

Loan Action due BOD shell Error Desc

ErrorCode	ProcessResult	BranchCode	BranchGroupCode	RunCount	ErrorDesc	SummaryText
2	2	1010	BRN_GRP_1	6	com.ofss.fc.fram...	300340757 201...
2	2	1010	BRN_GRP_1	6	com.ofss.fc.fram...	300332595 201...
2	2	1010	BRN_GRP_1	6	com.ofss.fc.fram...	300230097 201...

Note

Shells are categorized into two types that is, Stream based shells and Report based shells. Figure 9–22 displays the 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 analyze and know the reason behind the failure of that particular record.

Figure 9–23 Exception Log

ORACLE BANKING PLATFORM
 Posting Date: 28-Feb-2017
 Account ▾ Back Office ▾ CASA ▾ Collection ▾ Credit Card ▾ Insurance ▾ LCM ▾ Loan ▾ Origination ▾ Party ▾ Payment And Collection ▾ Term Deposit ▾ Fast Path ▾

OPA003 x

Monitor Batch Performance

Print [Icon] OK [Icon] Clear [Icon] Exit [Icon]

```

com.ofss.fc.framework.batch.exception.BatchFrameworkException: An error occurred in batch process. at com.ofss.fc.framework.batch.process.BatchProcess.execute(BatchProcess.java:950)
at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.executeBatch(RecoverableBatchProcess.java:458) at com.ofss.fc.framework.batch.process.RecoverableBatchProcess.processBatch(RecoverableBatchProcess.java:217) at
com.ofss.fc.framework.batch.process.BatchProcess.startBatchProcess(java:507) at com.ofss.fc.batch.StreamProcessHelper.requestBatchProcess(StreamProcessHelper.java:75) at com.ofss.fc.batch.StreamProcessHelper.processRequest(StreamProcessHelper.java:49) at
com.ofss.fc.batch.mdb.StreamListenerMDB.onMessage(StreamListenerMDB.java:91) Caused by: java.lang.reflect.InvocationTargetException at com.ofss.fc.framework.batch.process.BatchProcess.execute(BatchProcess.java:918) ... 6 more Caused by:
com.ofss.fc.framework.exception.BusinessException: The ledger was not found for the account role LN_INSTRANCE_PAYABLE and LPLUN. at
com.ofss.fc.domain.accounting.da.entity.transactionentry.AccountingTransactionContainerFactory.fetchDerivedQLForAliasOrFactBased(AccountingTransactionContainerFactory.java:618) at
com.ofss.fc.framework.batch.exception.BatchActionProcessingException: Exception occurred while executing Action(s). at com.ofss.fc.domain.accounting.service.action.executor.ActionSetProcessor.processActionSet(ActionSetProcessor.java:191) ... 7 more Caused by:
com.ofss.fc.domain.accounting.da.entity.transactionentry.AccountingTransactionContainerFactory.createInstance(AccountingTransactionContainerFactory.java:222) at
com.ofss.fc.domain.accounting.da.service.AccountingTemplateFetcher.generateAccountingContainerFromTemplate(AccountingTemplateFetcher.java:34) at com.ofss.fc.domain.accounting.da.service.AccountingEventService.processAccountingEvent(AccountingEventService.java:560) at
com.ofss.fc.app.accounting.service.da.AccountingEventApplicationService.raiseAndProcessAccountingEventBatchMode(AccountingEventApplicationService.java:1709) at com.ofss.fc.app.adapter.sml.LoanAccountingAdapter.raiseAccountingEvent(LoanAccountingAdapter.java:53) at
com.ofss.fc.domain.accounting.service.LoanAccountingService.raiseAccountingEvent(LoanAccountingService.java:105) at com.ofss.fc.app.loan.insurance.LoanInsuranceApplication.raiseAccounting(LoanInsuranceApplication.java:919) at
com.ofss.fc.app.loan.insurance.LoanInsuranceApplication.updateAndRaiseAccountingForNewPremium(LoanInsuranceApplication.java:4043) at com.ofss.fc.app.loan.insurance.LoanInsuranceApplication.updateInsuranceBalancesAndRaiseAccounting(LoanInsuranceApplication.java:3844)
at com.ofss.fc.app.loan.insurance.LoanInsuranceApplication.processChangeInInsuranceAmountOnReviewDate(LoanInsuranceApplication.java:4341) at com.ofss.fc.app.loan.account.CCRReviewApplication.processCCRReviewApplication(java:70) at
com.ofss.fc.domain.action.executor.LoanActionExecutor.executeAction(LoanActionExecutor.java:228) at com.ofss.fc.domain.action.executor.LoanActionExecutor.executeAction(LoanActionExecutor.java:71) at
com.ofss.fc.domain.account.service.action.executor.ActionSetProcessor.processActionSet(ActionSetProcessor.java:184) ... 7 more
    
```

OK Cancel

Relative Performance Summary

T14_BUILD_DATE=2015-05-26 TNS Details: PDBT14 = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)HOST = 0BPDB-RAC-CLUSTER-SCAN.in.oracle.com)PORT = 1521)):(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = PDBT14)) , DB Details: T14:d114id @ PDBT14 Host IP: 10.180.4.125
 Copyright © Oracle Financial Services Software Limited. All rights reserved.

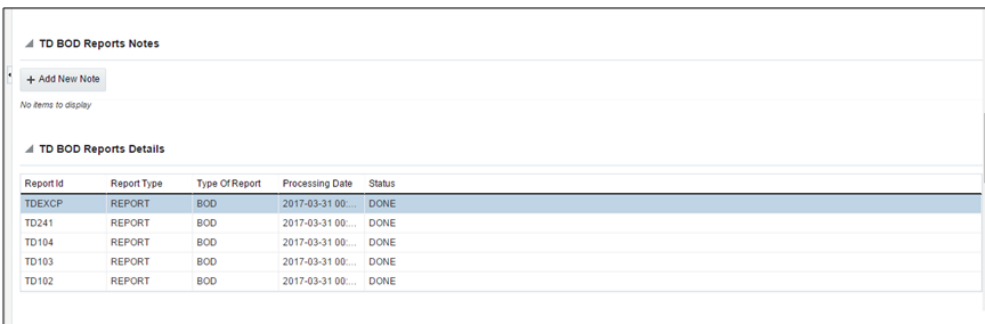
Reports Table

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

Table 9–5 Reports Table

Attribute Name	Description
Report Id	ID to uniquely identify report
Report Type	Report
Type Of Report	Indicates type of reports. Reports are classified based on category.
Processing Date	Indicates processing date of report.
Status	Indicates the status of the report. Valid Values: DONE, PENDING, RUNNING, ABORTED.
Error Message	Error message represents the reason for report failure. No message is displayed in case of successful run.

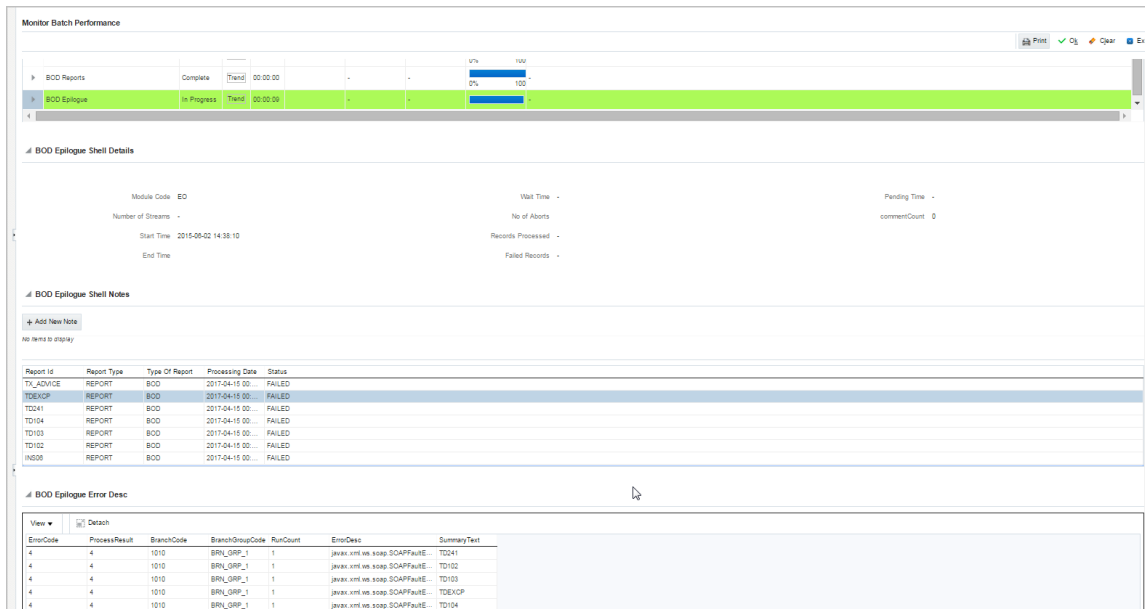
Figure 9–24 Report Based Shells



Report Id	Report Type	Type Of Report	Processing Date	Status
TDEXCP	REPORT	BOD	2017-03-31 00:...	DONE
TD241	REPORT	BOD	2017-03-31 00:...	DONE
TD104	REPORT	BOD	2017-03-31 00:...	DONE
TD103	REPORT	BOD	2017-03-31 00:...	DONE
TD102	REPORT	BOD	2017-03-31 00:...	DONE

The status of report based shell during batch run is shown in [Figure 9–25](#):

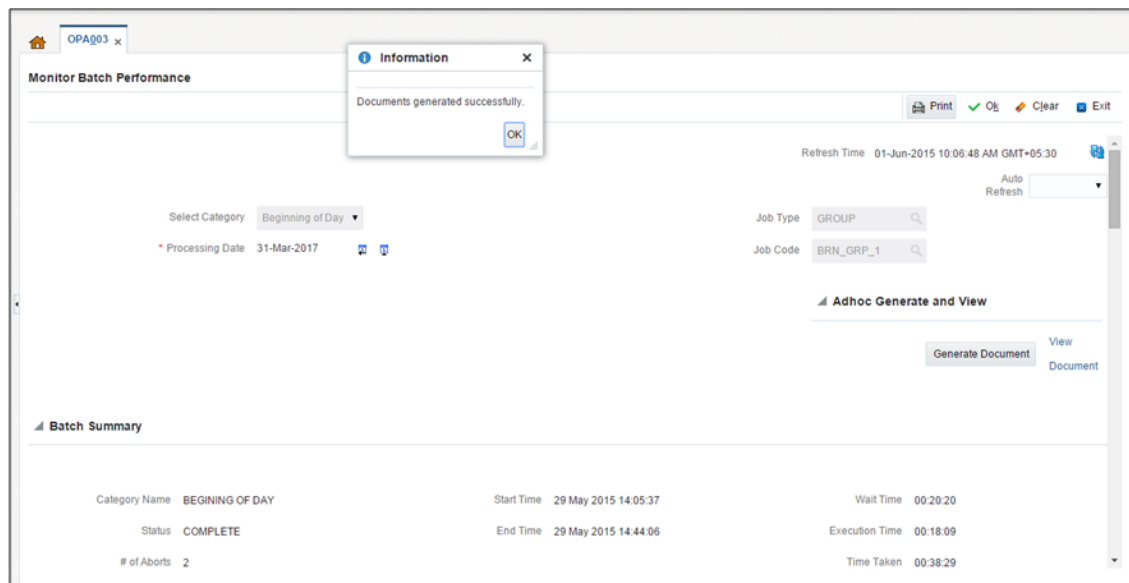
Figure 9–25 Status of Report Based Shell



Exception Report

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

Figure 9–26 Exception Report



The different parameter monitored at shell level and exception logs for all aborted shells are part of exception report. Figure 9–27 displays sample report for a particular shell.

Figure 9–27 Sample Report

BATCH EXCEPTION REPORT				
Bank : 10 EMERALD_BU			Job Type : GROUP	
Branch : 1010			Job Code: BRN_GRP_1	
Op. ID :			Report Date : 15-04-2017	
BOD Epilogue				
Start Time :	2-Jun-2015 9:08 AM	End Time :	2-Jun-2015 9:13 AM	
Records Skipped :		Duration :	00:05:18	
Number of Aborts :	1	Wait Time :	00:05:08	
Abort Statistics				
Abort Time	Restart Time	Abort Duration		
2-Jun-2015 9:08 AM	2-Jun-2015 9:13 AM	00:05:08		
Exception Summary				
Report Name	Report ID	Module Code	Error Code	Error Description
Maturities Due Report	TD102	TD	Time Of Last Update : 2015-06-02 14:38:11.411 javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.	javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.
Account/ Deposit Exceptions	TDEXCP	TD	Time Of Last Update : 2015-06-02 14:38:11.374 javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.	javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.
Matured Deposits with No Instructions	TD103	TD	Time Of Last Update : 2015-06-02 14:38:11.347 javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.	javax.xml.ws.soap.SOAPFaultException: oracle.xdo.webservice.exception.AccessDeniedException: java.lang.SecurityException: Failed to log into BI Publisher: invalid username or password.

Exception Log Table

The figure below provides the details of the exception log.

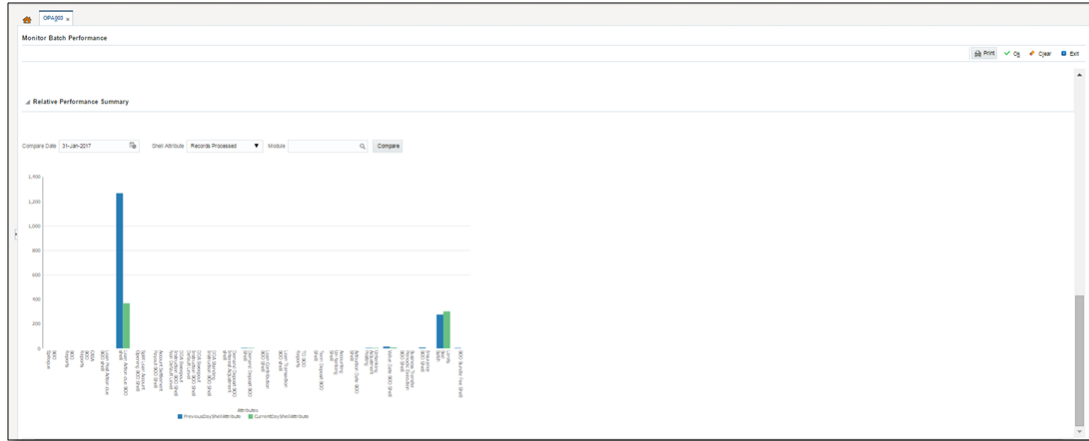
Figure 9–28 Exception Log Table

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 :862-8182-2552-7227 Error msg :An error occurred in batch process. Error cause :com.ofss.fc.framework.batch.exception.BatchFrameworkkException: 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 :862-8206-5767-2044 Error msg :An error occurred in batch process. Error cause :com.ofss.fc.framework.batch.exception.BatchFrameworkkException: 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:)	

9.2.3 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 9–29 Graphs



9.3 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 9–30 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 9–30 Input Parameters for Batch Handoff

The screenshot displays the 'Monitor Batch Performance' interface. At the top, there are navigation and utility buttons (Print, OK, Clear, Exit) and a refresh time indicator. The main area is divided into sections: 'Batch Summary' and 'Shell Details'. The 'Batch Summary' section provides key metrics for the 'Analytics Batch Data Hand-off' job, such as its status (ABORTED), start time (28 May 2015 16:11:19), and elapsed time (139:01:15). The 'Shell Details' section contains a table with the following data:

Name of Shell	State	Trend	Duration	No of Aborts	Throughput	Total Records	Status	Expected Completion Time
Analytics Hand-off	Complete	Trend	00:00:00	-	-	-	0% 100'	-
Analytics Data Hand-off	Complete	Trend	00:01:07	-	-	-	0% 100'	-
Analytics epilog	Aborted	Trend	139:00:08	12	-	-	-	-

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 9–31 Execution Unit

Analytics Data Hand-Off Details

Module Code DI Wait Time - Pending Time -
 Number of Streams - No of Aborts commentCount 0
 Start Time 2015-05-28 16:11:19 Records Processed -
 End Time 2015-05-28 16:12:26 Failed Records -

Analytics Data Hand-Off Notes

+ Add New Note

Execution Unit	Start Time	End Time	No Of Aborts	Duration	Service Provider	Execution Status	Records Processed
PKG_LN_CONTRACT_INTERFACES	28-May-2015 16:12:27	28-May-2015 16:13:34	0	00:01:07	ODI_SERVICE_PROVIDER	C	
PKG_ACCOUNT_RATE_TIERS	28-May-2015 16:12:26	28-May-2015 16:12:41	0	00:00:15	ODI_SERVICE_PROVIDER	C	
PKG_AS_ACCOUNT_ENTRY	28-May-2015 16:12:26	28-May-2015 18:05:03	1	01:52:37	ODI_SERVICE_PROVIDER	C	
PKG_TD_INTERFACES	28-May-2015 16:12:23	28-May-2015 18:34:17	1	02:21:54	ODI_SERVICE_PROVIDER	C	
PKG_PM_INTERFACES	28-May-2015 16:12:13	28-May-2015 16:12:32	0	00:00:19	ODI_SERVICE_PROVIDER	C	
PKG_PARTY_FIN_INTERFACES	28-May-2015 16:11:58	28-May-2015 16:12:27	0	00:00:29	ODI_SERVICE_PROVIDER	C	
PKG_PL_INTERFACES	28-May-2015 16:11:55	28-May-2015 16:12:38	0	00:00:43	ODI_SERVICE_PROVIDER	C	
PKG_OR_INTERFACES	28-May-2015 16:11:53	28-May-2015 16:12:49	0	00:00:56	ODI_SERVICE_PROVIDER	C	
PKG_MITIGANT_INTERFACES	28-May-2015 16:11:51	28-May-2015 16:12:40	0	00:00:49	ODI_SERVICE_PROVIDER	C	
PKG_LN_INTERFACES	28-May-2015 16:11:50	28-May-2015 16:12:39	0	00:00:49	ODI_SERVICE_PROVIDER	C	

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 9–32 Abort Statistics

Monitor Batch Performance

Print Refresh Clear Exit

Execution Unit	Start Time	End Time	No Of Aborts	Duration	Service Provider	Execution Status	Records Processed
PKG_TD_INTE...	28-May-2015 1...	28-May-2015 1...	1	02:21:54	ODI_SERVICE...	C	
PKG_PM_INTE...	28-May-2015 1...	28-May-2015 1...	0	00:00:19	ODI_SERVICE...	C	
PKG_PARTY_F...	28-May-2015 1...	28-May-2015 1...	0	00:00:29	ODI_SERVICE...	C	
PKG_PL_INTER...	28-May-2015 1...	28-May-2015 1...	0	00:00:43	ODI_SERVICE...	C	
PKG_OR_INTE...	28-May-2015 1...	28-May-2015 1...	0	00:00:56	ODI_SERVICE...	C	
PKG_MITIGAN...	28-May-2015 1...	28-May-2015 1...	0	00:00:49	ODI_SERVICE...	C	
PKG_LN_INTE...	28-May-2015 1...	28-May-2015 1...	0	00:00:49	ODI_SERVICE...	C	
PKG_GL_INTE...	28-May-2015 1...	28-May-2015 1...	0	00:00:11	ODI_SERVICE...	C	
PKG_FACILITY...	28-May-2015 1...	28-May-2015 1...	0	00:00:41	ODI_SERVICE...	C	
PKG_DDA_INT...	28-May-2015 1...	28-May-2015 1...	1	01:54:50	ODI_SERVICE...	C	

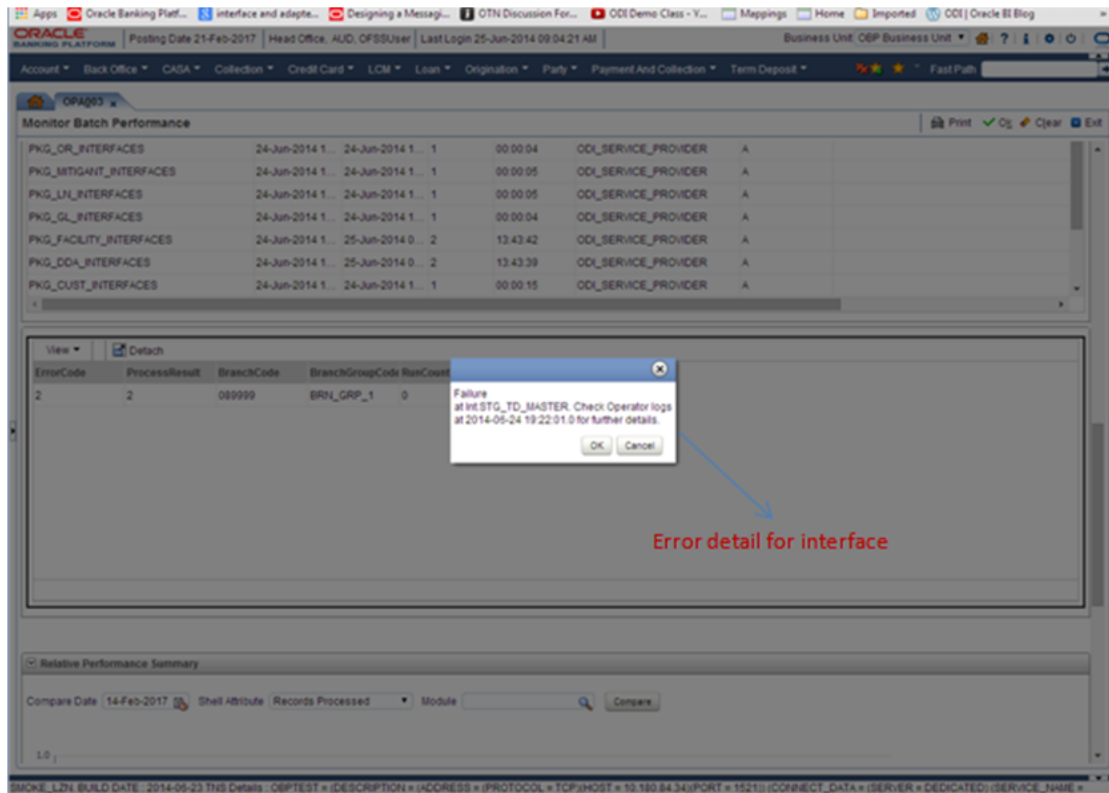
View Detach

ErrorCode	ProcessResult	BranchCode	BranchGroupCode	RunCount	ErrorDesc	SummaryText
2	2	1010	BRN_GRP_1	0	Failure at INTSTG_TD_MAS...	PKG_TD_INTERFACES

9.3 ODI Batch Handoff Monitoring

On click of the error description table, the error description message appears as shown in Figure 9–33:

Figure 9–33 Failure Error Description



10 Application Monitoring Using EM Plugin

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

10.1 Monitoring Application Using 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 analyze the behavior. The monitoring activities are mainly carried out by IT Technical staff.

10.1.1 Oracle Enterprise Manager (EM)







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

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

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

Some notations in EM are provided below:

Table 10–1 Notations in EM

	Indicates component is down
	Indicates component is up and running
	Indicates alerts
	Indicates warnings
	Indicates metric collection error
	Indicates healthy status

The following figure shows the environment view in Oracle Enterprise Manager:

Figure 10–1 Oracle Enterprise Manager

Name	Type	Status	Availability	Service Level Agreement Status	Incidents		System	Key Components				Key Tests							
					Permanen	Usage		Status	Incidents	Status	Monitoring Beacons								
OBP_SMOKEPROD261_SOA_Service	Generic Service	↑	Tests	-	-	-	-	-	-	-	-	n/a	0	0	0	0	↑1	1	
OBP_SMOKEPROD261_HOST_Service	Generic Service	↑	Tests	-	-	-	-	-	-	-	-	n/a	0	0	0	0	↑1	1	
OBP_SMOKEPROD261_Monitoring_Se...	Generic Service	↑	System	-	-	-	-	-	-	-	-	OBP_SMOKEPROD261_Monitoring_System	↑1	0	0	0	0	n/a	0
OBP_SMOKEPROD261_UI_Service	Generic Service	↑	Tests	-	-	-	-	-	-	-	-	/SMOKEPROD261_UI_ofss310490_in_oracle_co... /UI_domain	n/a	0	0	0	0	↑1	1
OBP_SMOKEPROD261_OD_Service	Generic Service	↑	Tests	-	-	-	-	-	-	-	-	/SMOKEPROD261_OD_ofss3121155_in_oracle_co... /ODDomain	n/a	0	0	0	0	↑1	1
OBP_SMOKEPROD261_View	Aggregate Service	↑	Sub Services	-	-	-	-	-	-	-	-	n/a	↑5	0	0	0	0	n/a	0

The views in the above figure include UI, Host, and SOA servers.

Security Stacks components such as OAAM, OID, OES, outbound components such as Oracle Analytics Publisher (formerly known as Business Intelligence Publisher), IPM, Documaker, ATM and POS channels are also part of the environment view.

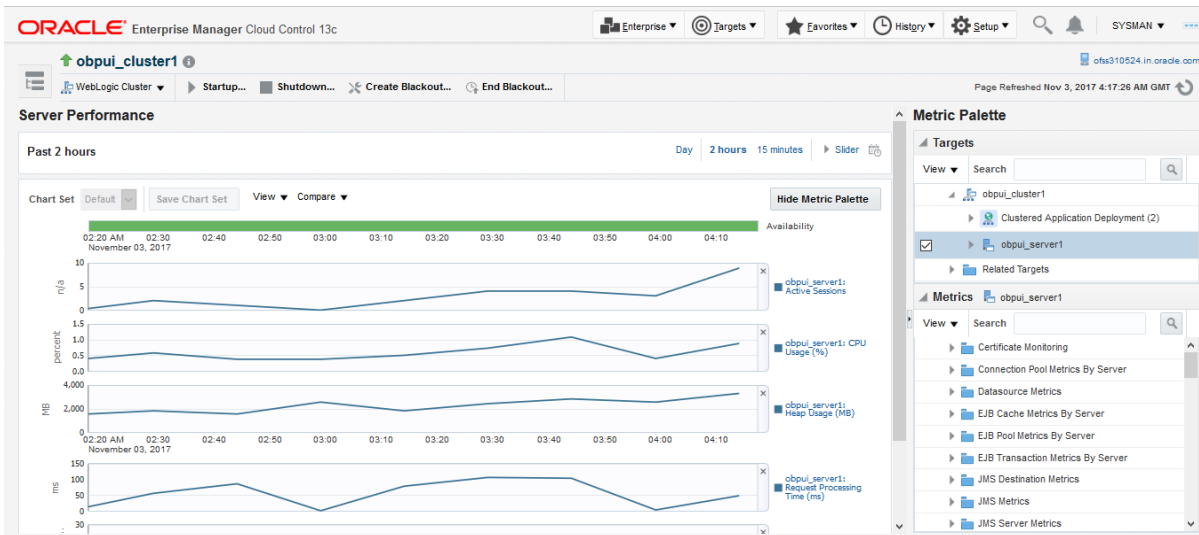
Each component can be further explored for details by clicking on the links provided for them.

10.1.2 UI

For UI, all the managed servers created under Weblogic cluster can be monitored. EM provides the following information for UI Cluster:

- Active Session about all Managed Servers
- CPU Usage
- Heap Usage
- Request Processing Time

Figure 10–2 UI Cluster in EM

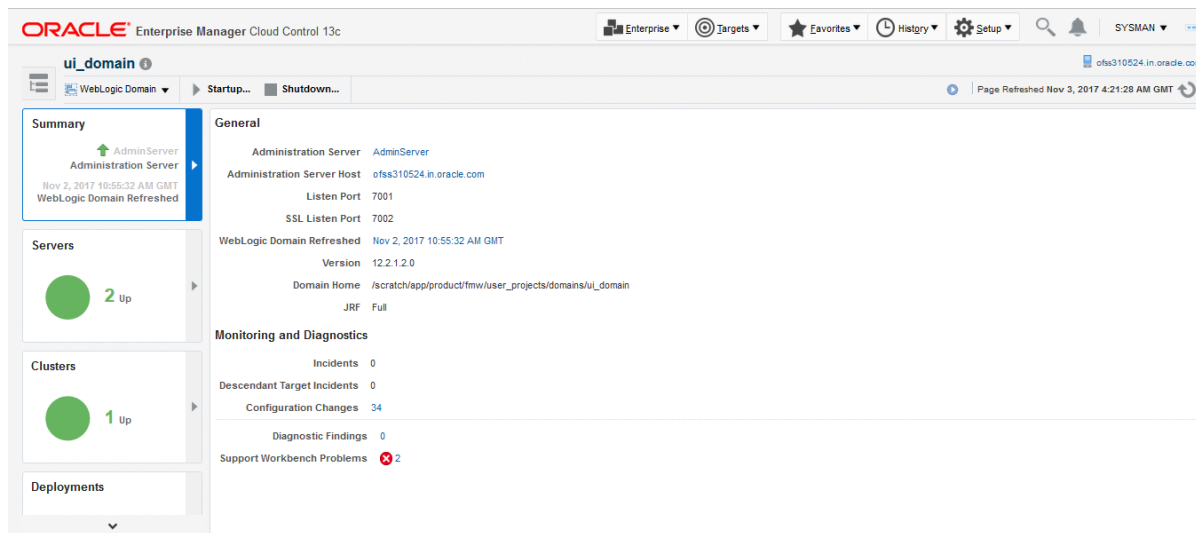


UI is hosted on WebLogic domain, so the EM target of UI machine is WebLogic domain. EM gives the following information for UI:

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

The following figure displays the WebLogic domain for UI.

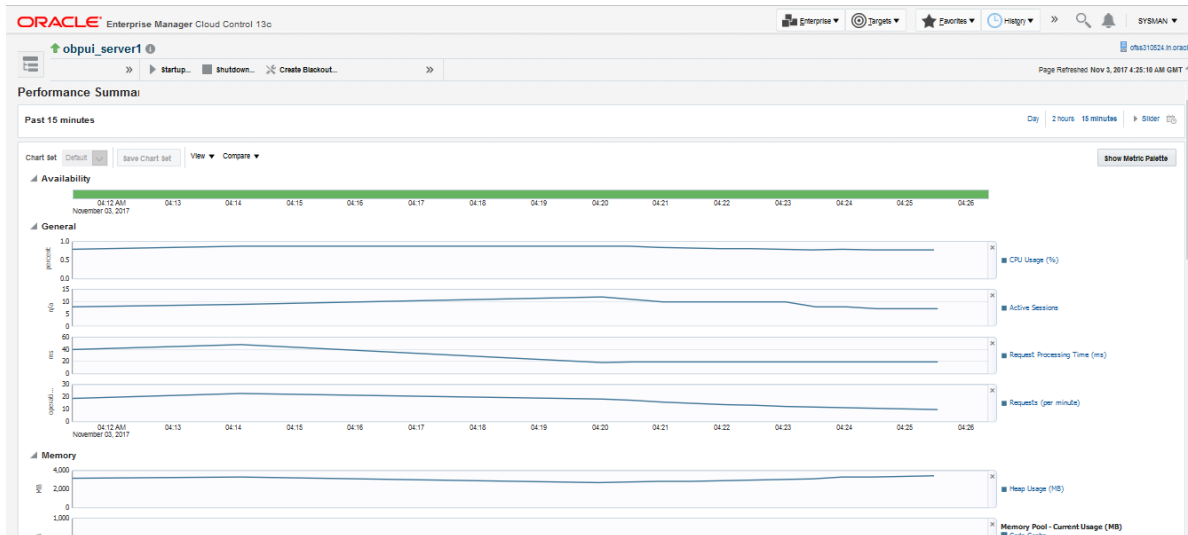
Figure 10–3 WebLogic Domain for UI



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

The following figure displays the metrics chart.

Figure 10–4 Metrics Chart

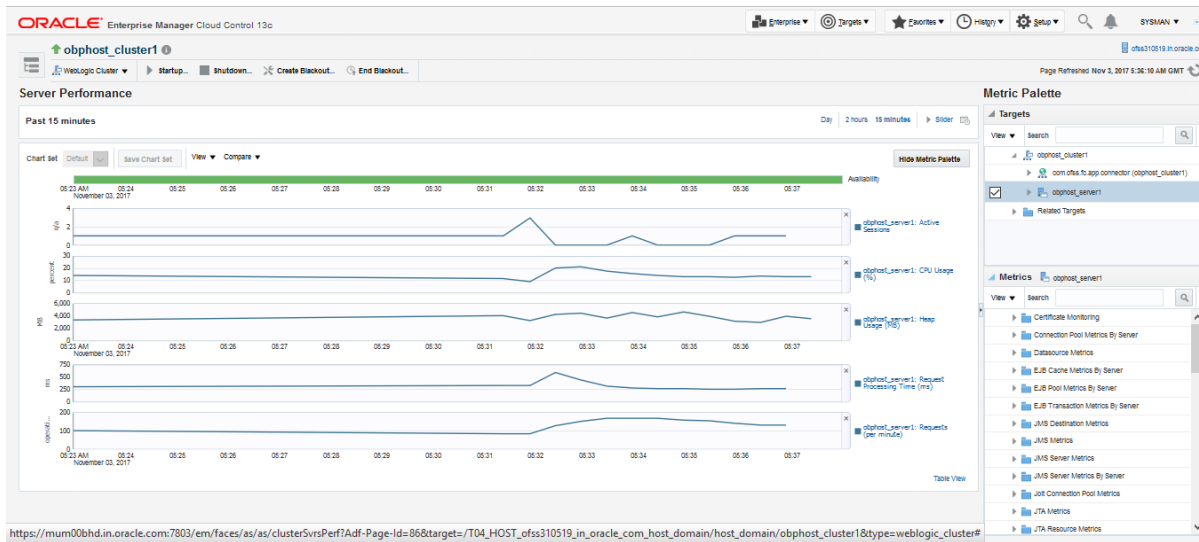


10.1.3 Host

For Host, all the managed servers created under Weblogic cluster can be monitored. EM provides the following information for Host Cluster:

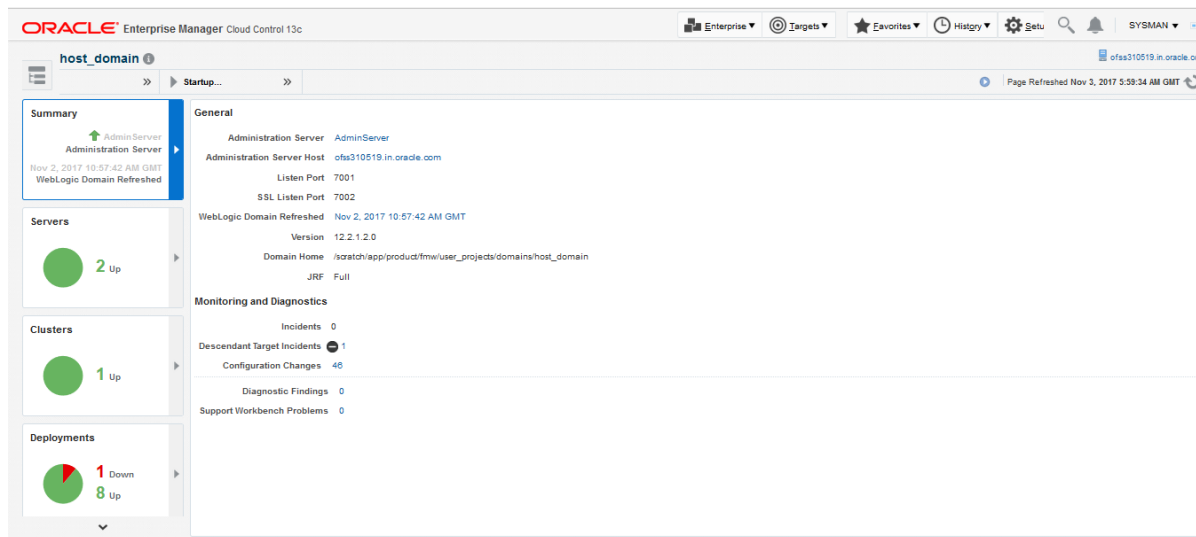
- Active Session about all Managed Servers
- CPU Usage
- Heap Usage
- Request Processing Time

Figure 10–5 Host Cluster in EM



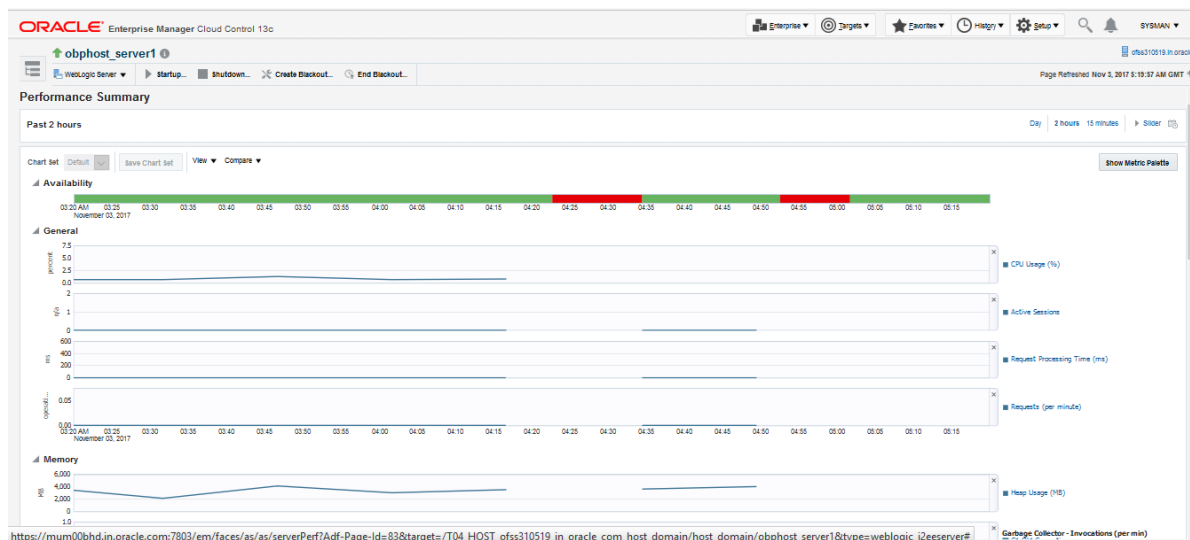
Similar to UI, Host is also deployed on WebLogic domain and has similar metrics like UI. The following figure displays the host target in EM.

Figure 10–6 Host Target in EM



The following figure displays the metric charts.

Figure 10–7 Metrics Chart

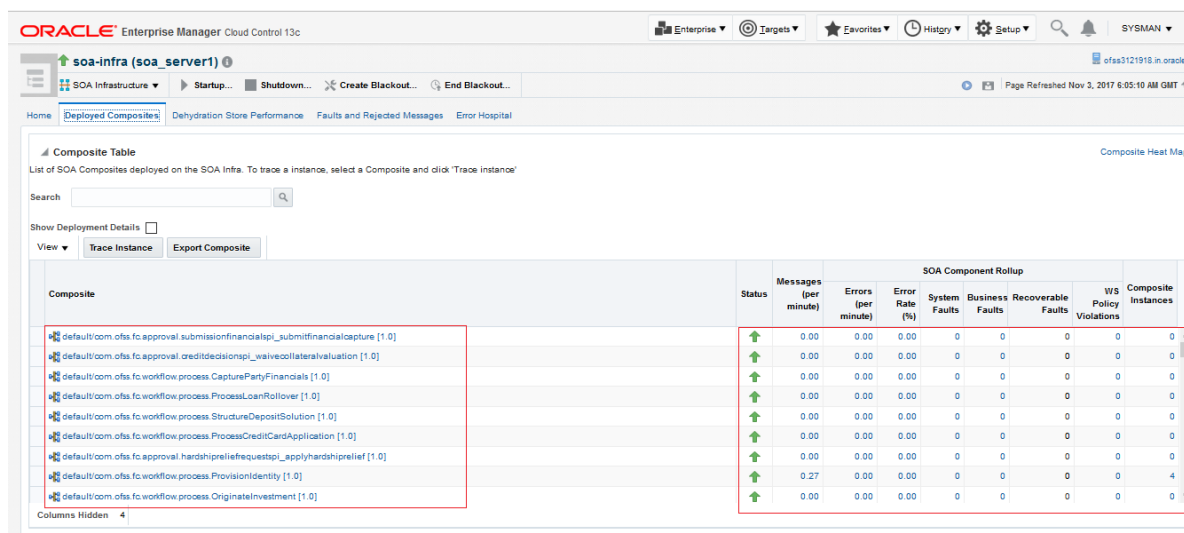


10.1.4 SOA

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. The following figure displays the process list.

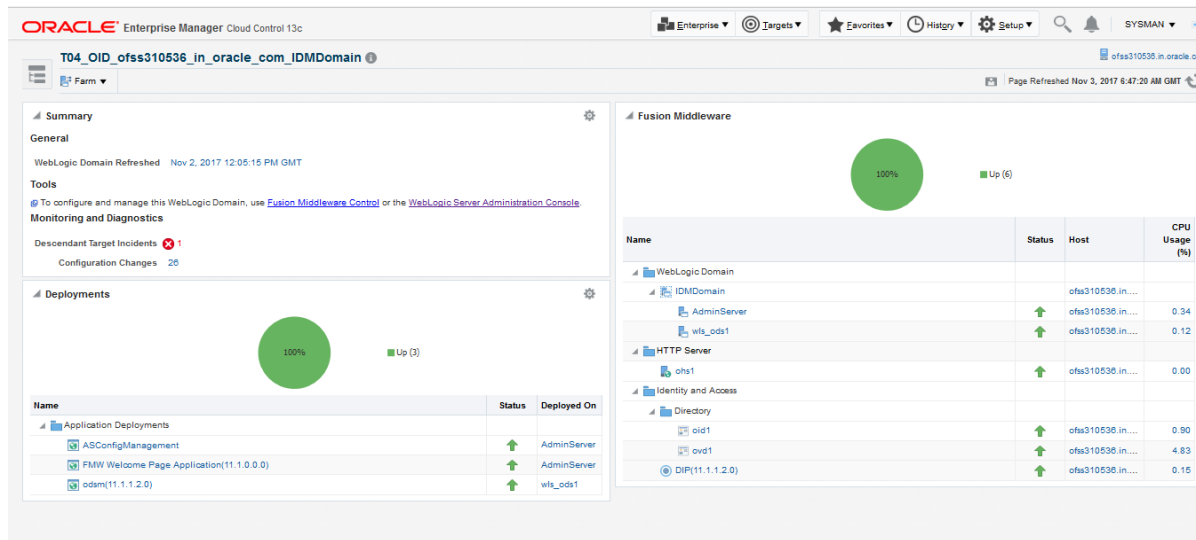
Figure 10–8 Viewing Process List



10.1.5 Security Stack (OID and OAM)

OID and OAM are also deployed as WebLogic domain.

Figure 10–9 OID WebLogic Domain



10.1.6 Document Generation Outbound Components (Documaker, Oracle Analytics Publisher, IPM)

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

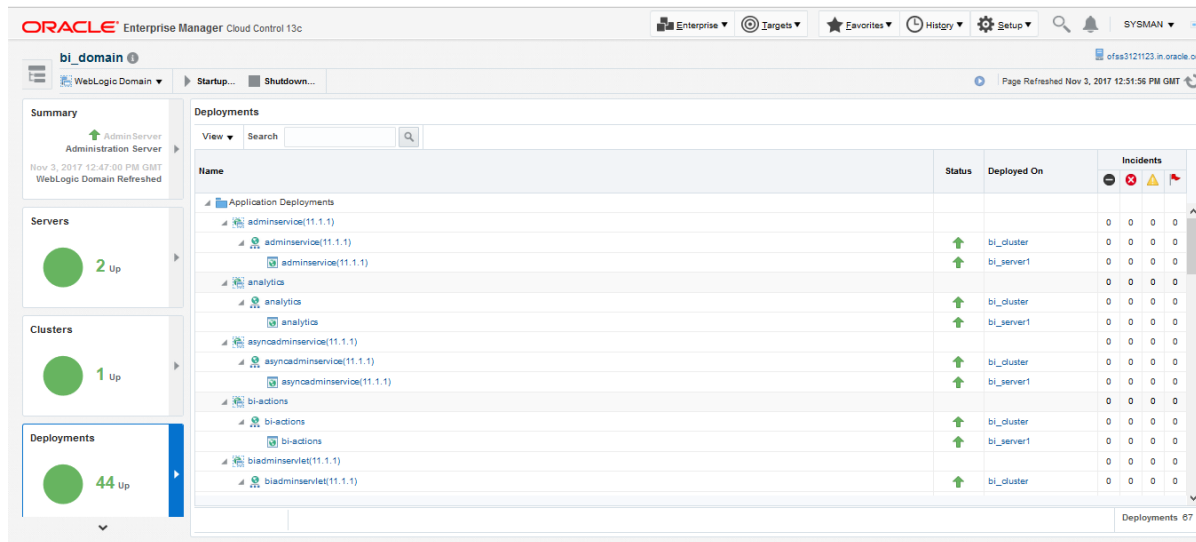
A webservice is invoked for generating the documents from the application with Oracle Analytics Publisher (formerly known as Business Intelligence Publisher (BIP)) as well as Documaker. From EM, we check whether that webservice is up and running or not, which gives the status of these components. The following figure displays the status from EM.

Figure 10–10 Document Generation Status

OBP_T12_BIP_Service	Generic Service	↑	
OBP_T12_Documaker_Service	Generic Service	↑	
OBP_T12_HOST_Service	Generic Service	↑	
OBP_T12_IPM_Service	Generic Service	↑	

Oracle Analytics Publisher is also deployed on WebLogic domain.

Figure 10–11 Oracle Analytics Publisher Deployment



10.1.7 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, check if the port is listening.

The following figure displays the status from EM.

Figure 10–12 EM Monitoring



10.1.8 Outbound OFSAA call

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

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

Figure 10–13 Web Monitoring



10.1.9 Monitoring Views

Monitoring views show the 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.

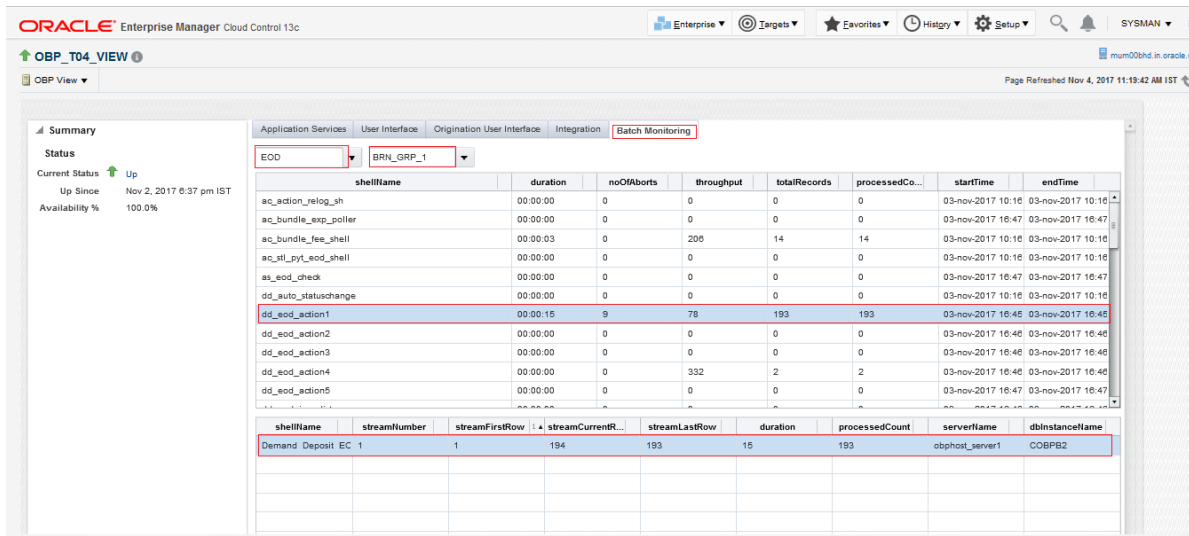
10.1.9.1 Batch Monitoring

Batch Monitoring shows detailed view for host and database server performance charts along with batch performance statistics.

The batch performance statistics are the details of the categories run in the application. 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 10–14 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.

The following figure displays the status from EM.

Figure 10–15 Batch Monitoring Status



Figure 10–16 Batch Configuration

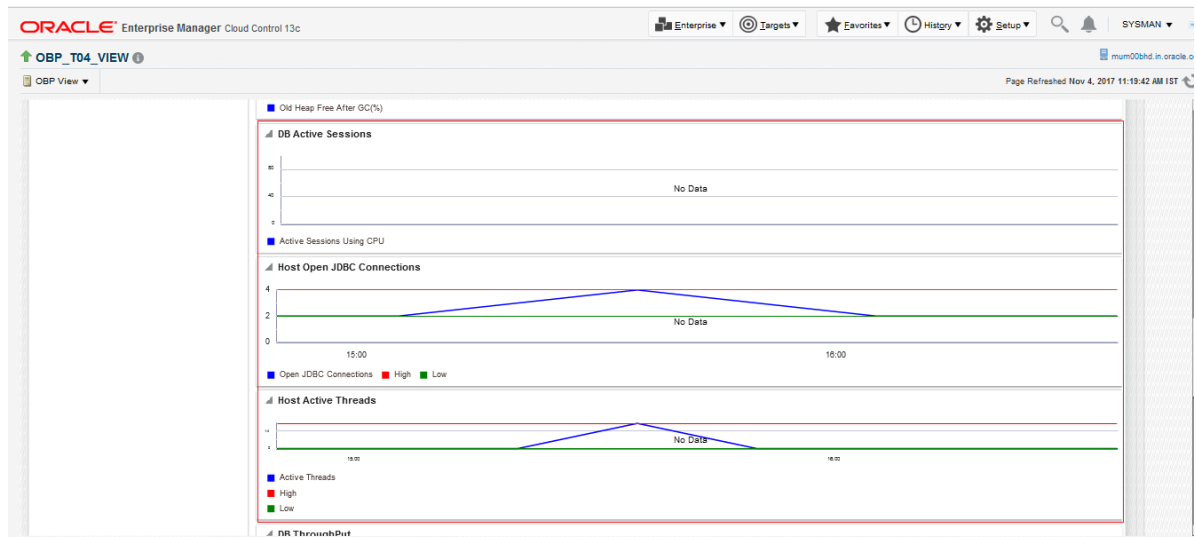
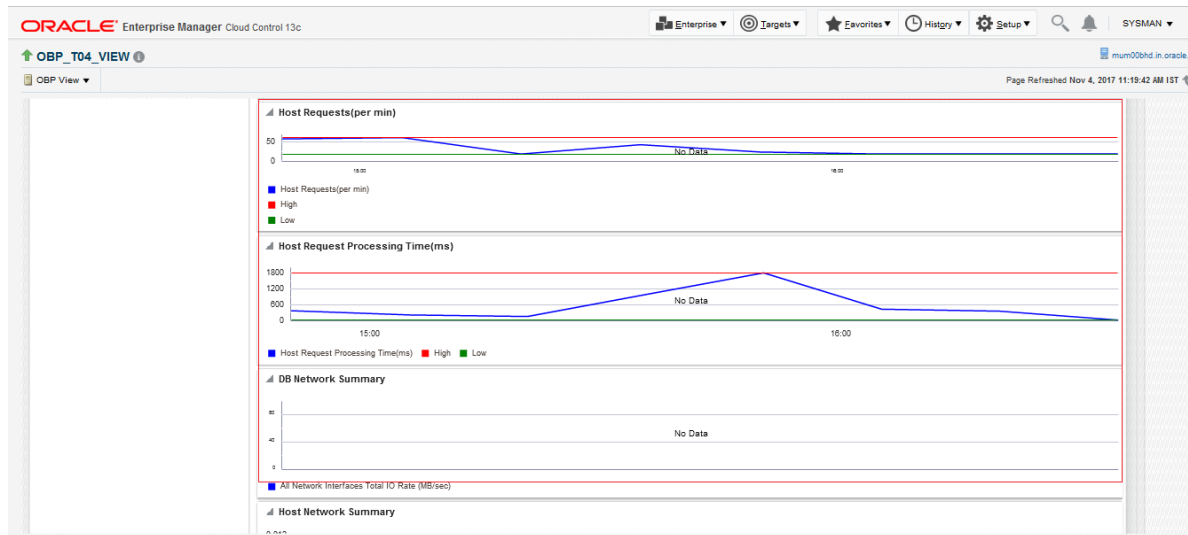


Figure 10–17 WebLogic Service Info



10.1.9.2 Application Monitoring

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

There are four separate tabs, namely Application Services, User Interface, Origination User Interface, and Integration.

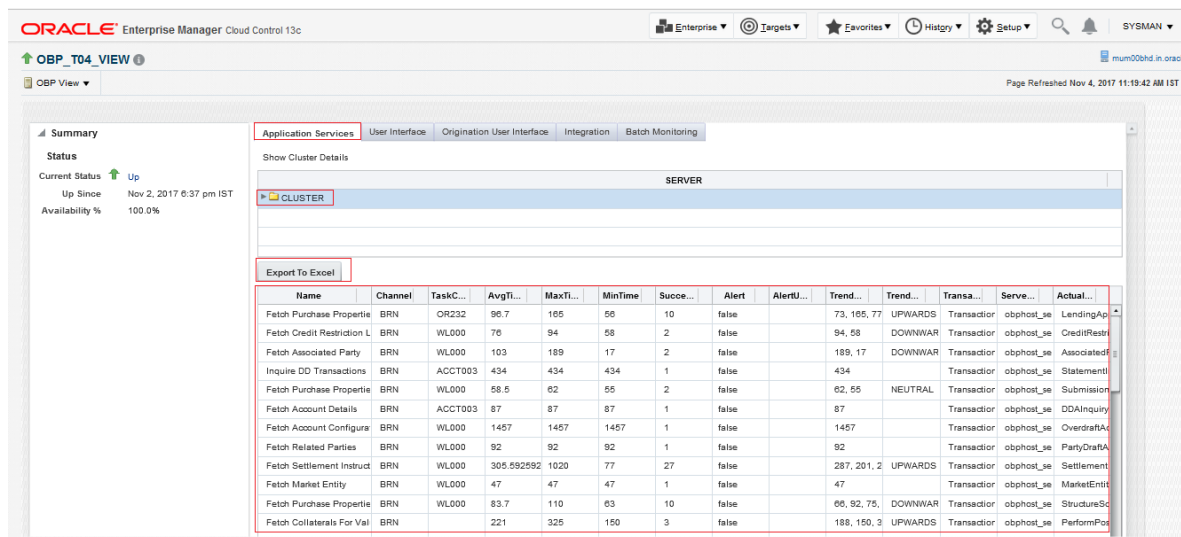
10.1.9.2.1 Application Services

This section provides performance metrics for all application services executed on Host Server. Metrics include timing, alert, trending information, and so on.

For cluster details, click the Show Cluster Details link.

Click CLUSTER to view application metrics for the servers present in the HOST cluster. User can export the application metric data by clicking the Export To Excel button.

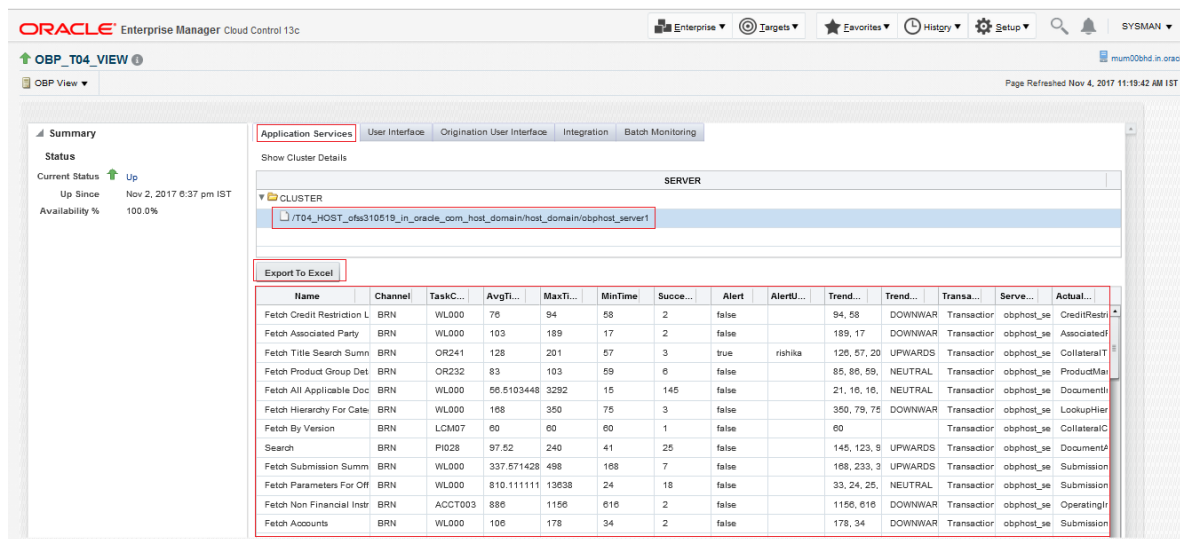
Figure 10–18 Application Metrics of Application Services for all servers in cluster



Expand CLUSTER to get a 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 as shown in the following figure.

Figure 10–19 Application Metrics of Application Services for selected server



The following table explains each column of the table present in the given snapshot:

Table 10–2 Details of the Application Metrics table of Application Services

Sr. No.	Column Name	Description
1	Name	Logical name of the application services

Sr. No.	Column Name	Description
2	Channel	Channel through which the transaction occurred Valid Values: Branch, ATM, and POS.
3	Task Code	Task code of the application page by which the transaction was triggered. Application module of which transaction is a part
4	Average Time	Average execution time of the application service
5	Max Time	Maximum time of execution of the application service
6	Min Time	Minimum time of execution of the application service
7	Success Count	Number of times application service executed successfully
8	Alert	Alert state of the application service
9	Alert User	Teller who performed the last alerted transaction
10	Trend Reference Queue	Execution time of last n transactions (n=5)
11	Trend	Trending of transaction Valid Values: Upwards, Downwards, Neutral
12	Transaction Type	Maximum time of execution of the transaction
13	Server Name	Server name
14	Actual Service Name	Service name of the transaction

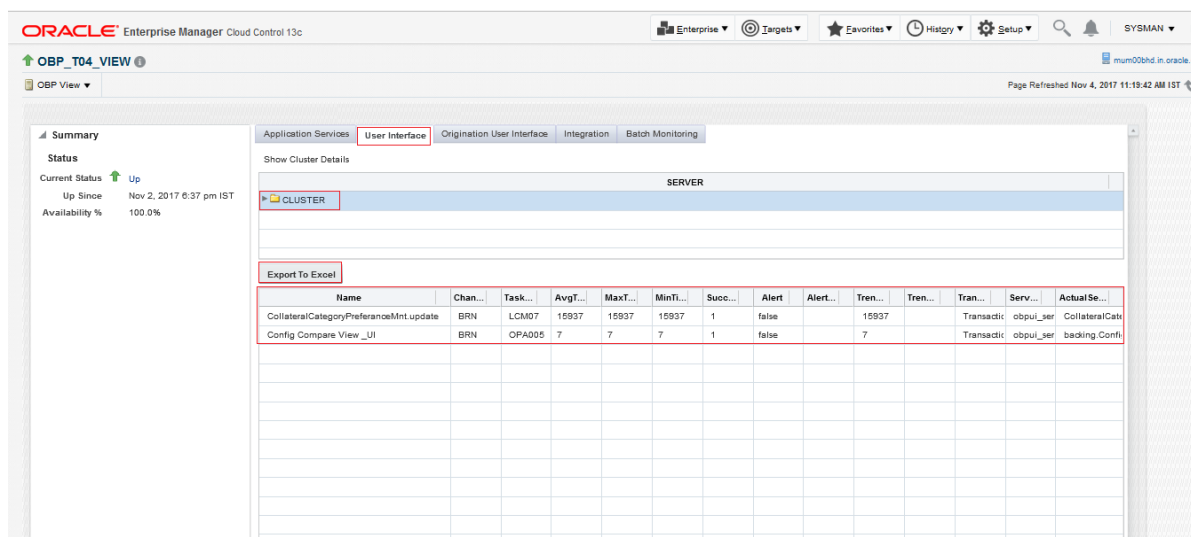
10.1.9.2.2 User Interface

This section provides performance metrics for all major UI components executed on UI Server. Metrics include timing, alert, trending information, and so on.

For cluster details, click the [Show Cluster Details](#) link.

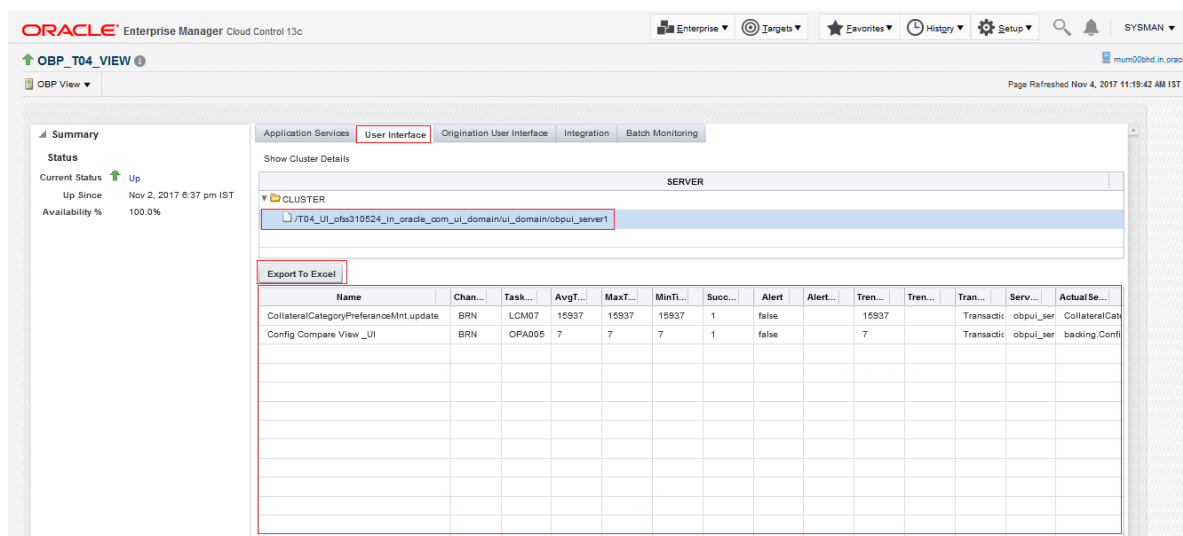
Click **CLUSTER** to view application metrics for the servers present in the UI cluster. User can export the application metric data by clicking the [Export To Excel](#) button.

Figure 10–20 Application Metric for all UI servers in cluster



Expand CLUSTER to get a 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 as shown in the following figure.

Figure 10–21 Application Metrics of UI components for selected server



The following table explains each column of the table present in the given snapshot:

Table 10–3 Details of the Application Metrics table of UI Components

Sr. No.	Column Name	Description
1	Name	Logical name of the UI component
2	Channel	Channel through which the transaction occurred

Sr. No.	Column Name	Description
		Valid Values: Branch, ATM, and POS
3	Task Code	Task code of the application page by which the transaction was triggered. Application module of which transaction is a part
4	Average Time	Average execution time of UI component
5	Max Time	Maximum time of execution of the UI component
6	Min Time	Minimum time of execution of the UI component
7	Success Count	Number of times UI component executed successfully
8	Alert	Alert state of the UI component
9	Alert User	Teller who performed the last alerted transaction
10	Trend Reference Queue	Execution time of last n transactions (n=5)
11	Trend	Trending of transaction Valid Values: Upwards, Downwards, Neutral
12	Transaction Type	Type of transaction
13	Server Name	UI Server name
14	Actual Service Name	Actual name of UI component

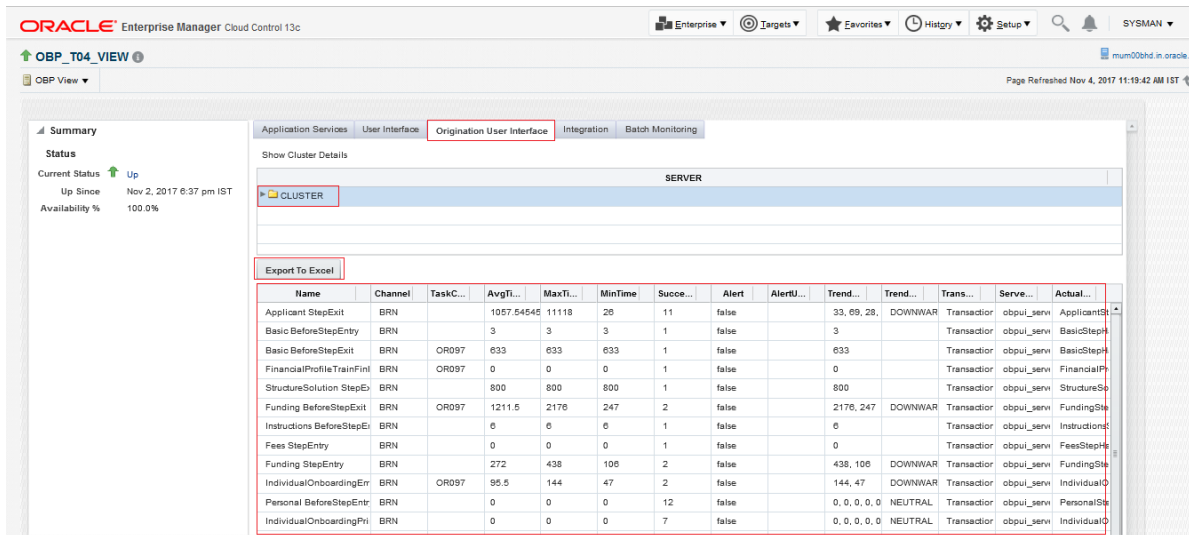
10.1.9.2.3 Origination User Interface

This section provides performance metrics for rendering all multistep train taskflows. The metrics capture the time taken for entering and exiting a particular step of the application form. If there are any host calls made to persist data before leaving a step or to fetch data from host server before entering a step, these metrics encapsulate those timings. Metrics include timing, alert, trending information, and so on.

For cluster details, click the Show Cluster Details link.

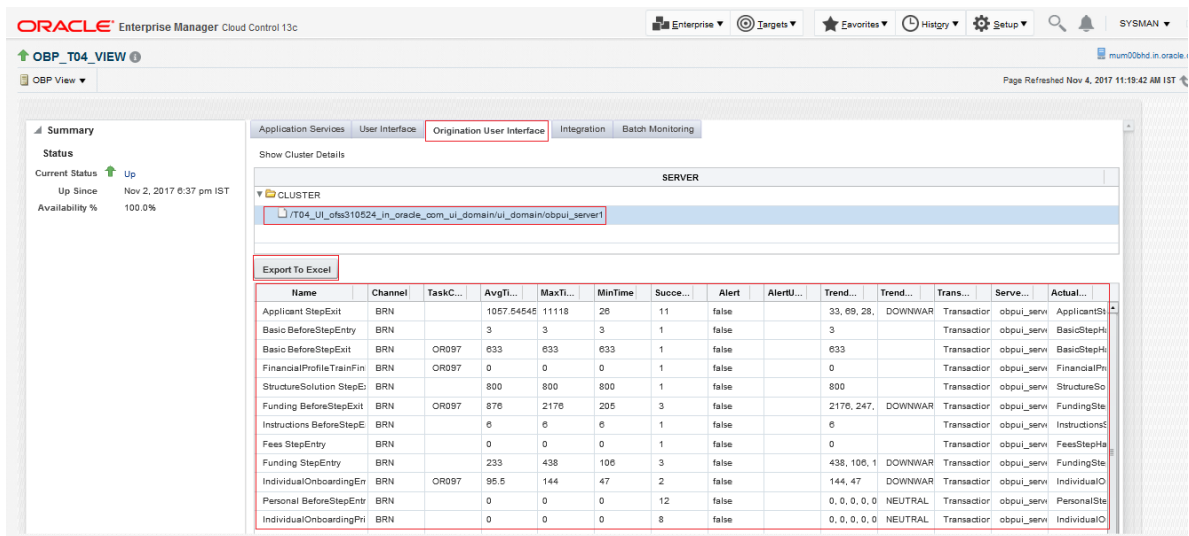
Click CLUSTER to view application metrics for the servers present in the UI cluster. User can export the application metric data by clicking the Export To Excel button.

Figure 10–22 Application Metrics of Origination UI Components for all UI servers in cluster



Expand 'CLUSTER' to get a 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 as shown in the following figure.

Figure 10–23 Application Metrics of Origination UI components for selected server



The following table explains each column of the table present in the given snapshot:

Table 10–4 Details of the Application Metrics table of Origination UI Components

Sr. No.	Column Name	Description
1	Name	Logical name of the multistep train taskflow component

Sr. No.	Column Name	Description
2	Channel	Channel through which the transaction occurred Valid Values: Branch, ATM, and POS.
3	Task Code	Task code of the application page by which the transaction was triggered. Application module of which transaction is a part
4	Average Time	Average execution time
5	Max Time	Maximum time of execution of the multistep train taskflow component
6	Min Time	Minimum time of execution of the multistep train taskflow component
7	Success Count	Number of times multistep train taskflow component executed successfully
8	Alert	Alert state of the multistep train taskflow component
9	Alert User	Teller who performed the last alerted transaction
10	Trend Reference Queue	Execution time of last n transactions (n=5)
11	Trend	Trending of transaction Valid Values: Upwards, Downwards, Neutral
12	Transaction Type	Type of transaction
13	Server Name	UI Server name
14	Actual Service Name	Actual name of multistep train taskflow component

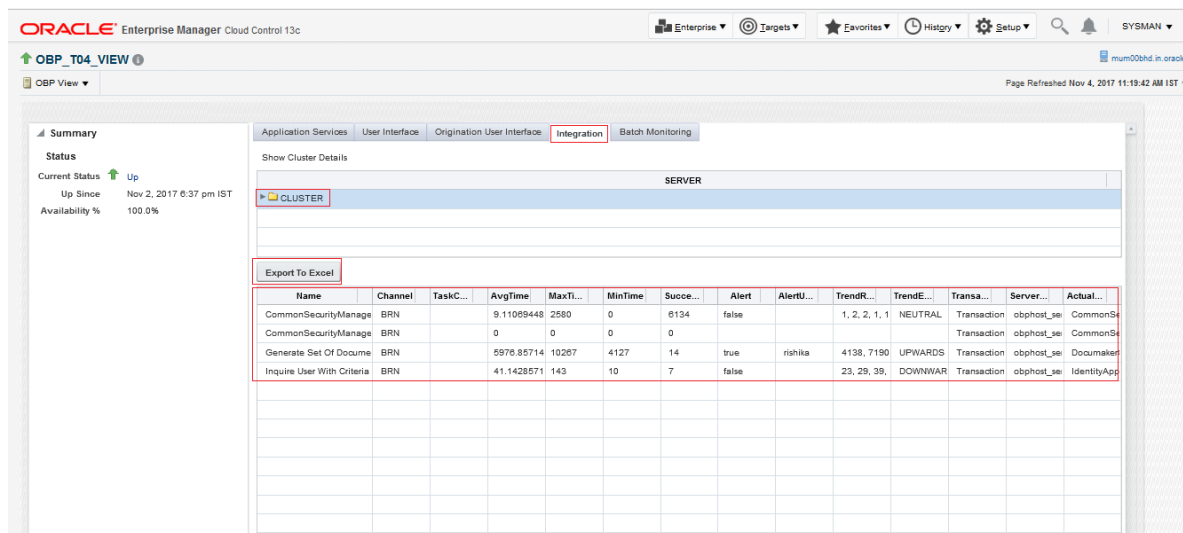
10.1.9.2.4 Integration

This section provides performance metric for all outbound services called from Host Server. Metrics include timing, alert, trending information, and so on.

For cluster details, click the Show Cluster Details link.

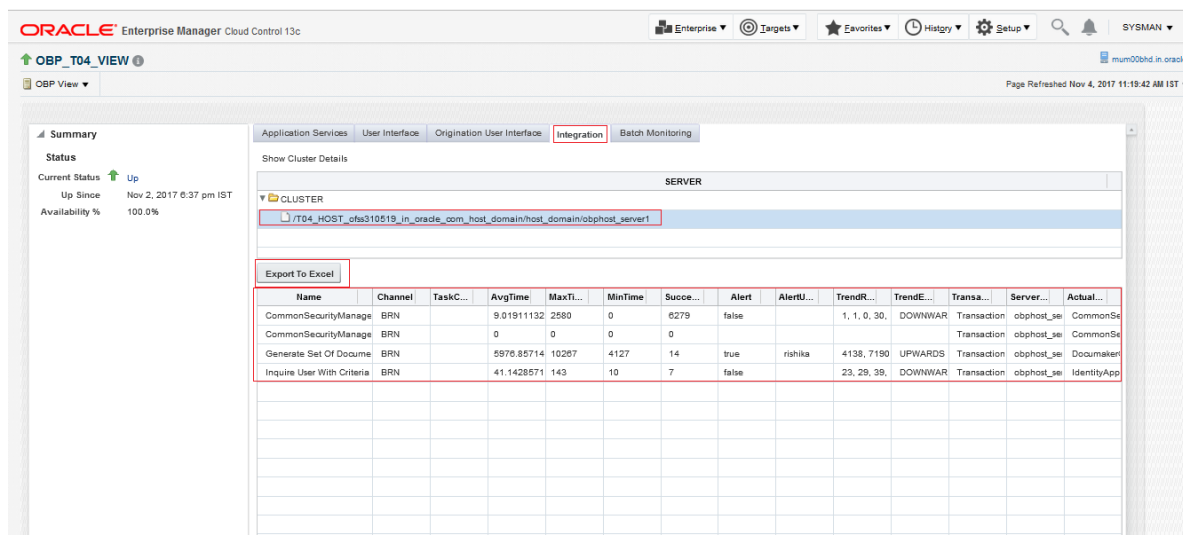
Click CLUSTER to view application metrics for the servers present in the HOST cluster. User can export the application metric data by clicking the Export To Excel button.

Figure 10–24 Application Metrics of all outbound services called from all host servers in cluster



Expand it to get a 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 as shown in the following figure.

Figure 10–25 Application Metrics of all outbound services called from selected server



The following table explains each column of the table present in the given snapshot:

Table 10–5 Details of the Application Metrics table of all Outbound Services

Sr. No.	Column Name	Description
1	Name	Logical name of the Outbound services
2	Channel	Channel through which the transaction occurred Valid Values: Branch, ATM, and POS.

Sr. No.	Column Name	Description
3	Task Code	Task code of the application page by which the transaction was triggered. Application module of which transaction is a part
4	Average Time	Average execution time
5	Max Time	Maximum time of execution of the outbound service
6	Min Time	Minimum time of execution of the outbound service
7	Success Count	Number of times outbound service executed successfully
8	Alert	Alert state of the outbound service
9	Alert User	Teller who performed the last alerted transaction
10	Trend Reference Queue	Execution time of last n transactions (n=5)
11	Trend	Trending of transaction Valid Values: Upwards, Downwards, Neutral
12	Transaction Type	Maximum time of execution of the transaction
13	Server Name	Server name
14	Actual Service Name	Service name of the transaction

11 Configuration Export-Import Operations

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

11.1 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.

11.2 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.

11.3 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.

11.4 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>MODELBANK</con:businessUnit>
<con:channel>BRN</con:channel>
<con:marketEntity>MODEL01</con:marketEntity>
<con:postingDateText>20130228000000</con:postingDateText>
<con:targetUnit>MODELBANK</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/ExportImportA
pplicationService" 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:configVersio
nId>
- <responseservice:status>
<responseservice:errorCode>0</responseservice:errorCode>
<responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016075018875027</respons
eservice:internalReferenceNumber>
<responseservice:isOverriden>>false</responseservice:isOverriden>

```



```

<responseservice:isServiceChargeApplied>>false</responseservice:isServiceChargeApplied>
- <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.

11.5 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>MODELBANK</con:businessUnit>
<con:channel>BRN</con:channel>
<con:marketEntity>MODEL01</con:marketEntity>
<con:postingDateText>20130228000000</con:postingDateText>
<con:targetUnit>MODELBANK</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 Response

```

- <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/ExportImportAp
plicationService"
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:internalReferen
ceNumber>
<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>

```

11.6 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.

11.7 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.

11.7.1 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.

11.7.2 File Data Store

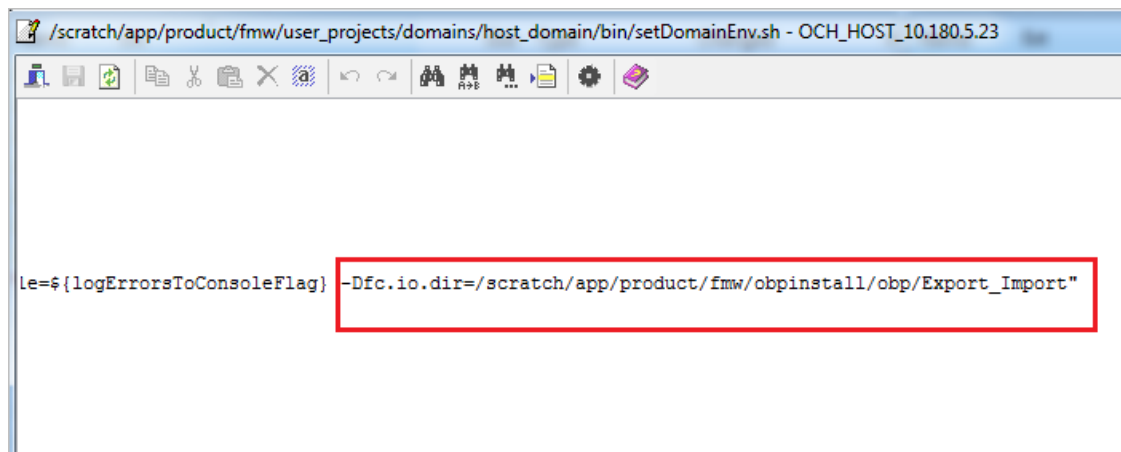
This configuration stores the exported data to file. For using this configuration, following changes have to be made:

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 11–1 File Data Store



11.7.2.1 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**.

11.7.2.2 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.

11.8 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 11.7 Data Store Configuration](#).

11.8.1 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/ActivityEventActionApplicationService" 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.processNotificationForCompliance</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/">r00ABXdWABx
3ZWJsb2dpYy5hcHAub2JwLXd1YnN1cnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrY
XJlYS5TdHJpbmdXb3JrQ29udGV4dAAJMi4yLjAuMC4wAAA=</work:WorkContext>
</S:Header>
- <S:Body>
-
<ns14:fetchActivityEventActionAndExportActivityEventActionResponse
xmlns:ns14="http://action.service.ep.app.fc.ofss.com/ActivityEvent
ActionApplicationService"
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.ap
p.fc.ofss.com">
- <ns14:return>

<responseservice:configVersionId>187</responseservice:configVersio
nId>

```

```

- <responseservice:status>
<responseservice:errorCode>0</responseservice:errorCode>
<responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016305031622005</respons
eservice:internalReferenceNumber>
<responseservice:isOverriden>>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>>false</responseservice:isS
erviceChargeApplied>
- <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</dtocommondomainframewor
k:auditSequence>

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

<dtocommondomainframework:generatedPackageId>>false</dtocommondomai
nframework:generatedPackageId>

<dtocommondomainframework:lastUpdatedBy>OFSSUser</dtocommondomainf
ramework:lastUpdatedBy>
- <dtocommondomainframework:lastUpdatedDate>
<datatype:dateString>20140721162124</datatype:dateString>
</dtocommondomainframework:lastUpdatedDate>

<dtocommondomainframework:version>2</dtocommondomainframework:vers
ion>
<actiondtoepapp:alertName>Covenant Status
Complied</actiondtoepapp:alertName>
- <actiondtoepapp:alertTemplate>

<dtocommondomainframework:auditSequence>1</dtocommondomainframewor
k:auditSequence>

<dtocommondomainframework:generatedPackageId>>false</dtocommondomai
nframework:generatedPackageId>

```

```
<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:generatedPackageId>

<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:generatedPackageId>

<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.DefaultRuleHandler</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.BatchCovenantService.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:generatedPackageId>

<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:generatedPackageId>

<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:generatedPackageId>

<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.DefaultRuleHandler</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.BatchCovenantService.processNotificationForCompliance</subscriberactiondtoepapp:activityId>

<subscriberactiondtoepapp:destinationType>EMAIL</subscriberactiondtoepapp:destinationType>
<subscriberactiondtoepapp:eventId>LM_STATUS_COV</subscriberactiondtoepapp:eventId>
<subscriberactiondtoepapp:messageTemplateId>LCM_Covenant status is Complied</subscriberactiondtoepapp:messageTemplateId>

<subscriberactiondtoepapp:subscriberType>PARTY</subscriberactiondtoepapp:subscriberType>

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

<subscriberactiondtoepapp:recipientType>INTERNAL</subscriberactiondtoepapp:recipientType>
</actiondtoepapp:recipientMessageTemplates>
</actiondtoepapp:activityEventActionDTO>
</ns14:return>
```

```

</ns14:fetchActivityEventActionAndExportActivityEventActionResponse>
</S:Body>
</S:Envelope>

```

11.8.2 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/">r00ABXdWABx
3ZWJsb2dpYy5hcHAub2JwLXd1YnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrY
XJlYS5TdHJpbmdXb3JrQ29udGV4dAAJMi4yLjAuMC4wAAA=</work:WorkContext>
</S:Header>
- <S:Body>
- <ns13:fetchAllAndExportExportImportResponse
xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportA
pplicationService" 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:configVersio
nId>
- <responseservice:status>
<responseservice:errorCode>0</responseservice:errorCode>
<responseservice:extendedReply />

<responseservice:internalReferenceNumber>2016305031622003</respons
eservice:internalReferenceNumber>
<responseservice:isOverriden>>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>>false</responseservice:isS
erviceChargeApplied>
- <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 11.7 Data Store Configuration](#).

Figure 11–2 Exported Data

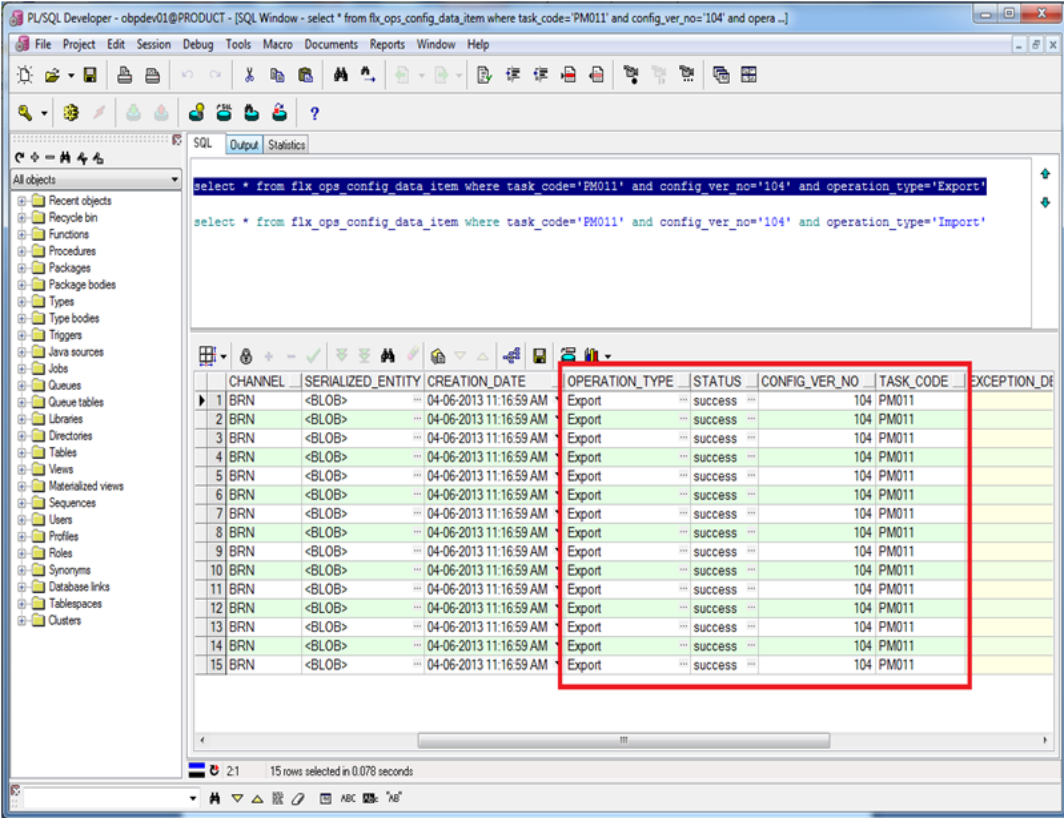
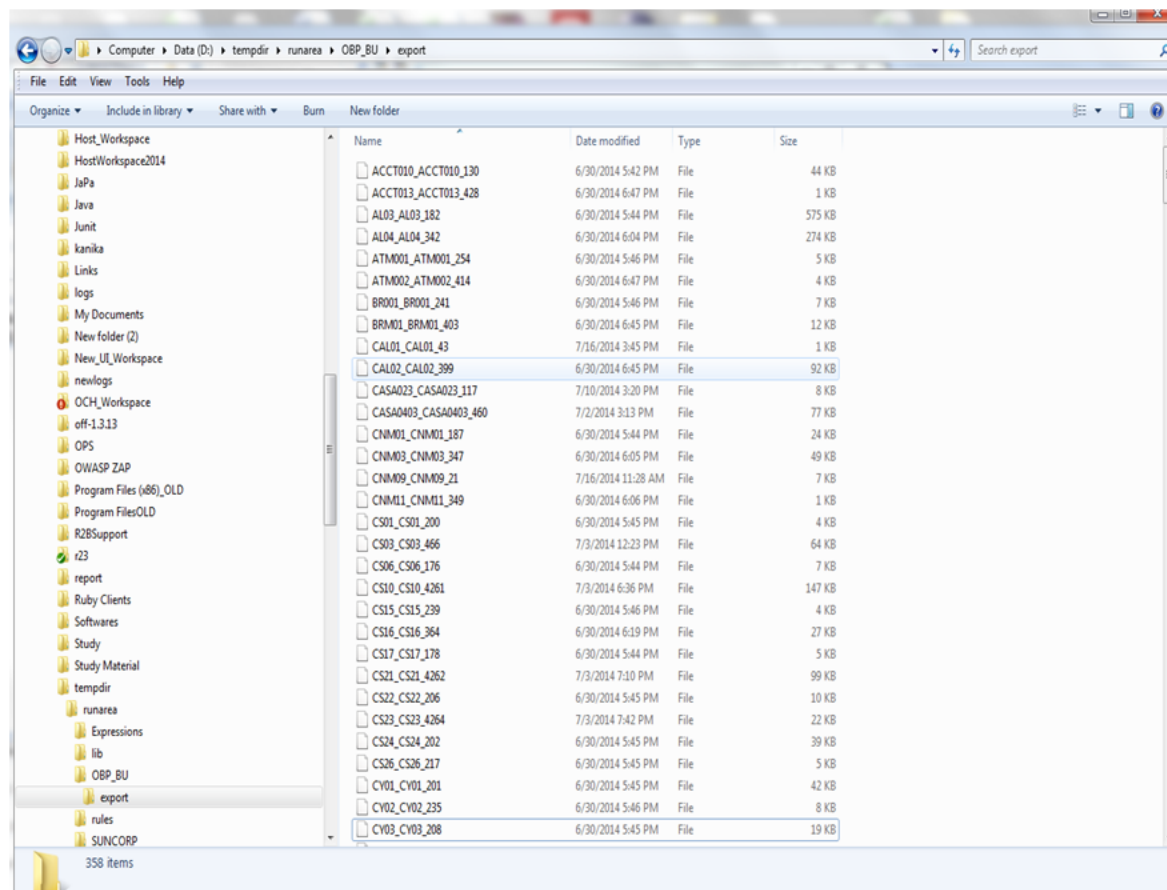


Figure 11–3 Exported Files



11.9 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.

11.9.1 Importing Single Record

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

11.9.1.1 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/ExportImportAp
  plicationService" 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/">r00ABXdWABx
  3ZWJsb2dpYy5hcHAub2JwLXd1YnN1cnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrY
  XJlYS5TdHJpbmdXb3JrQ29udGV4dAAAJMi4yLjAuMC4wAAA=</work:WorkContext>
</S:Header>
- <S:Body>
- <ns13:importAllExportImportResponse
  xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportA
  pplicationService" 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:internalReferenceNumber>
<responseservice:isOverriden>>false</responseservice:isOverriden>

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

```

11.9.2 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/">r00ABXdWABx
3ZWJsb2dpYy5hcHAub2JwLXdlYnNlcnZpY2VzAAAA1gAAACN3ZWJsb2dpYy53b3JrY
XJlYS5TdHJpbmdXb3JrQ29udGV4dAAAJMi4yLjAuMC4wAAA=</work:WorkContext>
</S:Header>
- <S:Body>
- <ns13:importAllExportImportResponse
xmlns:ns13="http://eximp.service.ops.app.fc.ofss.com/ExportImportA
pplicationService" 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</respons
eservice:internalReferenceNumber>
<responseservice:isOverriden>>false</responseservice:isOverriden>

<responseservice:isServiceChargeApplied>>false</responseservice:iss
erviceChargeApplied>
<responseservice:replyCode>0</responseservice:replyCode>
<responseservice:spReturnValue>0</responseservice:spReturnValue>

```

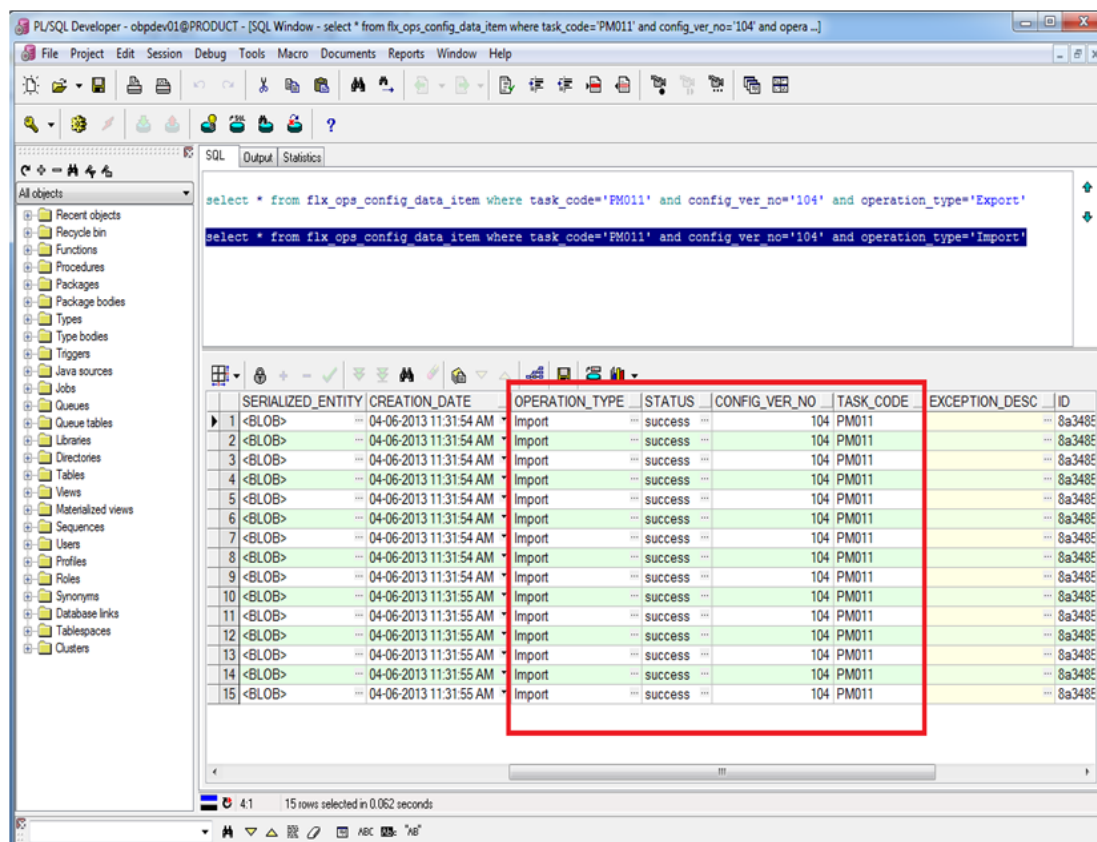
```

</ns13:return>
</ns13:importAllExportImportResponse>
</S:Body>
</S:Envelope>

```

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

Figure 11–4 Importing Data Using SOAP UI - Storing Response



11.10 Configuration Comparison

This section describes the details of configuration comparison.

11.10.1 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).

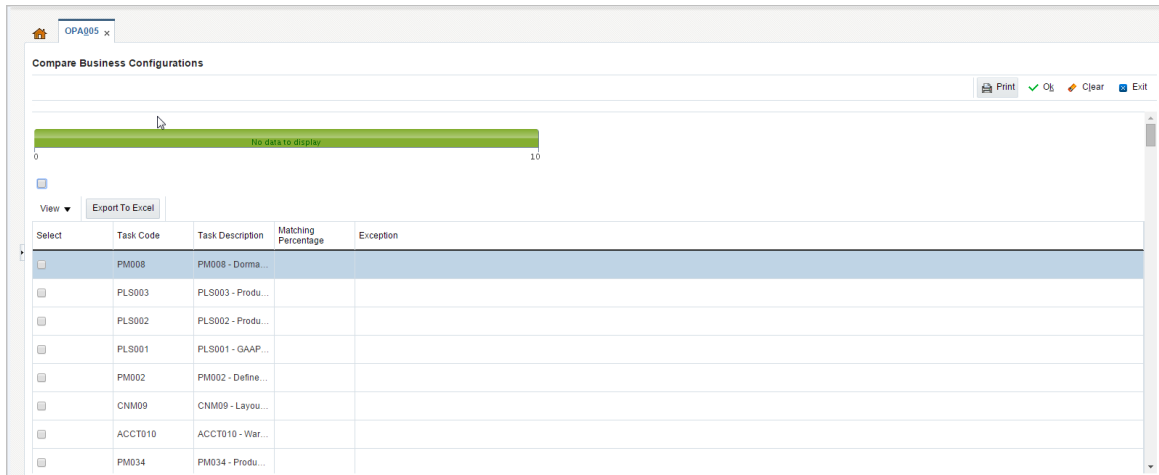
- Present in both, but data is different.

11.10.2 Usage

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

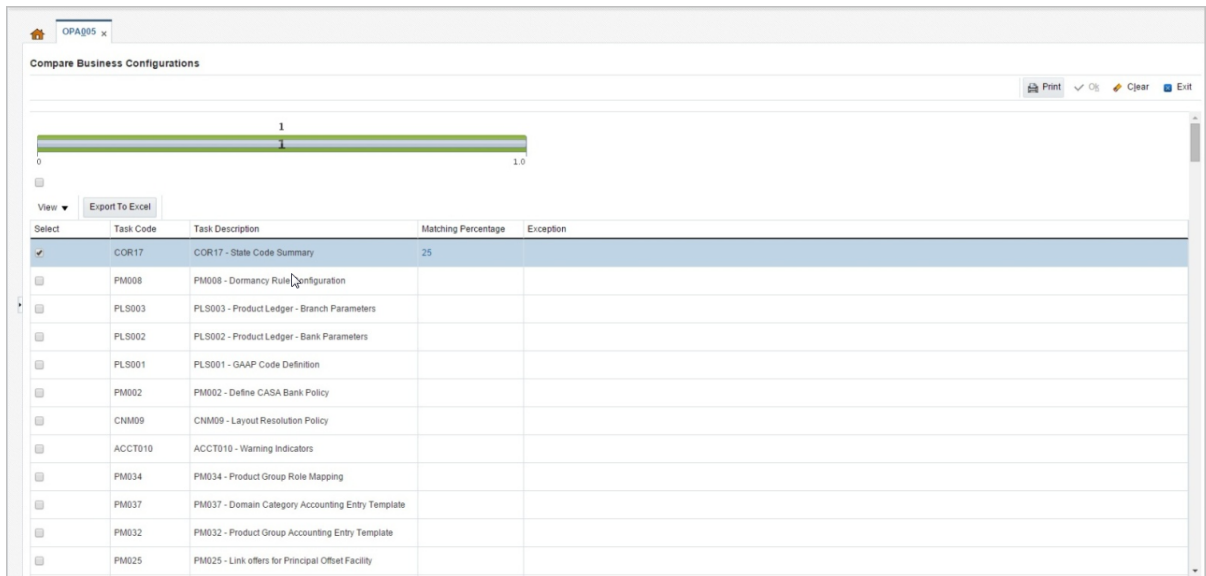
1. Open the OPA005 page, which loads all the entities configured in the table `flx_ops_task_defn`.

Figure 11–5 Entity Comparison



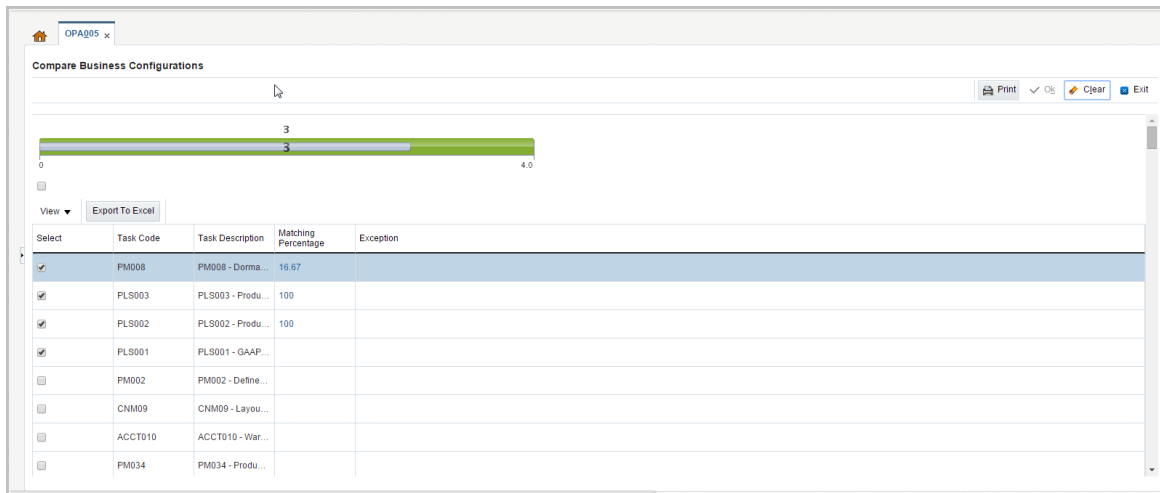
2. Select the option under **Select** column to do the comparison of configuration/configurations which shows the matching % of data in the two environment.

Figure 11–6 Entity Comparison Results



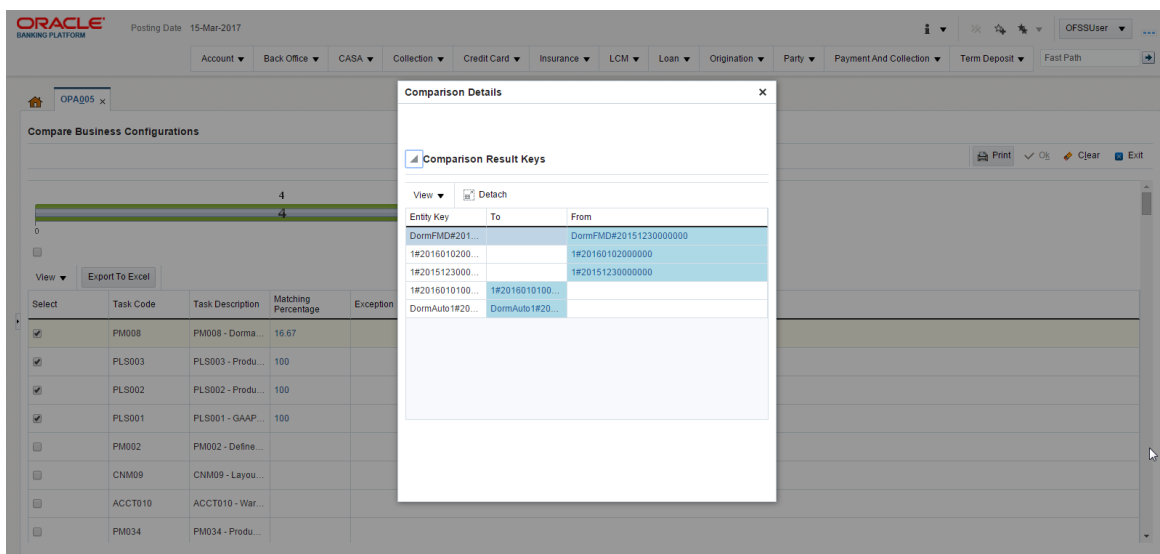
Progress bar denoting that the compare operation has finished.

Figure 11–7 Progress Bar



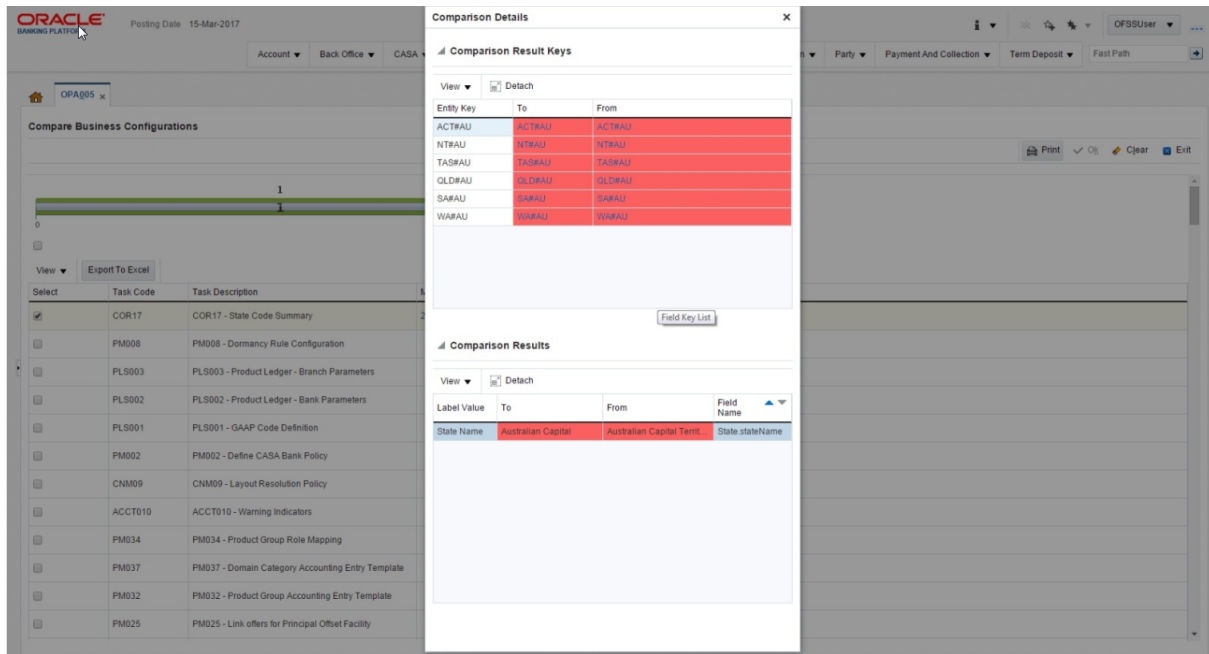
3. Select the % match to launch a pop up which shows the comparison result with different keys associated to it.

Figure 11–8 Comparison Details



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

Figure 11–9 Attributes Difference



11.11 Application Configuration

This section describes the application configuration details.

11.11.1 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.PropertiesFileC
onfigProvider" parent="" propertyFileName="jdbc.properties"
      syncTimeInterval="600000" />
    <Preference name="ConfigurationVariable"
      PreferencesProvider="com.ofss.fc.infra.config.impl.DBBasedProperty
Provider" 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.JavaConstantsCo
nfigProvider" parent="jdbcpreference"
      propertyFileName="com.ofss.fc.common.ChannelConstantsConfiguration"
      syncTimeInterval="600000" />
  
```

```
<Preference name="JSONServiceMap"
  overriddenBy="JSONServiceMapOverride"
  PreferencesProvider="com.ofss.fc.infra.config.impl.JavaConstantsCo
nfigProvider" 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 provider 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.

12 Batch Shells in OBP

This chapter describes the batch shells used in OBP and their execution sequence.

12.1 Batch Shells Description

The following table lists the batch shells along with their detailed description.

Table 12–1 Batch Shell Descriptions

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
1	0	Reports Health Check	reports_chk_shell	Reports Health Check	FW	This is dummy shell. It is used to check whether report framework is ready to process reports and to check if there is any issue on framework or not.
2	1	End of Day	ac_action_relog_sh	Relog actions on holiday	AC	The actions are logged into the base tables (Module Specific). When there change in calendars, this shell rearranges the pickup dates for all the affected future actions, as per the new calendar dates.
3	1	End of Day	ac_bundle_fee_shell	Bundle Fee Shell	BN	This shell charges the bundle period fees defined while originating CASA.
4	1	End of Day	bundle_report_shell	Bundle Reports	BN	This shell generates the various bundle offer related reports.
5	1	End of Day	dd_auto_statuschange	DDA Auto status change shell	DD	The shell marks the accounts whose last activity date is breached and can be either auto closed or identified for manual closure. Similarly based on last activity date, the account can be auto unclaimed or identified for unclaim that could be unclaimed after taking an operational decision.
6	1	End of Day	idl_actn_eod_shell	Indirect Lending Action EOD Shell	IL	
7	1	End of Day	ins_quote_exp_eod	Expire Quote EOD Shell	LM	This shell processes all the Insurance quote's quote status to Expired on quote expiry date.
8	1	End of Day	je_susp_bal_shell	Journal Entry Suspense	JE	This shell checks if each of the single entry batches are balanced, and if not it passes a suspense entry into the branch suspense GL under the same branch.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
				Balancing EOD Shell		It checks for the balancing Branch wise (Contingent/Real) or Period Code wise.
9	1	End of Day	pc_blr_remit_shell	Biller Remittance Shell	PC	This shell does the consolidation biller amount settlement in EOD, for example, selected settlement mode of biller like CASA or GL during biller setup.
10	1	End of Day	pc_value_date_eod_sh	Value Date EOD Shell	PC	This shell moves the funds from unclear to clear balance, where float of instrument is getting over at EOD. This is the 1st shell that runs post cut off. Value Date Clearing shell is run for 0 Day float items. Hence, it becomes mandatory to run it with the current process date even though cut off has been run. This is taken care of by the Value Date Clearing Shell. As the float realization is derived based on branch dates, the date is always set to the working day of the branch.
11	1	End of Day	pi_srv_ord_sta_eod	Service Order Deactivation EOD Shell	PI	This shell handles de-activation of Service Orders based on Service Order end date. (US localization specific shell)
12	1	End of Day	py_bank_val_date_eod	Bank Value Date EOD Shell	PY	This shell will intimate the respective account modules (CASA\Loans\TD) on a given Bank Value Date for any valid transaction posted successfully in Payments module. The respective account module will update the Bank Float Balance bucket for the respective accounts on this date which is taken further for interest calculations. The funds are still not available for the customer.
13	1	End of Day	py_regcc_sch_gen	Reg CC Schedule Generation EOD Shell	PY	This shell will pick the valid transactions posted in Payments module and generate the schedule of fund availability for the respective account (Reg CC capable CASA accounts only) as per the rules defined in the system for Reg CC. (US localization specific shell)
14	1	End of Day	py_regcc_sch_rel_eod	Reg CC Schedule Release EOD Shell	PY	This shell will process only the accounts which follow the RegCC contract in the US localization. This shell will make funds available to the customer as per the generated schedule from "Reg CC Schedule Generation EOD Shell". (US localization specific shell)
15	1	End of Day	dd_inst_rearrg_sh	DD Instruction Rearrangement For Calender	DD	This shell rearranges instruction pickup dates after calendar change for SI and sweep outs.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
				Change		
16	1	End of Day	je_batch_hist_shell	Journal Entry Batch History EOD Shell	JE	<p>This shell performs the following activities:</p> <ol style="list-style-type: none"> 1. Moves the Journal Entries (Single and Multi) details into history. 2. Flushes the Batch codes table (Single and Multi). 3. Drops the Sequences related to the above batch codes which are used to generate the Journal Entry Reference numbers.
17	1	End of Day	pl_onl_ledg_bal_upd	Online Ledger Entries Update	AS	This shell updates the GL balance of all the online transactions in the day. If the GL balance is updated using MDB, then there is contention for huge number of entries and hence batch shell to update the GL balance ap_as_gl_bal_update_on
18	1	End of Day	pc_report_shell	PC Reports	PC	This shell generates Payments and Clearing Reports.
19	1	End of Day	py_cust_val_date_eod	Customer Value Date EOD Shell	PY	This shell will intimate the respective account modules (CASA\Loans\TD) on a given Customer Value Date for any valid transaction posted successfully in Payments module. The respective account module will make the funds available for the customer on this date. This will process the transactions in EOD.
20	1	End of Day	eod_preval	EOD pre validation Shell	DD	Reprice Failed Accounts Verify. The shell is to cross check if there are still pending accounts, for which repricing was failed. EOD will not start if there are such pending accounts.
21	1	End of Day	pr_price_changes_eod	Price change action log shell for account opened today	PR	This shell identifies the accounts (which are opened today) eligible for rate and fee changes and logs action for further processing in pr_reprice_acn_eod shell.
22	1	End of Day	pr_reprice_acn_eod	Reprice Action shell for account opened today	PR	This shell processes accounts (which are opened today) for interest or fee changes happened in the system.
23	1	End of Day	dd_si_eod	DDA Standing Instruction	DD	This shell executes all Standing Instructions that need to be executed for the day, based on holiday processing parameter.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
				EOD Shell		
24	1	End of Day	dd_swp_eod_default	DDA Sweepout Instruction EOD Shell Default Level	DD	Account may be set up to sweep the balance above a certain amount to other CASA or TD account. This is the primary shell where all accounts with priority 1 are executed. This is executed in EOD based on customers preference.
25	1	End of Day	dd_swp_eod_non_def	DDA Sweepout Instruction EOD Shell Non Default Level	DD	This shell behaves same as DDA Sweepout Instruction EOD Shell Default Level. But priority 2 and above sweep out instructions are executed. This is required to divide the priority 1 and priority 2 above to avoid conflicts among accounts.
26	1	End of Day	dd_swp_hold_remove	SweepIn Hold Removal shell	DD	This shell removes the sweep hold marked during the day.
27	1	End of Day	ln_contri_eod_shell	Loan Contribution EOD Shell	LN	This shell processes customer's Contribution Instruction set at time of Origination.
28	1	End of Day	ac_stl_pyt_eod_shell	Account Settlement Payout EOD Shell	AC	This shell execute settlement pay out instructions that are maintained on CASA & Loan Accounts
29	1	End of Day	td_remove_swpin_lien	TD Remove Sweep In Shell	TD	During sweep in process, when CASA is short of funds, funds will be taken from TD as provider account by marking lien to TD account. If CASA is not funded before EOD processing, system will remove the lien from TD account and debit TD account.
30	1	End of Day	ln_split_eod_shell	Split Loan Account Opening EOD Shell	LN	Split Loan Account Opening and Disbursement request execution
31	1	End of Day	td_eod_action	Term Deposit EOD Shell	TD	This shell logs the following actions: <ul style="list-style-type: none"> ■ Lien Expiry - mark lien as inactive during EOD process.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<ul style="list-style-type: none"> ■ Back dated Interest computation - For Contract Modification OR Backdated rate Change. ■ Balance Change - Balance change (at time of contact modification) due to Contract Modification, Part Redemption. ■ Interest Capitalization - Interest capitalization to TD. ■ TD statement generation - Generates TD statements. ■ Account Closure - Close the TD account due for that day.
32	1	End of Day	In_actn_eod_shell	Loans Action Due EOD Shell	LN	<p>This shell performs the following actions:</p> <ul style="list-style-type: none"> ■ Interest Charging ■ Interest Compounding ■ Interest Cap ■ Fee Cap ■ Customer Initiated Rollover ■ Auto Disbursement ■ Repayment Review (Apply and Generate) also called as periodic reschedule
33	1	End of Day	td_eod_report_shell	TD EOD Reports	TD	This shell generates TD related reports in EOD.
34	1	End of Day	In_actn_post_eod_sh	Loan Post Action Due EOD Shell	LN	When a backdated index change is done and as part of the processing in EOD, it fails due to a proper business reason (funds not there or beyond redraw) as part of the replay process, the backdated rate change is skipped. Subsequently, when a payment happens on the account action for retrying the backdated rate change is logged on the account which gets picked up in Loan Post Action Due EOD Shell. It will get picked up if the action date happens to be a holiday.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
35	1	End of Day	dd_prdc_fee_shell	Demand Deposit Periodic Fee EOD Shell	DD	All periodic fees that applicable to CASA will be applied in EOD. For example, account keeping fees or debit card annual fees. Once SI is complete, the charges that need to be levied are available and this is then executed for all accounts, based on holiday processing parameter.
36	1	End of Day	ln_report_shell	Loan Reports	LN	Loan Related Reports
37	1	End of Day	dd_rev_swp_eod	Processing of Reverse Sweep	DD	This shell reverses any extra funds available in CASA account which is fetched during online sweep-in.
38	1	End of Day	ins_eod_action	Insurance EOD Shell	LM	This shell is used for insurance expiry alerts. Customer gets the notification about expired insurances.
39	1	End of Day	dd_eod_action1	Demand Deposit EOD Shell Non Alternate	DD	<p>Amortization of fees - Guided by holiday processing logic This shell runs actions of non-alternate and non-OD accounts and actions, except interest related, for non-alternate OD and non-OD accounts with alternate account. All transactions on alternate account are done before their EOD actions are performed. Interest related actions on OD accounts are performed in a separate shell as proportionate OD calculations are done after balance freeze (Proportionate OD calculation logic needs to be specified/doc linked). Following actions are performed as per holiday processing parameter:</p> <ul style="list-style-type: none"> ■ Execute any applicable rate changes (batch based) for the account. Holiday treatment: Next Working Day ■ Based on the cheques under clearing if there are any advance under unclear funds that need update that function is executed. Holiday treatment: Next Working Day ■ Computation of Minimum Average Balance. Holiday treatment: Next Working Day ■ Balance Change Action: After making adjustments for balances, system executes an interest computation action (Done in Interest Module). These snapshot balances are used to compute interest and also form the basis for the future audits and rebuild of balances when the back dated transactions are done. This is done only for accounts which are not OD.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<ul style="list-style-type: none"> ■ Credit and Debit Interest Settlement - All cases where capitalization of interest needs to be executed, are picked up and processed. Interest adjustment for back dated transactions happen. It is run for non-OD accounts. Holiday treatment: obtained from IRD. ■ Product based events log for alerts and outbound documentation. Holiday treatment: Next Working Day ■ Statement generation. Holiday treatment: Next Working Day ■ All status checks and updates are done such as inactivity, unclaimed, dormancy. Holiday treatment: Next Working Day ■ Reward Credit action is performed. Holiday treatment: Next Working Day ■ Offset linkage/expiry action is performed for both offset account and alternate account. Holiday treatment: Next Working Day
40	1	End of Day	dd_eod_action2	Demand Deposit EOD Shell Alternate	DD	This shell is run for all actions of accounts which are alternates and all actions except interest related actions for accounts which are alternate with OD facility. It will run the same set of actions as the shell above, based on holiday processing parameter.
41	1	End of Day	dd_eod_in_adjst	Demand Deposit Interest Adjustment shell	DD	This shell does Proportionate OD calculation and Offset balances calculation.
42	1	End of Day	dd_eod_action3	Demand Deposit EOD Shell Alternate OD	DD	<p>This shell is run for interest related actions of alternate accounts which are OD. Following actions are performed based on holiday processing parameter:</p> <ul style="list-style-type: none"> ■ Proportionate OD action: Calculation and stamping of proportionate OD and OL utilization. This action is done on any balance change on accounts which share the same facility. ■ Offset interest benefit: calculation of total offset balance for a beneficiary

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<p>account. This action will be done on balance change on any of the offset accounts.</p> <ul style="list-style-type: none"> ■ Intimate interest for computation: After the above two actions are executed, interest will be intimated for computation. ■ Credit and Debit Interest Settlement: All cases where capitalization of interest needs to be executed are picked up and processed. Interest adjustment for back dated transactions happens. This is run for accounts which are OD. ■ Statement generation
43	1	End of Day	dd_eod_action4	Demand Deposit EOD Shell Non Alternate OD	DD	This shell is run for interest related actions for non-alternate accounts which have OD and for non-OD accounts that have alternate. It runs the set of actions as the above shell, based on holiday processing parameter.
44	1	End of Day	lm_market_reval	Market Revaluation Shell	LM	This shell revaluates the Collaterals as per current market values.
45	1	End of Day	lm_util_reval	Utilization Revaluation	LM	If the utilization of the credit facility is done in a currency which is not the same as its currency, then revaluation process will be triggered as part of the End of the Day process.
46	1	End of Day	lm_batch_p_eod_shell	LCM EOD Pre Processor	LM	This shell is used to form a group of lcm eod actions for processing in multi streams.
47	1	End of Day	lm_batch_eod_shell	Limits And Collaterals Batch	LM	<p>Credit Facility Review: Credit Facility can be subjected to review based on the parameters set during the creation of the facility. On the review date, the facility is submitted to the rule engine for processing. The rule ID is maintained as part of the facility category. On successful completion the next review date, facility conduct is updated on the credit facility.</p> <p>Credit Facility Earmarking Expiry: LCM system provides an option to block/earmark the credit facility for a certain amount. This amount will not contribute towards the available amount of the facility. On the start date, the amount has to be moved into the earmarked amount column which will ensure</p>

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<p>that the amount will not be utilized. A reinstatement of the earmarked amount is to be done on the earmark expiry date.</p> <p>Credit Facility Transfer Expiry: LCM system provides option of transfer of the available amount of the facility to the another facility. The facility transfer is bound by a start and end date. On the facility transfer expiry date, the transfer amount needs to be reinstated to the facility. This is handled in the batch.</p> <p>Facility Amount Schedule Processing: System provides an option define the facility amount in schedules. The amount can be an increase/decrease to the existing sanctioned amount. The amount stored is the effective amount. On the start date, the effective amount maintained will become the effective Limit of the facility. On the end date, the amount will be removed and the new limit amount will be calculated.</p> <p>Credit Facility Availment Expiry Date: The credit facility created is expected to be utilized within a particular date. For example, first drawdown of the loan has to be done before the Availment expiry date. In case it is not done, the facility will be made unavailable.</p> <p>Facility Expiry Date Processing: The facility being created can have an expiry date. On the expiry date of the facility, system is expected to make the facility unavailable for further utilization.</p>
48	1	End of Day	dd_eod_action5	Demand Deposit EOD Shell Linkage Expiry	DD	This shell handles offset expiry actions.
49	1	End of Day	In_intrst_comp_shell	Loan Interest Computation EOD Shell	LN	This shell does Interest Computation (balance change).
50	1	End of Day	In_stmt_log_shell	Loan Statement logging Shell	LN	Statement Item generation based on frequency maintained at the product level Records from the last statement generation date to current date is copied to Statement Item. Statement Generation is not done in this shell. Data is copied from Loan Financial Statement to Statement Item.
51	1	End of Day	ac_bundle_exp_poller	Bundle Expiry Poller Shell	BN	This shell maintains Track Bundle expiry.
52	1	End of Day	dd_eod_offset_int	Demand Deposit Offset Interest EOD	DD	The shell calculates and credits interest, if eligible for offset saving accounts. Based on priority, if there are offset accounts whose balance is not used to offset loan or overdraft account, then the unused balance gets the credit

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
				Shell		interest.
53	1	End of Day	np_eod_acion	Asset Classification EOD action	NP	This shell processes all EOD actions of Asset Classification module using action framework.
54	1	End of Day	ch_eod_report_shell	CASA EOD Reports	CH	This shell generates CASA related batch reports.
55	1	End of Day	np_account_classify	Account Level Asset Classification	NP	This is the first and mandatory shell for asset classification which processes all the Loan, CASA, and OD accounts which are due for classification as per the current processing date. The process derives the expected classification code using the rule setup on the product class. The rule is based on arrears information received from the module along with other information like limit expiry, TOD or overline. Based on the classification code details, it will be arrived at whether the classification has to be Manual or Automatic. In case of Manual classification, the expected classification code is stamped and the process continues with the next account. In case of Automatic classification, the expected classification code and actual classification code will be stamped the same. Necessary accounting entries will be passed for Normal to Suspended or Suspended to Normal movements. The new classification code, classification status (Normal or Suspended) and clawback details (only in case of forward movement) will be sent to Loan/CASA/OD module. Customer alerts will be generated as required.
56	1	End of Day	In_pledge_acct_shell	Loan Pledged Account Action Shell	LN	
57	1	End of Day	np_party_classify	Party Level Asset Classification	NP	This shell runs immediately after account shell, if the asset classification applicability level is Party. This process first determines the worst classification of the party by comparing the expected code of all accounts under the party that have been classified today and actual code of the remaining accounts under the party. Based on the classification code details, it will be arrived at whether the classification has to be Manual or Automatic. In case of Manual classification, the worst classification code will be stamped as the expected classification code of the party and the same will be propagated as the expected code of all accounts under the party, and the process continues with the next party. In case of Automatic classification, the worst classification code will be stamped

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						as the expected classification code and actual classification code of the party. The same will be propagated as the expected classification code and actual classification code of all accounts under the party. For each account under the party, necessary accounting entries will be passed for Normal to Suspended or Suspended to Normal movements. The new classification code, classification status (Normal or Suspended) and clawback details (only in case of forward movement) will be sent to Loan/CASA/OD module. Customer alerts will be generated as required.
58	1	End of Day	np_facility_classify	Facility Level Asset Classification	NP	This shell runs immediately after account shell, if the asset classification applicability level is Facility. This process first determines the worst classification of the facility by comparing the expected code returned from the facility rule, the expected code of all accounts under the facility that have been classified today and actual code of the remaining accounts under the facility. Based on the classification code details, it will be arrived at whether the classification has to be Manual or Automatic. In case of Manual classification, the worst classification code will be stamped as the expected classification code of the facility and the same is propagated as the expected code of all accounts under the facility, and the process continues with the next facility. In case of Automatic classification, the worst classification code will be stamped as the expected classification code and actual classification code of the facility. The same is propagated as the expected classification code and actual classification code of all accounts under the facility. For each account under the facility, necessary accounting entries will be passed for Normal to Suspended or Suspended to Normal movements. The new classification code, classification status (Normal or Suspended) and clawback details (only in case of forward movement) will be sent to Loan/CASA/OD module. Customer alerts will be generated as required.
59	1	End of Day	eod_report_shell	EOD Reports	AL	This shell generates reports in EOD.
60	1	End of Day	as_eod_check	Accounting Verification	AS	This shell ensures that the Accounting events raised for the day are processed and all the balances updated. It also verifies whether the system is balanced (Credits and Debits match) for certain criteria for all the accounting entries raised and also for the period balance update.
61	1	End of Day	eod_epilogue	EOD Epilogue	EO	This shell waits for all the reports to be generated and changes the Process date.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
62	2	Beginning of Day	as_untank_adjustment	Untanking Adjustment Posting	AS	The shell is used to process Untanking Adjustment entries during the batch.
63	2	Beginning of Day	ins_bod_action	Insurance BOD Shell	LM	This shell is responsible for renewing the premium of CCI policy for next review period. If CCI is loan funded, then next review premium is charged on the nominated loan or OD account.
64	2	Beginning of Day	lz_pi_child_exmpt_sh	Exemption code BOD shell	PI	The shell is used to automatically update the Identification Type to TFN and Identification value to NO TFN of a party (onboarded as child) when the party turns 17 years. (Australia localization specific shell)
65	2	Beginning of Day	pc_business_transfr	Business Transfer Periodic Execution BOD Shell	PC	Business Transfer SI Execution Shell
66	2	Beginning of Day	pc_value_date_shell	Value Date BOD Shell	PC	This shell moves the funds from unclear to clear balance, where float of instrument is getting over at BOD. Value Date Clearing shell in BOD is run for more than 0 day float items. Hence it becomes mandatory to run this with the current process date, even though cut off has been run. This is taken care of by the Value Date Clearing Shell. As the float realization will be derived based on branch dates, the date will be always set to the working day of the branch.
67	2	Beginning of Day	pi_srv_ord_sta_bod	Service Order Activation BOD Shell	PI	This shell handles activation of Service Orders based on Service Order start date. (US localization specific shell)
68	2	Beginning of Day	py_auto_credit_bod	Payments Auto Credit Run BOD Shell	PY	This shell processes records for outward collection transactions and intimate the respective modules (CASA/LOANS) on customer value date configured in the Bank Float Definition. This shell is effective only if the Bank Float Definition is configured to not wait for network settlement in case of non-unclear balance method treatment.
69	2	Beginning of Day	py_cust_val_date_bod	Customer Value Date BOD Shell	PY	This shell will intimate the respective account modules (CASA\Loans\TD) on a given Customer Value Date for any valid transaction successfully posted in Payments module. The respective account module will make the funds

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						available for the customer on this date. This will process the transactions in BOD.
70	2	Beginning of Day	py_regcc_sch_rel_bod	Reg CC Schedule Release BOD Shell	PY	This shell will intimate the account module (Reg CC capable CASA accounts only) with fund value that has to be made available for the customer on this date as per the schedule generated in Reg CC Schedule Generation EOD Shell for the respective accounts. (US localization specific shell)
71	2	Beginning of Day	as_untank_bod_shell	Accounting Un-tanking Shell	AS	This shell is used to un-tank the accounting entries during batch process. Online Transactions during EOD do not process accounting entries but are tanked. This shell processes accounting of the tanked records.
72	2	Beginning of Day	lm_batch_p_bod_shell	LCM BOD Pre Processor	LM	This shell is used to form a group of lcm bod actions for processing in multi streams.
73	2	Beginning of Day	pc_activ_date_shell	Activation Date BOD Shell	PC	This shell takes the data of those transaction which are getting activated for processing for that day. For e.g- transaction like Outward collection, flat extended inward and outward clearing, future dated outgoing payments etc
74	2	Beginning of Day	lm_batch_bod_shell	Limits Bod Batch	LM	Same as Limits And Collaterals Batch shell in EOD Category
75	2	Beginning of Day	td_bod_action	Term Deposit BOD Shell	TD	This shell logs the following actions: <ul style="list-style-type: none"> ■ Maturity Processing - Maturity proceeds till yesterday will be credited to chosen settlement mode. ■ Future dated Redemption - Process the future dated redemption request due for that day. ■ Interest Payout - Interest payout to settlement mode. ■ Unclaim Processing - Movement of Unclaim TD to Unclaim GL.
76	2	Beginning of Day	ln_post_trans_bod	Loan Transaction BOD shell	LN	This shell processes all the transactions done after cutoff is run.
77	2	Beginning of	td_bod_	TD BOD	TD	This shell generates TD related reports in BOD.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
		Day	report_shell	Reports		
78	2	Beginning of Day	ln_contri_bod_shell	Loan Contribution BOD Shell	LN	This shell processes customer's Contribution Instruction set at time of Origination.
79	2	Beginning of Day	dd_bod_action	Demand Deposit BOD Shell	DD	<p>This shell runs in BOD before other CASA shells in BOD. It handles the following actions:</p> <ul style="list-style-type: none"> ■ Earmark expiry. Holiday treatment: Next Working Day ■ OD linkage action. Holiday treatment: Next Working Day ■ Offset account linkage action. Holiday treatment: Next Working Day
80	2	Beginning of Day	dd_bod_in_adjst	Demand Deposit BOD Interest Adjustment shell	DD	In case of OD, if facility is shared, the shell proportionates the utilized amount among all the stake holders.
81	2	Beginning of Day	dd_si_bod	DDA Standing Instruction BOD Shell	DD	This shell executes all Standing Instructions that need to be executed for the day - Based on holiday processing parameter
82	2	Beginning of Day	dd_swp_bod_default	DDA Sweepout Instruction BOD Shell Default Level	DD	Account may be set up to sweep the balance above a certain amount to other CASA or TD account. This is the primary shell where all accounts with priority 1 are executed. This is executed in BOD based on customer's preference.
83	2	Beginning of Day	dd_swp_bod_non_def	DDA Sweepout Instruction BOD Shell Non Default Level	DD	This shell behaves same as DDA Sweepout Instruction BOD Shell Default Level. But priority 2 and above sweep out instructions are executed. This is required to divide the priority 1 and priority 2 and above to avoid conflicts among accounts.
84	2	Beginning of Day	ac_stl_pyt_bod_shell	Account Settlement Payout BOD Shell	AC	This shell executes settlement pay out instructions that are maintained on CASA and Loan Accounts.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
85	2	Beginning of Day	In_split_bod_shell	Split Loan Account Opening BOD Shell	LN	Split Loan Account Opening and Disbursement request execution - Guided by holiday processing logic
86	2	Beginning of Day	In_actn_bod_shell	Loan Action due BOD shell	LN	<p>This shell does the following actions:</p> <ul style="list-style-type: none"> ■ Interest Charging ■ Interest Compounding ■ Interest Cap ■ Fee Cap ■ Customer Initiated Rollover ■ Auto Disbursement - Guided by holiday processing logic ■ Repayment Review (Apply and Generate) also called as periodic reschedule - Guided by holiday processing logic ■ Backdated PPF - These transactions will get processed as a part of host EOD. ■ Back dated installment - These transactions will get processed as a part of host EOD.
87	2	Beginning of Day	In_actn_post_bod_sh	Loan Post Action due BOD shell	LN	When a backdated index change is done and as part of the processing in EOD, it fails due to a proper business reason (funds not there or beyond redraw) as part of the replay process, the backdated rate change is skipped. Subsequently, when a payment happens on the account, action for retrying the backdated rate change is logged on the account which gets picked up in Loan Post Action Due BOD Shell.
88	2	Beginning of Day	ch_bod_report_shell	CASA BOD Reports	CH	This shell generates CASA Reports in BOD.
89	2	Beginning of Day	bod_report_shell	BOD Reports	AL	This shell generates reports.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
90	2	Beginning of Day	bod_epilogue	BOD Epilogue	EO	This shell waits for all the reports to be generated.
91	2	Beginning of Day	br_open_batch	Open Teller and Branch Batch Shell	BR	This batch job is for automation of opening of branch batches and non-cash teller batches for branches in current branch group.
92	2	Beginning of Day	dd_txn_untanking_sh	DDA Transaction Untanking BOD shell	DD	
93	3	Cut Off	br_batch_closure	Teller Branch Batch Closure	BR	This batch job is for automation of closure of non-cash teller batches and branch batches for branches in current branch group.
94	3	Cut Off	co_cutoff_prologue	Cutoff Prologue	CO	This shell is used to indicate that the cut off has been started by setting the flg_cutoff_run_today in flx_cs_branch_dates_b to 'Y'. (This flag is set back to 'N' at the end of the EOD category.)
95	10	Settlement Notification Processing	or_setl_notif_rtv	Settlement Notification Retrieval	OR	This shell retrieves Settlement Workspace Notifications.
96	10	Settlement Notification Processing	or_setl_notif_prc	Settlement Notification Processing	OR	This shell executed the Settlement Workspace Notifications.
97	12	FSDF master data hand-off	handoff_initialise	Analytics Hand-Off Initialisation Shell	DI	This shell initializes the start time of CSA (Common staging area) data handoff time frame.
98	12	FSDF master data hand-off	handoff_shell	Analytics Hand-Off Shell	DI	This shell initiates all ODI scenario execution defined in FLX_DI_ETL_JOB_DEFINITION table.
99	12	FSDF master data hand-off	epilogue_shell	Analytics epilogue Shell	DI	This shell checks all running ODI scenario execution status. If any error occurs, the scenario execution is restarted after resolve.
100	13	FSDF EOD data hand-off	handoff_shell	Analytics Hand-Off Shell	DI	This shell initiates all ODI scenario execution defined in FLX_DI_ETL_JOB_DEFINITION table.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
101	13	FSDF EOD data hand-off	epilogue_shell	Analytics epilogue Shell	DI	This shell checks all running ODI scenario execution status. If any error occurs, the scenario execution is restarted after resolve.
102	14	FSDF Txn data Hand-off	handoff_shell	Analytics Hand-Off Shell	DI	This shell initiates all ODI scenario execution defined in FLX_DI_ETL_JOB_DEFINITION table.
103	14	FSDF Txn data Hand-off	epilogue_shell	Analytics epilogue Shell	DI	This shell checks all running ODI scenario execution status. If any error occurs, the scenario execution is restarted after resolve.
104	15	Analytics Batch Data Hand-off	handoff_initialise	Analytics Hand-Off Initialisation Shell	DI	This shell initializes the start time of CSA (Common staging area) data handoff time frame.
105	15	Analytics Batch Data Hand-off	handoff_shell	Analytics Hand-Off Shell	DI	This shell initiates all ODI scenario execution defined in FLX_DI_ETL_JOB_DEFINITION table.
106	15	Analytics Batch Data Hand-off	epilogue_shell	Analytics epilogue Shell	DI	This shell checks all running ODI scenario execution status. If any error occurs, the scenario execution is restarted after resolve.
107	16	Internal System EOD	ac_actn_ieod_shell	Account Action Internal EOD Shell	LN	This shell processes all Internal EOD actions of Account module using action framework.
108	16	Internal System EOD	interest_eod_action	Interest Shell	IN	<p>This shell does following activities:</p> <ul style="list-style-type: none"> ■ Interest Accrual Process: When interest accrual is done, system raises events for the current working day, even though the cut off has been executed. This is required to enable posting of transaction on the current date and not for the next working day. It looks up at the host date. ■ Interest Capitalization: Based on holiday processing logic. ■ Interest Compounding: Based on holiday processing parameter. ■ Related tax events for any of the above functions, as configured, based on holiday processing parameter.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
109	16	Internal System EOD	lending_acct_stat	Lending Account Statistics Shell	AC	This shell is responsible for recording data required for reporting purposes of a Loan and LOC account. Delinquency, account status, arrears of an account are populated which is further extracted for reporting purposes.
110	16	Internal System EOD	pi_eod_shell	Party EOD Shell	PI	This shell is used to process the Party Due Diligence Expiry if the Expiry date is a holiday.
111	16	Internal System EOD	as_batch_verify	Accounting batch Verification Shell	AS	<p>There are certain checks and validations that are done as part of the Accounting in the EOD framework. These checks ensure that the system has balanced accounting entries, correct ledger balances and system account balances. Some of the checks are listed below:</p> <ul style="list-style-type: none"> ■ Ensure that all the accounting events raised for the day are processed. ■ Ensure that all the accounting entries generated for the day are updated for the ledger and system account balances. ■ Ensure that the accounting entries raised for the day are balanced. The balancing has to be branch wise and not group wise. ■ Ensure that period balances for ledger and system account are balanced.
112	16	Internal System EOD	pl_ledg_bal_upd	Deferred Ledger Balance Update	AS	For all the events raised during the batch process, the ledger balances will not be updated near online (no messages will be put into the Balance queue for processing). The ledger balances for such events will be Deferred and the balance will be updated through a procedure in the Internal EOD and as part of BOD (last batch in BOD). This is done to ensure that there is no lock on the ledgers when the balance update happens. (Scenario: when most of the accounts are mapped to the same product and hence update the same GL.)
113	16	Internal System EOD	pl_sdeadjmt_eod_shel	SDE Adjustment Shell	AS	This shell is used to adjust the system account balances if there are any change in the account, party and product attributes.
114	16	Internal System EOD	as_verify_prereval	Accounting batch Pre Revaluation Verify Shell	AS	<p>Some of the checks that are done are listed below:</p> <ul style="list-style-type: none"> ■ Ensure that all the accounting events raised for the day are processed. ■ Ensure that all the accounting entries generated for the day are updated for the ledger and system account balances.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<ul style="list-style-type: none"> ■ Ensure that the accounting entries raised for the day are balanced. The balancing has to be branch wise and not group wise. ■ Ensure that period balances for ledger and system account are balanced.
115	16	Internal System EOD	pl_reval_eod_shell	Revaluation Shell	AS	Ledgers whose balances are maintained in the Foreign Currency will be revalued (if required) based on a specific Reval Rate. This is done when all the batch transactions are completed in the OBP system, since the current/closing balance of the ledgers are used for this purpose.
116	16	Internal System EOD	as_verify_prefinclr	Accounting batch Pre Fin Closure Verify Shell	AS	Same as per accounting verification.
117	16	Internal System EOD	pl_fin_period_shell	Balance FIN Period Creation Shell	AS	It creates the system period FIN whenever applicable.
118	16	Internal System EOD	pl_finclose_eod_shel	Financial Closure Shell	AS	When the financial cycle changes, the system has to calculate the Profit/Loss for the current financial year and has to post to the Profit/Loss ledgers maintained. This will include the zeroization of the Income/Expense Ledgers and posting the profit or loss to the Profit/Loss Accounts maintained. This operation is done on the system period code FIN. All the accounting entries created for this purpose are posted in the FIN period.
119	16	Internal System EOD	as_verify_prprptglns	Accounting batch Pre Reporting Lines Verify Shell	AS	Same as per accounting verification
120	16	Internal System EOD	pl_rptglns_eod_shell	ReportingLines Shell	AS	If the bank requires Central Bank and Head Office Reporting system obtains the rule code maintained for the ledger, derives the Reporting GL and creates balances for the Reporting ledger.
121	16	Internal System EOD	as_verify_prearchiv	Accounting batch Pre Archive Verify Shell	AS	There are certain checks and validations that are done as part of the Accounting in the EOD framework. These validations are to ensure that the system has balanced accounting entries, correct ledger balances and system account balances. Some of the checks are listed below:

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
						<ul style="list-style-type: none"> ■ Ensure that all the accounting events raised for the day are processed. ■ Ensure that all the accounting entries generated for the day are updated for the ledger and system account balances. ■ Ensure that the accounting entries raised for the day are balanced. The balancing has to be branch wise and not group wise. ■ Ensure that period balances for ledger and system account are balanced. All the checks are with respect to each account branch and not branch group.
122	16	Internal System EOD	pl_intr_eod_rep_shel	PL Internal EOD Report Shell	PL	This shell generates Ledger related reports in EOD.
123	16	Internal System EOD	pl_balance_history	Periodic PL balance history	PL	This shell archives GL Period balances.
124	16	Internal System EOD	pl_period_bod_shell	Balance Period Creation Shell	AS	If there is a period change, then the system moves the period balances of the last month to the current month. The opening balances are updated and the movements for the current period is set to 0. The period balance update and the working of this shell are tightly coupled.
125	16	Internal System EOD	int_eod_report_shell	Internal EOD Reports	AL	This shell generates Reports during Internal system EOD.
126	16	Internal System EOD	int_eod_epilogue	Int EOD Epilogue	EO	This shell waits for all the reports to be generated.
127	17	Sales Offer Handoff	pm_sales_offer_hoff	Sales Offer Handoff Shell	PM	
128	17	Sales Offer Handoff	handoff_shell	Analytics Hand-Off Shell	DI	This shell initiates all ODI scenario execution defined in FLX_DI_ETL_JOB_DEFINITION table.
129	17	Sales Offer Handoff	epilogue_shell	Analytics epilogue Shell	DI	This shell checks all running ODI scenario execution status. If any error occurs, the scenario execution is restarted after resolve.
130	18	Clearing	pc_rej_	Scan Reject	PC	This shell reprocesses Reject Items.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
			reproc_shell	Reprocess Shell		
131	18	Clearing	pc_scn_pas_plr_shell	Scan Pass Poller Shell	PC	Honored Items Poller
132	18	Clearing	pc_oc_return_shell	Outgoing Clearing Return Shell	PC	OC Return Shell
133	19	CMD Hand-off	cmd_handoff_shell	CMD Hand-Off Shell	DI	This shell initiates ODI task for Datamart population for Credit Monitor Dashboard.
134	19	CMD Hand-off	cmd_epilogue_shell	CMD epilogue Shell	DI	This shell monitors and allows for restart of ODI task for Datamart population for Credit Monitor Dashboard.
135	23	Global Float	pc_glb_fit_ext_shell	Global Float Extension Shell	PC	Global Float Extension
136	24	CCM Scheduler	lm_titlesearch_shell	Collateral Title Search Report Fetch Shell	LM	
137	30	Scan Payment Queue Process	pc_scn_rej_plr_shell	Scan Reject Poller Shell	PC	Rejected Items Poller
138	63	Business Transfer	pc_b_colln_cdt_shell	Business Collection Credit Shell	PC	Business Collection Credit
139	63	Business Transfer	pc_b_pymt_dbt_shell	Business Payment Debit Shell	PC	Business Payment Debit
140	63	Business Transfer	pc_b_colln_dbt_shell	Business Collection Debit Shell	PC	Business Collection Debit

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
141	63	Business Transfer	pc_b_pymt_cdt_shell	Business Payment Credit Shell	PC	Business Payment Credit
142	69	mdm_Publish	mdm_publish_shell	mdm Publish	IN	<p>This shell publishes to OCH following status updates taken place during EOD:</p> <ul style="list-style-type: none"> ■ Account closures during batch execution ■ Account opening during batch execution ■ KYC updates during batch execution
143	100	Health Check	excep_pending_shell	Batch Exceptions Pending Check Shell	FW	This shell will check in exception log for all unprocessed records which marked as deferred. If any record is found, then this shell will be marked as aborted as there are still pending exception record available.
144	100	Health Check	health_chk_shell	Health Checkup Shell	FW	This is dummy shell. It is used to check whether framework is ready to process batch and to check if there is any issue on framework or not.
145	101	Metro2 reporting data population	m2_date_initialise	Metro2 Hand-Off Initialisation Shell	DI	This batch job is used to initiate shell to setup the date duration for metro2 data population. (US localization specific shell)
146	101	Metro2 reporting data population	m2_handoff_shell	Metro2 Hand-Off Shell	DI	This batch job is used to send ODI request to ODI server for metro2 data population. (US localization specific shell)
147	101	Metro2 reporting data population	m2_epilogue_shell	Metro2 epilogue Shell	DI	This batch job is used to monitor all the scenario execution status initiated during Metro2 Hand-Off Shell. (US localization specific shell)
148	102	Analytics Result Upload	rslt_handoff_shell	Analytics Result Upload Shell	DI	This shell initiates OFSAA LLFP result area data upload scenario execution.
149	102	Analytics Result Upload	rslt_epilogue_shell	Analytics Result Upload Epilogue Shell	DI	This shell checks OFSAA LLFP result area data upload scenario execution status.If any error occurs, the scenario execution is restarted after resolve.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
150	103	Debt cancellation data population	debtcan_initialize	Debtcan Hand-Off Initialisation Shell	DI	This batch job is used to initiate shell to setup the date duration for cancellation of debt (1099C) data population. (US localization specific shell)
151	103	Debt cancellation data population	debt_handoff_shell	Debtcan Hand-Off Shell	DI	This batch job is used to send ODI request to ODI server for for cancellation of debt (1099C) data population. (US localization specific shell)
152	103	Debt cancellation data population	debt_epilogue_shell	Debtcan epilogue Shell	DI	This batch job is used to monitor all the scenario execution status initiated during Debtcan Hand-Off Shell. (US localization specific shell)
153	117	Housekeeping	ac_arrear_history	Arrear history shell.	AC	This shell moves the details from FLX_AC_ARREAR_DETAILS_B to FLX_AC_ARREAR_DETAILS_H.
154	117	Housekeeping	ac_casa_housekeeping	CASA housekeeping shell.	AC	This shell updates the account statistics by moving the details to history in last day of the month.
155	117	Housekeeping	ac_stl_alrt_scdl	Procssing of Frequency based Alerts	DD	This shell executes the recurring alerts.
156	117	Housekeeping	acct_benefit_shell	Account Benefit Shell	AC	This shell calculates amount benefited due to offset account.
157	117	Housekeeping	as_hist_log_eod_shel	Accounting Transaction History Log Shell	AS	This shell archives accounting events and accounting entries into history table.
158	117	Housekeeping	ba_purge_shell_proc	Procedure based Purge shell to be run during Housekeeping	DD	This procedure based shell purges data with desired retention policy at configured purge frequency. This shell works on purge configuration (seed data) that stores purge related configuration like when to purge, how much data to retain, at what frequency, any additional where clause if any, whether or not to move data to history table, etc.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
159	117	Housekeeping	bd_unclaimed_shell	Bank Draft unclaimed shell	PC	The bank draft which has breached unclaimed period, is picked up and marked the instrument status as Unclaimed.
160	117	Housekeeping	da_event_h_eod_shel	Accounting Event History Shell	AS	This shell archives accounting events and accounting entries into history table.
161	117	Housekeeping	dd_hsk_actions	Processing of House Keeping shell	DD	This shell logs periodic alerts and maintain counters for CASA account.
162	117	Housekeeping	lm_auto_closur_shell	Facility Auto Closure Shell	LM	For non revolving facilities, auto closure will be done and all the records are picked up for processing in this shell.
163	117	Housekeeping	lm_exp_track_shell	Exposure Tracking Shell	LM	The exposures for the bank is tracked across the dimensions party and account. LCM tracks all the exposure of the transaction based on the facts that are created for the account and party. The exposure values are calculated and shown across each fact values. For example, the bank might require to track the exposure under different currency. The account fact is defined as the currency code and each transaction in the system will be tracked against the account currency and the exposures are calculated under each different currency.
164	117	Housekeeping	lm_insider_exp	Insider Exposure Tracking Shell	LM	This shell calculates the exposure of an insider. This data will be used while creating/amending the facility to check the total exposure of an insider for breach calculation.
165	117	Housekeeping	ln_adv_alert_shell	Loan Advance Alert Action BOD Shell	LN	This shell sends alerts based on Lead and Lag Days defined at product levels.
166	117	Housekeeping	move_drv_table_data	Move driver table data to History table	DD	This shell moves Driver table data into History Table.
167	117	Housekeeping	np_prvsn_updt_shell	Account Level Provision Update	NP	This shell is used for Account Level Provision Update.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
168	117	Housekeeping	or_pr_apr_exp_shell	Submission Pricing Approval EOD Expiry Shell	OR	This shell is use to mark status of Submission Level Pricing Approval as Expired.
169	117	Housekeeping	or_sub_exp_shell	Submission EOD Expiry Shell	OR	This shell expires the submission IDs which have breached the defined submission period.
170	117	Housekeeping	pi_bod_shell	Party BOD Shell	PI	This shell is used to process the Party Due Diligence Expiry if it falls on a working day. This shell also updates the future party address getting active on this day. It updates such addresses to be current and marks the previous current address as past.
171	117	Housekeeping	td_stmt_gen_shell	TD Statement Generation Shell	TD	This shell generates TD statements.
172	117	Housekeeping	ln_stmt_gen_shell	Loan Statement Generation EOD Shell	LN	This shell generates all the loan account statements.
173	117	Housekeeping	ln_stm_log_shell_bod	Loan Statement logging Shell BOD	LN	This batch job is responsible for logging of non-financial statement entries in case of a rate value change on an account.
174	117	Housekeeping	dd_stmt_gen_shell	CASA Statement Generation Shell	DD	This shell generates CASA statements.
175	117	Housekeeping	ac_pop_domain_ind	Indexing for Transactional Search	AC	This shell populates the domain index table, which is used for Oracle text search.
176	120	Alert Generation	ep_generation_shell	Alert generation Shell	DI	All pending alert requests is picked and processed. If the status of the alert is generated state, it means processing is pending for the alerts. This shell picks the alerts which are less than current date.

12.1 Batch Shells Description

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
177	125	Adhoc Scheduler Online	as_online_ledg_upd	Online Ledger Balance Update Shell	AS	This shell handles the Ledger balance posting of online accounting entries.
178	125	Adhoc Scheduler Online	dd_acct_online	Adhoc action shell for CASA	DD	
179	125	Adhoc Scheduler Online	dd_si_online	DDA No Retry Standing Instruction Online Shell	DD	
180	127	Human Task Action	wf_task_resume_shell	Human Task Resume Shell	WF	
181	128	Party Export	pi_export_handoff	Party Export Hand-Off Shell	DI	
182	128	Party Export	pi_export_epilogue	Party Export Epilogue Shell	DI	
183	247	Reprice	pr_acrl_batch_shell	Price Accrual Batch Shell	PR	This shell identifies accounts and performs fee accrual.
184	247	Reprice	pr_price_changes	Price change log online shell	PR	This shell identifies the accounts (which are not opened today) eligible for rate and fee changes and logs action for further processing in pr_reprice_action shell.
185	247	Reprice	pr_reprice_action	Reprice action online Shell	PR	This shell processes accounts (which are not opened today) for interest or fee changes happened in the system.
186	248	Payment outbound message online Shell	py_out_msg_action	Payment outbound message online Shell	PY	This shell is used to process the outbound messages.
187	250	OP Party Handoff Shell	py_op_party_handoff	OP Party handoff Shell	PY	This shell is used for OBP Party data handoff to OBPM.
188	251	OP Account	py_op_acct_	OP Account	PY	This shell is used for OBP Account data handoff to OBPM.

SR#	Process Category	Category Description	Shell	Shell Description	Module Code	Description
		Handoff Shell	handoff	handoff Shell		
189	352	Account Reprocess	or_acc_reprocess_sh	Account Reprocess Shell	OR	This shell is used to retry account opening for deposits in case of account opening failure.
190	70	Archival Category	sel_part_ch_nobook	Archival Shell	AR	

12.2 Batch Shells Execution Sequence

The following table presents the execution sequence of the batch shells.

Table 12–2 Shell Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
1	0	Reports Health Check	Optional	1	reports_chk_shell	Reports Health Check	FW		
2	1	End of Day	Mandatory	1	ac_action_relog_sh	Relog actions on holiday	AC		
3	1	End of Day	Mandatory	1	ac_bundle_fee_shell	Bundle Fee Shell	BN		
4	1	End of Day	Mandatory	1	bundle_report_shell	Bundle Reports	BN		
5	1	End of Day	Mandatory	1	dd_auto_statuschange	DDA Auto status change shell	DD		
6	1	End of Day	Mandatory	1	idl_actn_eod_shell	Indirect Lending Action EOD Shell	IL		
7	1	End of Day	Mandatory	1	ins_quote_exp_eod	Expire Quote EOD Shell	LM		

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
8	1	End of Day	Mandatory	1	je_susp_bal_shell	Journal Entry Suspense Balancing EOD Shell	JE		
9	1	End of Day	Mandatory	1	pc_blr_remit_shell	Biller Remittance Shell	PC		
10	1	End of Day	Mandatory	1	pc_value_date_eod_sh	Value Date EOD Shell	PC		
11	1	End of Day	Mandatory	1	pi_srv_ord_sta_eod	Service Order Deactivation EOD Shell (US localization specific shell)	PI		
12	1	End of Day	Mandatory	1	py_bank_val_date_eod	Bank Value Date EOD Shell	PY		
13	1	End of Day	Mandatory	1	py_regcc_sch_gen	Reg CC Schedule Generation EOD Shell (US localization specific shell)	PY		
14	1	End of Day	Mandatory	1	py_regcc_sch_rel_eod	Reg CC Schedule Release EOD Shell (US localization specific shell)	PY		
15	1	End of Day	Mandatory	2	dd_inst_rearrg_sh	DD Instruction Rearrangement For Calender Change	DD	ac_action_relog_sh	Relog actions on holiday
16	1	End of Day	Mandatory	2	je_batch_hist_shell	Journal Entry Batch History EOD Shell	JE	je_susp_bal_shell	Journal Entry Suspense Balancing EOD Shell
17	1	End of Day	Mandatory	2	pl_onl_ledg_bal_upd	Online Ledger Entries Update	AS	je_susp_bal_shell	Journal Entry Suspense Balancing EOD Shell

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
18	1	End of Day	Mandatory	2	pc_report_shell	PC Reports	PC	pc_value_date_eod_sh	Value Date EOD Shell
19	1	End of Day	Mandatory	2	py_cust_val_date_eod	Customer Value Date EOD Shell	PY	py_bank_val_date_eod	Bank Value Date EOD Shell
20	1	End of Day	Mandatory	3	eod_preval	EOD pre validation Shell	DD	pc_report_shell	PC Reports
21	1	End of Day	Mandatory	4	pr_price_changes_eod	Price change action log shell for account opened today	PR	eod_preval	EOD pre validation Shell
22	1	End of Day	Mandatory	5	pr_reprice_acn_eod	Reprice Action shell for account opened today	PR	pr_price_changes_eod	Price change action log shell for account opened today
23	1	End of Day	Mandatory	6	dd_si_eod	DDA Standing Instruction EOD Shell	DD	ac_bundle_fee_shell dd_auto_statuschange pr_reprice_acn_eod	Bundle Fee Shell DDA Auto status change shell Reprice Action shell for account opened today
24	1	End of Day	Mandatory	7	dd_swp_eod_default	DDA Sweepout Instruction EOD Shell Default Level	DD	dd_si_eod	DDA Standing Instruction EOD Shell
25	1	End of Day	Mandatory	8	dd_swp_eod_non_def	DDA Sweepout Instruction EOD Shell Non Default Level	DD	dd_swp_eod_default	DDA Sweepout Instruction EOD Shell Default Level
26	1	End of Day	Mandatory	9	dd_swp_hold_remove	SweepIn Hold Removal shell	DD	dd_swp_eod_non_def	DDA Sweepout Instruction EOD Shell Non Default Level
27	1	End of Day	Mandatory	9	ln_contri_eod_shell	Loan Contribution EOD Shell	LN	dd_swp_eod_non_def	DDA Sweepout Instruction EOD Shell Non Default Level

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
28	1	End of Day	Mandatory	10	ac_stl_pyt_eod_shell	Account Settlement Payout EOD Shell	AC	In_contri_eod_shell	Loan Contribution EOD Shell
29	1	End of Day	Mandatory	10	td_remove_swpin_lien	TD Remove Sweep In Shell	TD	dd_swp_hold_remove	SweepIn Hold Removal shell
30	1	End of Day	Mandatory	11	In_split_eod_shell	Split Loan Account Opening EOD Shell	LN	ac_stl_pyt_eod_shell	Account Settlement Payout EOD Shell
31	1	End of Day	Mandatory	11	td_eod_action	Term Deposit EOD Shell	TD	td_remove_swpin_lien	TD Remove Sweep In Shell
32	1	End of Day	Mandatory	12	In_actn_eod_shell	Loans Action Due EOD Shell	LN	In_split_eod_shell	Split Loan Account Opening EOD Shell
33	1	End of Day	Mandatory	12	td_eod_report_shell	TD EOD Reports	TD	td_eod_action	Term Deposit EOD Shell
34	1	End of Day	Mandatory	13	In_actn_post_eod_sh	Loan Post Action Due EOD Shell	LN	In_actn_eod_shell	Loans Action Due EOD Shell
35	1	End of Day	Mandatory	14	dd_prdc_fee_shell	Demand Deposit Periodic Fee EOD Shell	DD	In_actn_post_eod_sh	Loan Post Action Due EOD Shell
36	1	End of Day	Mandatory	14	In_report_shell	Loan Reports	LN	In_actn_post_eod_sh	Loan Post Action Due EOD Shell
37	1	End of Day	Mandatory	15	dd_rev_swp_eod	Processing of Reverse Sweep	DD	dd_prdc_fee_shell	Demand Deposit Periodic Fee EOD Shell
38	1	End of Day	Mandatory	15	ins_eod_action	Insurance EOD Shell	LM	In_report_shell	Loan Reports
39	1	End of Day	Mandatory	16	dd_eod_action1	Demand Deposit EOD Shell Non Alternate	DD	dd_rev_swp_eod	Processing of Reverse Sweep
40	1	End of Day	Mandatory	17	dd_eod_action2	Demand Deposit EOD Shell Alternate	DD	dd_eod_action1	Demand Deposit EOD Shell Non Alternate

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
41	1	End of Day	Mandatory	18	dd_eod_in_adjst	Demand Deposit Interest Adjustment shell	DD	dd_eod_action2	Demand Deposit EOD Shell Alternate
42	1	End of Day	Mandatory	19	dd_eod_action3	Demand Deposit EOD Shell Alternate OD	DD	dd_eod_in_adjst	Demand Deposit Interest Adjustment shell
43	1	End of Day	Mandatory	20	dd_eod_action4	Demand Deposit EOD Shell Non Alternate OD	DD	dd_eod_action3	Demand Deposit EOD Shell Alternate OD
44	1	End of Day	Mandatory	21	lm_market_reval	Market Revaluation Shell	LM	dd_eod_action4	Demand Deposit EOD Shell Non Alternate OD
45	1	End of Day	Mandatory	22	lm_util_reval	Utilization Revaluation	LM	lm_market_reval	Market Revaluation Shell
46	1	End of Day	Mandatory	23	lm_batch_p_eod_shell	LCM EOD Pre Processor	LM	lm_util_reval	Utilization Revaluation
47	1	End of Day	Mandatory	24	lm_batch_eod_shell	Limits And Collaterals Batch	LM	lm_batch_p_eod_shell	LCM EOD Pre Processor
48	1	End of Day	Mandatory	25	dd_eod_action5	Demand Deposit EOD Shell Linkage Expiry	DD	lm_batch_eod_shell	Limits And Collaterals Batch
49	1	End of Day	Mandatory	26	ln_intrst_comp_shell	Loan Interest Computation EOD Shell	LN	dd_eod_action5	Demand Deposit EOD Shell Linkage Expiry
50	1	End of Day	Mandatory	26	ln_stmt_log_shell	Loan Statement logging Shell	LN	dd_eod_action5	Demand Deposit EOD Shell Linkage Expiry
51	1	End of Day	Mandatory	27	ac_bundle_exp_poller	Bundle Expiry Poller Shell	BN	ln_intrst_comp_shell	Loan Interest Computation EOD Shell

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
52	1	End of Day	Mandatory	27	dd_eod_offset_int	Demand Deposit Offset Interest EOD Shell	DD	In_intrst_comp_shell	Loan Interest Computation EOD Shell
53	1	End of Day	Mandatory	27	np_eod_acion	Asset Classification EOD action	NP	In_intrst_comp_shell	Loan Interest Computation EOD Shell
54	1	End of Day	Mandatory	28	ch_eod_report_shell	CASA EOD Reports	CH	dd_eod_offset_int	Demand Deposit Offset Interest EOD Shell
55	1	End of Day	Mandatory	28	np_account_classify	Account Level Asset Classification	NP	np_eod_acion	Asset Classification EOD action
56	1	End of Day	Mandatory	29	In_pledge_acct_shell	Loan Pledged Account Action Shell	LN	np_account_classify	Account Level Asset Classification
57	1	End of Day	Mandatory	29	np_party_classify	Party Level Asset Classification	NP	np_account_classify	Account Level Asset Classification
58	1	End of Day	Mandatory	30	np_facility_classify	Facility Level Asset Classification	NP	np_party_classify	Party Level Asset Classification
59	1	End of Day	Mandatory	31	eod_report_shell	EOD Reports	AL	bundle_report_shell ch_eod_report_shell np_facility_classify	Bundle Reports CASA EOD Reports Facility Level Asset Classification
60	1	End of Day	Mandatory	32	as_eod_check	Accounting Verification	AS	eod_report_shell td_eod_report_shell ch_eod_report_shell	EOD Reports TD EOD Reports CASA EOD Reports
61	1	End of Day	Mandatory	33	eod_epilogue	EOD Epilogue	EO	as_eod_check	Accounting Verification

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
62	2	Beginning of Day	Mandatory	1	as_untank_adjustment	Untanking Adjustment Posting	AS		
63	2	Beginning of Day	Mandatory	1	ins_bod_action	Insurance BOD Shell	LM		
64	2	Beginning of Day	Mandatory	1	lz_pi_child_exmpt_sh	Exemption code BOD shell (Australia localization specific shell)	PI		
65	2	Beginning of Day	Mandatory	1	pc_business_transfr	Business Transfer Periodic Execution BOD Shell	PC		
66	2	Beginning of Day	Mandatory	1	pc_value_date_shell	Value Date BOD Shell	PC		
67	2	Beginning of Day	Mandatory	1	pi_srv_ord_sta_bod	Service Order Activation BOD Shell (US localization specific shell)	PI		
68	2	Beginning of Day	Mandatory	1	py_auto_credit_bod	Payments Auto Credit Run BOD Shell	PY		
69	2	Beginning of Day	Mandatory	1	py_cust_val_date_bod	Customer Value Date BOD Shell	PY		
70	2	Beginning of Day	Mandatory	1	py_regcc_sch_rel_bod	Reg CC Schedule Release BOD Shell (US localization specific shell)	PY		
71	2	Beginning of Day	Mandatory	2	as_untank_bod_shell	Accounting Un-tanking Shell	AS	as_untank_adjustment	Untanking Adjustment Posting
72	2	Beginning of Day	Mandatory	2	lm_batch_p_bod_shell	LCM BOD Pre Processor	LM	ins_bod_action	Insurance BOD Shell

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
73	2	Beginning of Day	Mandatory	2	pc_activ_date_shell	Activation Date BOD Shell	PC	pc_value_date_shell	Value Date BOD Shell
74	2	Beginning of Day	Mandatory	3	lm_batch_bod_shell	Limits Bod Batch	LM	lm_batch_p_bod_shell	LCM BOD Pre Processor
75	2	Beginning of Day	Mandatory	3	td_bod_action	Term Deposit BOD Shell	TD	pc_activ_date_shell	Activation Date BOD Shell
76	2	Beginning of Day	Mandatory	4	ln_post_trans_bod	Loan Transaction BOD shell	LN	td_bod_action	Term Deposit BOD Shell
77	2	Beginning of Day	Mandatory	4	td_bod_report_shell	TD BOD Reports	TD	td_bod_action	Term Deposit BOD Shell
78	2	Beginning of Day	Mandatory	5	ln_contri_bod_shell	Loan Contribution BOD Shell	LN	ln_post_trans_bod	Loan Transaction BOD shell
79	2	Beginning of Day	Mandatory	6	dd_bod_action	Demand Deposit BOD Shell	DD	ins_bod_action lm_batch_bod_shell ln_contri_bod_shell	Insurance BOD Shell Limits Bod Batch Loan Contribution BOD Shell
80	2	Beginning of Day	Mandatory	7	dd_bod_in_adjst	Demand Deposit BOD Interest Adjustment shell	DD	dd_bod_action	Demand Deposit BOD Shell
81	2	Beginning of Day	Mandatory	8	dd_si_bod	DDA Standing Instruction BOD Shell	DD	dd_bod_in_adjst	Demand Deposit BOD Interest Adjustment shell
82	2	Beginning of Day	Mandatory	9	dd_swp_bod_default	DDA Sweepout Instruction BOD Shell Default Level	DD	dd_si_bod	DDA Standing Instruction BOD Shell
83	2	Beginning of Day	Mandatory	10	dd_swp_bod_non_def	DDA Sweepout Instruction BOD Shell Non Default Level	DD	dd_swp_bod_default	DDA Sweepout Instruction BOD Shell Default Level

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
84	2	Beginning of Day	Mandatory	11	ac_stl_pyt_bod_shell	Account Settlement Payout BOD Shell	AC	dd_swp_bod_non_def	DDA Sweepout Instruction BOD Shell Non Default Level
85	2	Beginning of Day	Mandatory	12	ln_split_bod_shell	Split Loan Account Opening BOD Shell	LN	ac_stl_pyt_bod_shell	Account Settlement Payout BOD Shell
86	2	Beginning of Day	Mandatory	13	ln_actn_bod_shell	Loan Action due BOD shell	LN	ln_split_bod_shell	Split Loan Account Opening BOD Shell
87	2	Beginning of Day	Mandatory	14	ln_actn_post_bod_sh	Loan Post Action due BOD shell	LN	ln_actn_bod_shell	Loan Action due BOD shell
88	2	Beginning of Day	Mandatory	15	ch_bod_report_shell	CASA BOD Reports	CH	ln_actn_post_bod_sh	Loan Post Action due BOD shell
89	2	Beginning of Day	Mandatory	16	bod_report_shell	BOD Reports	AL	ch_bod_report_shell	CASA BOD Reports
90	2	Beginning of Day	Mandatory	17	bod_epilogue	BOD Epilogue	EO	bod_report_shell ln_actn_post_bod_sh	BOD Reports Loan Post Action due BOD shell
91	2	Beginning of Day	Mandatory	18	br_open_batch	Open Teller and Branch Batch Shell	BR		
92	2	Beginning of Day	Mandatory	19	dd_txn_untanking_sh	DDA Transaction Untanking BOD Shell	DD		
93	3	Cut Off	Mandatory	1	br_batch_closure	Teller Branch Batch Closure	BR		
94	3	Cut Off	Mandatory	2	co_cutoff_prologue	Cutoff Prologue	CO		
95	10	Settlement Notification Processing	Optional	1	or_setl_notif_rtv	Settlement Notification Retrieval	OR		

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
96	10	Settlement Notification Processing	Optional	2	or_setl_notif_prc	Settlement Notification Processing	OR	or_setl_notif_rtv	Settlement Notification Retrieval
97	12	FSDF master data hand-off	Optional	1	handoff_initialise	Analytics Hand-Off Initialisation Shell	DI	co_cutoff_prologue	Cutoff Prologue
98	12	FSDF master data hand-off	Optional	2	handoff_shell	Analytics Hand-Off Shell	DI	handoff_initialise	Analytics Hand-Off Initialisation Shell
99	12	FSDF master data hand-off	Optional	3	epilogue_shell	Analytics epilogue Shell	DI	handoff_shell	Analytics Hand-Off Shell
100	13	FSDF EOD data hand-off	Optional	1	handoff_shell	Analytics Hand-Off Shell	DI		
101	13	FSDF EOD data hand-off	Optional	2	epilogue_shell	Analytics epilogue Shell	DI	handoff_shell	Analytics Hand-Off Shell
102	14	FSDF Txn data Hand-off	Optional	1	handoff_shell	Analytics Hand-Off Shell	DI		
103	14	FSDF Txn data Hand-off	Optional	2	epilogue_shell	Analytics epilogue Shell	DI	handoff_shell	Analytics Hand-Off Shell
104	15	Analytics Batch Data Hand-off	Optional	1	handoff_initialise	Analytics Hand-Off Initialisation Shell	DI		
105	15	Analytics Batch Data Hand-off	Optional	2	handoff_shell	Analytics Hand-Off Shell	DI	handoff_initialise	Analytics Hand-Off Initialisation Shell
106	15	Analytics Batch Data Hand-off	Optional	3	epilogue_shell	Analytics epilogue Shell	DI	handoff_shell	Analytics Hand-Off Shell
107	16	Internal System EOD	Mandatory	1	ac_actn_ieod_shell	Account Action Internal EOD Shell	LN		
108	16	Internal System EOD	Mandatory	1	interest_eod_action	Interest Shell	IN		

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
109	16	Internal System EOD	Mandatory	1	lending_acct_stat	Lending Account Statistics Shell	AC		
110	16	Internal System EOD	Mandatory	1	pi_eod_shell	Party EOD Shell	PI		
111	16	Internal System EOD	Mandatory	2	as_batch_verify	Accounting batch Verification Shell	AS	interest_eod_action	Interest Shell
112	16	Internal System EOD	Mandatory	3	pl_ledg_bal_upd	Deferred Ledger Balance Update	AS	as_batch_verify	Accounting batch Verification Shell
113	16	Internal System EOD	Mandatory	4	pl_sdeadjmt_eod_shel	SDE Adjustment Shell	AS	pl_ledg_bal_upd	Deferred Ledger Balance Update
114	16	Internal System EOD	Mandatory	5	as_verify_prereval	Accounting batch Pre Revaluation Verify Shell	AS	pl_sdeadjmt_eod_shel	SDE Adjustment Shell
115	16	Internal System EOD	Mandatory	6	pl_reval_eod_shell	Revaluation Shell	AS	as_verify_prereval	Accounting batch Pre Revaluation Verify Shell
116	16	Internal System EOD	Mandatory	7	as_verify_prefinclr	Accounting batch Pre Fin Closure Verify Shell	AS	pl_reval_eod_shell	Revaluation Shell
117	16	Internal System EOD	Mandatory	8	pl_fin_period_shell	Balance FIN Period Creation Shell	AS	as_verify_prefinclr	Accounting batch Pre Fin Closure Verify Shell
118	16	Internal System EOD	Mandatory	9	pl_finclose_eod_shel	Financial Closure Shell	AS	pl_fin_period_shell	Balance FIN Period Creation Shell
119	16	Internal System EOD	Mandatory	10	as_verify_prerptglns	Accounting batch Pre Reporting Lines Verify Shell	AS	pl_finclose_eod_shel	Financial Closure Shell
120	16	Internal System EOD	Mandatory	11	pl_rptglns_eod_shell	ReportingLines Shell	AS	as_verify_prerptglns	Accounting batch Pre Reporting Lines Verify Shell

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
121	16	Internal System EOD	Mandatory	12	as_verify_prearchiv	Accounting batch Pre Archive Verify Shell	AS	pl_rptglns_eod_shell	ReportingLines Shell
122	16	Internal System EOD	Mandatory	13	pl_intr_eod_rep_shel	PL Internal EOD Report Shell	PL	as_verify_prearchiv	Accounting batch Pre Archive Verify Shell
123	16	Internal System EOD	Mandatory	14	pl_balance_history	Periodic PL balance history	PL	pl_intr_eod_rep_shel	PL Internal EOD Report Shell
124	16	Internal System EOD	Mandatory	15	pl_period_bod_shell	Balance Period Creation Shell	AS	pl_balance_history	Periodic PL balance history
125	16	Internal System EOD	Mandatory	16	int_eod_report_shell	Internal EOD Reports	AL	pl_period_bod_shell	Balance Period Creation Shell
126	16	Internal System EOD	Mandatory	17	int_eod_epilogue	Int EOD Epilogue	EO	int_eod_report_shell	Internal EOD Reports
127	17	Sales Offer Handoff	Optional	1	pm_sales_offer_hoff	Sales Offer Handoff Shell	PM		
128	17	Sales Offer Handoff	Optional	2	handoff_shell	Analytics Hand-Off Shell	DI	pm_sales_offer_hoff	Sales Offer Handoff Shell
129	17	Sales Offer Handoff	Optional	3	epilogue_shell	Analytics epilogue Shell	DI	handoff_shell	Analytics Hand-Off Shell
130	18	Clearing	Optional	1	pc_rej_reproc_shell	Scan Reject Reprocess Shell	PC		
131	18	Clearing	Optional	1	pc_scn_pas_plr_shell	Scan Pass Poller Shell	PC		
132	18	Clearing	Optional	2	pc_oc_return_shell	Outgoing Clearing Return Shell	PC	pc_rej_reproc_shell	Scan Reject Reprocess Shell
133	19	CMD Hand-off	Optional	1	cmd_handoff_shell	CMD Hand-Off Shell	DI		
134	19	CMD Hand-off	Optional	2	cmd_	CMD epilogue Shell	DI	cmd_	CMD Hand-Off Shell

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
					epilogue_shell			handoff_shell	
135	23	Global Float	Optional	1	pc_glbflt_ext_shell	Global Float Extension Shell	PC		
136	24	CCM Scheduler	Optional	1	lm_titlesearch_shell	Collateral Title Search Report Fetch Shell	LM		
137	30	Scan Payment Queue Process	Optional	1	pc_scnrej_plr_shell	Scan Reject Poller Shell	PC		
138	63	Business Transfer	Optional	1	pc_b_colln_cdt_shell	Business Collection Credit Shell	PC		
139	63	Business Transfer	Optional	1	pc_b_pynt_dbt_shell	Business Payment Debit Shell	PC		
140	63	Business Transfer	Optional	2	pc_b_colln_dbt_shell	Business Collection Debit Shell	PC	pc_b_colln_cdt_shell	Business Collection Credit Shell
141	63	Business Transfer	Optional	2	pc_b_pynt_cdt_shell	Business Payment Credit Shell	PC	pc_b_pynt_dbt_shell	Business Payment Debit Shell
142	69	mdm_Publish	Optional	1	mdm_publish_shell	mdm Publish	IN		
143	100	Health Check	Optional	1	excep_pending_shell	Batch Exceptions Pending Check Shell	FW		
144	100	Health Check	Optional	1	health_chk_shell	Health Checkup Shell	FW		
145	101	Metro2 reporting data population	Optional	1	m2_date_initialise	Metro2 Hand-Off Initialisation Shell (US localization specific shell)	DI		
146	101	Metro2 reporting	Optional	2	m2_handoff_	Metro2 Hand-Off Shell	DI	m2_date_	Metro2 Hand-Off

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
		data population			shell	(US localization specific shell)		initialise	Initialisation Shell
147	101	Metro2 reporting data population	Optional	3	m2_epilogue_shell	Metro2 epilogue Shell (US localization specific shell)	DI	m2_handoff_shell	Metro2 Hand-Off Shell
148	102	Analytics Result Upload	Optional	1	rslt_handoff_shell	Analytics Result Upload Shell	DI		
149	102	Analytics Result Upload	Optional	2	rslt_epilogue_shell	Analytics Result Upload Epilogue Shell	DI	rslt_handoff_shell	Analytics Result Upload Shell
150	103	Debt cancellation data population	Optional	1	debtcan_initialise	Debtcan Hand-Off Initialisation Shell (US localization specific shell)	DI		
151	103	Debt cancellation data population	Optional	2	debt_handoff_shell	Debtcan Hand-Off Shell (US localization specific shell)	DI	debtcan_initialise	Debtcan Hand-Off Initialisation Shell
152	103	Debt cancellation data population	Optional	3	debt_epilogue_shell	Debtcan epilogue Shell (US localization specific shell)	DI	debt_handoff_shell	Debtcan Hand-Off Shell
153	117	Housekeeping	Mandatory	1	ac_arrear_history	Arrear history shell.	AC		
154	117	Housekeeping	Mandatory	1	ac_casa_housekeeping	CASA housekeeping shell.	AC		
155	117	Housekeeping	Mandatory	1	ac_stl_alrt_scdl	Procssing of Frequency based Alerts	DD		
156	117	Housekeeping	Mandatory	1	acct_benefit_shell	Account Benefit Shell	AC		

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
157	117	Housekeeping	Mandatory	1	as_hist_log_eod_shell	Accounting Transaction History Log Shell	AS		
158	117	Housekeeping	Mandatory	1	ba_purge_shell_proc	Procedure based Purge shell to be run during Housekeeping	DD		
159	117	Housekeeping	Mandatory	1	bd_unclaimed_shell	Bank Draft unclaimed shell	PC		
160	117	Housekeeping	Mandatory	1	da_event_h_eod_shell	Accounting Event History Shell	AS		
161	117	Housekeeping	Mandatory	1	dd_hsk_actions	Processing of House Keeping shell	DD		
162	117	Housekeeping	Mandatory	1	lm_auto_closur_shell	Facility Auto Closure Shell	LM		
163	117	Housekeeping	Mandatory	1	lm_exp_track_shell	Exposure Tracking Shell	LM		
164	117	Housekeeping	Mandatory	1	lm_insider_exp	Insider Exposure Tracking Shell	LM		
165	117	Housekeeping	Mandatory	1	ln_adv_alert_shell	Loan Advance Alert Action BOD Shell	LN		
166	117	Housekeeping	Mandatory	1	move_drv_table_data	Move driver table data to History table	DD		
167	117	Housekeeping	Mandatory	1	np_prvsn_updt_shell	Account Level Provision Update	NP		
168	117	Housekeeping	Mandatory	1	or_pr_apr_exp_shell	Submission Pricing Approval EOD Expiry Shell	OR		
169	117	Housekeeping	Mandatory	1	or_sub_exp_	Submission EOD Expiry	OR		

12.2 Batch Shells Execution Sequence

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
					shell	Shell			
170	117	Housekeeping	Mandatory	1	pi_bod_shell	Party BOD Shell	PI		
171	117	Housekeeping	Mandatory	1	td_stmt_gen_shell	TD Statement Generation Shell	TD		
172	117	Housekeeping	Mandatory	2	ln_stmt_gen_shell	Loan Statement Generation EOD Shell	LN	td_stmt_gen_shell	TD Statement Generation Shell
173	117	Housekeeping	Mandatory	3	ln_stm_log_shell_bod	Loan Statement logging Shell BOD	LN	ln_stmt_gen_shell	Loan Statement Generation EOD Shell
174	117	Housekeeping	Mandatory	4	dd_stmt_gen_shell	CASA Statement Generation Shell	DD	ln_stm_log_shell_bod	Loan Statement logging Shell BOD
175	117	Housekeeping	Mandatory	5	ac_pop_domain_ind	Indexing for Transactional Search	AC	dd_stmt_gen_shell	CASA Statement Generation Shell
176	120	Alert Generation	Optional	1	ep_generation_shell	Alert generation Shell	DI		
177	125	Adhoc Scheduler Online	Optional	1	as_online_ledg_upd	Online Ledger Balance Update Shell	AS		
178	125	Adhoc Scheduler Online	Optional	1	dd_acct_online	Adhoc action shell for CASA	DD		
179	125	Adhoc Scheduler Online	Optional	1	dd_si_online	DDA No Retry Standing Instruction Online Shell	DD		
180	127	Human Task Action	Optional	1	wf_task_resume_shell	Human Task Resume Shell	WF		
181	128	Party Export	Optional	1	pi_export_handoff	Party Export Hand-Off Shell	DI		
182	128	Party Export	Optional	2	pi_export_epilogue	Party Export Epilogue Shell	DI	pi_export_handoff	Party Export Hand-Off Shell

SR#	Process Category	Category Description	Category Significance	Shell Execution Sequence	Shell	Shell Description	Module Code	Required Shells	Required Shell Description
183	247	Reprice	Default	1	pr_acrl_batch_shell	Price Accrual Batch Shell	PR		
184	247	Reprice	Default	1	pr_price_changes	Price change log online shell	PR		
185	247	Reprice	Default	2	pr_reprice_action	Reprice action online Shell	PR	pr_price_changes	Price change log online shell
186	248	Payment outbound message online Shell	Optional	1	py_out_msg_action	Payment outbound message online Shell	PY		
187	250	OP Party Handoff Shell	Optional	1	py_op_party_handoff	OP Party handoff Shell	PY		
188	251	OP Account Handoff Shell	Optional	1	py_op_acct_handoff	OP Account handoff Shell	PY		
189	352	Account Reprocess	Optional	1	or_acc_reprocess_sh	Account Reprocess Shell	OR		
190	70	Archival Category	Optional	1	sel_part_ch_nobook	Archival Shell	AR		

13 Information Lifecycle Management (ILM)

This chapter describes the configuration, installation, and policy setup of Information Lifecycle Management (ILM).

Information Lifecycle Management is a set of techniques and technologies available from Oracle that assist in managing the lifecycle of data to support business needs and minimize storage costs. OBP drives ILM at the Oracle database level using database options and features to manage and move data as it evolves during its lifetime.

13.1 Configuration

The following values for the duration of data retention need to be determined. These values are used to drive ILM configuration.

Table 13–1 Values for ILM Configuration

Pattern Name	Partition Range Type	Data Retention in Active Tier	Data Retention in Less Active Tier	Data Retention in Historical Tier	Purge After
Lifecycle_Pattern_1	MONTH	2 month	N/A	2 year	2 year
Lifecycle_Pattern_2	MONTH	6 month	Will be provided by business	N/A	Will be provided by business
Lifecycle_Pattern_3	YEAR	N/A	N/A	N/A	Will be provided by business
Lifecycle_Pattern_4	YEAR	1 year	Will be provided by business		N/A
Lifecycle_Pattern_5	YEAR	1 year	N/A	Will be provided by business	
Lifecycle_Pattern_6	MONTH	1 month			1 month

13.2 Installation

This section explains the process of ILM installation.

13.2.1 Prepare Scripts

Operator needs to create partition creation script and ADO policy creation script manually based on data provided in ILM_Config.xlsx for each ILM qualified table and attached lifecycle pattern.

Parameters required for populating partition creation script are as follows:

- Table Name (OBP Tables Worksheet)
- ILM Column (OBP Tables Worksheet)
- Partition Range Type (Lifecycle Pattern Worksheet)

Parameters required for populating ADO policy creation script are as follows:

- Table Name (OBP Tables Worksheet)
- Lifecycle Definition (OBP Tables Worksheet)
- Data retention in different tier (Lifecycle Pattern Worksheet)
- Purging time (Lifecycle Pattern Worksheet)

The following sections describe the steps to be performed during the OBP database creation.

13.2.2 Create Tablespace

Separate tablespaces need to be created for the following tiers:

- Active tier
- Less Active tier
- Historical tier

The following command is to be used for creation of the above tiers:

```
CREATE TABLESPACE <tablespace_name> datafile <datafile_name> SIZE <allocated_size> SEGMENT SPACE management auto extent management local autoallocate;
```

For example:

```
CREATE TABLESPACE less_active_data datafile '/oracleE2POC/data01/s2poc/less_active_data01.dbf' SIZE 10m SEGMENT SPACE management auto extent management local autoallocate;
```

13.2.3 Create Partition Script

Partitioning script can be generated through partition script creation utility. For ILM qualified tables, the tables should always be partitioned based on range. Partition script can be generated based on Day, Month and Year. The following parameters need to be provided to the utility:

- Table Name
- ILM Column Name
- Schema Name (decided by DBA)
- Partition Interval (Default 1)
- Partition Type (DAY, MONTH and YEAR)
- Directory where partition script will be created (decided by DBA)

This utility can be run as follows:

1. Connect to OBP Database.
2. Run the following SQL statement:

```
DECLARE  
PI_TABLE_NAME VARCHAR2(200);
```

```
PARTITION_COLUMN_NAME VARCHAR2(200);
SRC_SCHEMA_NAME VARCHAR2(200);
PARTITION_INTERVAL NUMBER;
PARTITION_TYPE VARCHAR2(200);
DIRECTORY_NAME VARCHAR2(200);
DURATION NUMBER;
DURATION_TYPE VARCHAR2(200);
BEGIN
PI_TABLE_NAME := <ILM qualified table name>;
PARTITION_COLUMN_NAME := <ILM column name>;
SRC_SCHEMA_NAME := <Source schema name>;
PARTITION_INTERVAL := <Duration>;
PARTITION_TYPE :=< Partition type as DAY,MONTH or YEAR>;
DIRECTORY_NAME := <Location where partition script will be
created>;
DURATION := 0;
DURATION_TYPE := NULL;

AP_OPA_ILM_CREATE_PARTITION(
PI_TABLE_NAME => PI_TABLE_NAME,
PARTITION_COLUMN_NAME => PARTITION_COLUMN_NAME,
SRC_SCHEMA_NAME => SRC_SCHEMA_NAME,
PARTITION_INTERVAL => PARTITION_INTERVAL,
PARTITION_TYPE => PARTITION_TYPE,
DIRECTORY_NAME => DIRECTORY_NAME,
DURATION => DURATION,
DURATION_TYPE => DURATION_TYPE
);
--rollback;
END
```

Figure 13–1 Partition Script - SQL Statement

```
PL/SQL Block
DECLARE
PI_TABLE_NAME VARCHAR2(200);
PARTITION_COLUMN_NAME VARCHAR2(200);
SRC_SCHEMA_NAME VARCHAR2(200);
PARTITION_INTERVAL NUMBER;
PARTITION_TYPE VARCHAR2(200);
DIRECTORY_NAME VARCHAR2(200);
DURATION NUMBER;
DURATION_TYPE VARCHAR2(200);
BEGIN
PI_TABLE_NAME := 'FLX_DA_ACCT_EVENT_H_DEMO';
PARTITION_COLUMN_NAME := 'POSTING_DATE';
SRC_SCHEMA_NAME := 'OBPIUTT10_ILM';
PARTITION_INTERVAL := 1;
PARTITION_TYPE := 'MONTH';
DIRECTORY_NAME := '/scratch/app/ILM_PARTITION_DIR/';
DURATION := 0;
DURATION_TYPE := NULL;

AP_OPA_ILM_CREATE_PARTITION(
PI_TABLE_NAME => PI_TABLE_NAME,
PARTITION_COLUMN_NAME => PARTITION_COLUMN_NAME,
SRC_SCHEMA_NAME => SRC_SCHEMA_NAME,
PARTITION_INTERVAL => PARTITION_INTERVAL,
PARTITION_TYPE => PARTITION_TYPE,
DIRECTORY_NAME => DIRECTORY_NAME,
DURATION => DURATION,
DURATION_TYPE => DURATION_TYPE
);
--rollback;
END;
```

3. After execution, the utility table creation script appears as shown in the below figure.

Figure 13–2 Utility Table Creation Script

```

CREATE TABLE "OBPIUTT10_ILM"."FLX_DA_ACCT_EVENT_H_DEMO"
(
  "TRN_REFERENCE_CODE" VARCHAR2(16),
  "TRN_EVENT_SEQ_NUM" NUMBER,
  "TRN_EVENT_CODE" VARCHAR2(30),
  "TRN_BANK_CODE" VARCHAR2(10),
  "TRN_BRANCH_CODE" VARCHAR2(10),
  "TRN_DESC" VARCHAR2(750),
  "CHANNEL_CODE" VARCHAR2(20),
  "EVENT_OCCURED_DATE" TIMESTAMP (6),
  "POSTING_DATE" DATE,
  "PROCESS_DATE" DATE,
  "PRODUCT_CODE" VARCHAR2(30),
  "PARTY_CODE" VARCHAR2(40),
  "RELATED_ACCOUNT_CODE" VARCHAR2(40),
  "RELATED_ACCT_BRANCH_CODE" VARCHAR2(10),
  "RELATED_MODULE_TYP" VARCHAR2(2),
  "RELATED_ACCT_STATUS" VARCHAR2(20),
  "DOMAIN_CATEGORY" VARCHAR2(2),
  "ORIGINAL_TRN_REF_CODE" VARCHAR2(16),
  "ORIGINAL_EVENT_SEQ_NUM" NUMBER,
  "REVERSED_FLAG" VARCHAR2(1),
  "REVERSAL_PROC_FLAG" VARCHAR2(1),
  "DELETED_FLAG" VARCHAR2(1),
  "AUTHORIZED_FLAG" VARCHAR2(1),
  "ACCOUNTING_STAGE" VARCHAR2(10),
  "EVENT_TYP" VARCHAR2(10),
  "EVENT_STATUS_TYP" VARCHAR2(3),
  "BATCH_EVENT_FLAG" VARCHAR2(1),
  "EVENT_PROCESSED_DATE" DATE,
  "ERROR_CODE" VARCHAR2(20),
  "ERR_CODE_DESC" VARCHAR2(3000),
  "CREATED_BY" VARCHAR2(254),
  "AUTHORIZED_BY" VARCHAR2(254)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "OBPIUTT10_ILM" PARTITION BY RANGE (POSTING_DATE)
(INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))
(PARTITION p0 VALUES less than (TO_DATE('21-DEC-2014','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p1 VALUES less than (TO_DATE('21-JAN-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p2 VALUES less than (TO_DATE('21-FEB-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p3 VALUES less than (TO_DATE('21-MAR-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p4 VALUES less than (TO_DATE('21-APR-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p5 VALUES less than (TO_DATE('21-MAY-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p6 VALUES less than (TO_DATE('21-JUN-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p7 VALUES less than (TO_DATE('21-JUL-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p8 VALUES less than (TO_DATE('21-AUG-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p9 VALUES less than (TO_DATE('21-SEP-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p10 VALUES less than (TO_DATE('21-OCT-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p11 VALUES less than (TO_DATE('21-NOV-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
, PARTITION p12 VALUES less than (TO_DATE('21-DEC-2015','DD-MON-YYYY')) tablespace OBPIUTT10_ILM
);
INSERT INTO OBPIUTT10_ILM.FLX_DA_ACCT_EVENT_H_DEMO select * from OBPIUTT10_ILM.FLX_DA_ACCT_EVENT_H_DEMOT ;
DROP TABLE OBPIUTT10_ILM.FLX_DA_ACCT_EVENT_H_DEMOT ;
comment on table FLX_DA_ACCT_EVENT_H_DEMO is

```

13.2.4 Run Partition Script

The steps to run the partition script are as follows:

1. Download the newly created partition script from specified directory.
2. Verify created partition script before running.
3. Execute the script on OBP database as follows:
 - a. Connect to OBP Database.
 - b. Run partition creation script:

```
@ <Tablename>par.sql
```

For example:

```
@ /scratch/app/ILM_PARTITION_DIR/FLX_DA_ACCT_EVENT_H_DEMOparsql
```

13.2.5 Create and Register ADO Policies based on Lifecycle Pattern

Automatic Data Optimization (ADO) is used to create policies and automate actions based on those policies, for implementing the ILM strategy. The data is moved across storage tiers. The following script needs to be executed to create the ADO policies:

1. Connect to OBP Database.
2. Run ADO policy creation script:

```
@ <Tablename>ado.sql
```

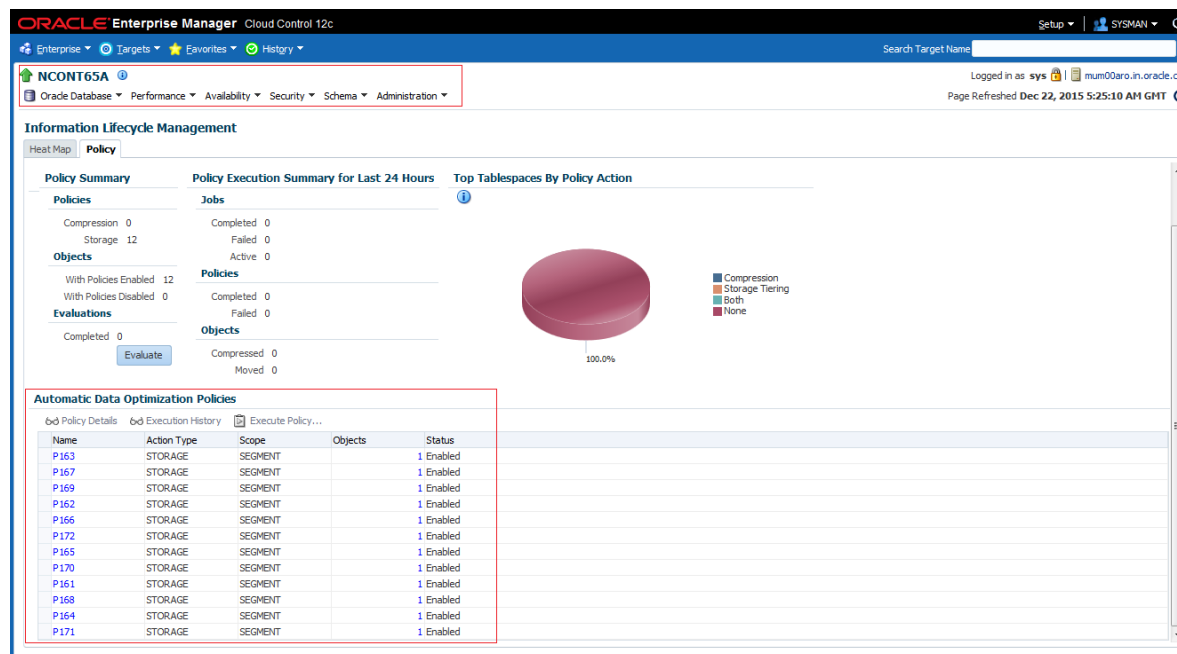
For example:

```
@ /scratch/app/ILM_ADO_DIR/FLX_DA_ACCT_EVENT_H_DEMOado.sql
```

13.2.6 Verify Registered ADO Policies

The created ADO policies can be verified through Oracle Enterprise Manager.

Figure 13–3 Verify ADO Policies



13.3 Policy Execution

ADO policies are required to be scheduled to execute automatically by configuring the database maintenance period. This can be determined during the implementation phase.

These ADO policies can be additionally executed manually with the following command:

```
declare
  v_executionid number;
begin
  dbms_ilm.execute_ilm (ilm_scope=>dbms_ilm.scope_schema,
    execution_mode=>dbms_ilm.ilm_execution_offline,
    task_id=>v_executionid);
end;
/
```

14 Transparent Data Encryption (TDE)

This chapter describes the configuration, installation, and policy setup of Transparent Data Encryption (TDE).

Transparent Data Encryption is a technology used to encrypt database files. This feature enables you to protect sensitive data in database columns stored in operating system files by encrypting it. Then, to prevent unauthorized decryption, it stores encryption keys in a security module external to the database.

14.1 Configuration

The following is the classification of information related to OBP. This information is used to drive TDE configuration.

Table 14–1 TDE Configuration

Classification	Details	Access and Distribution	Action
Public	This information is not sensitive, and there is no value with it remaining confidential to Bank.	No restrictions	No Encryption
Confidential Internal	It is important that this information remains confidential to Bank.	May be accessed by and distributed to all support person. Distribution to third parties must be authorized by the information owner and requires that an appropriate confidential disclosure agreement be in place.	No Encryption
Confidential Restricted	It is very important that this information remains confidential to Bank and that access within bank is restricted on a need-to-know basis.	Internal access/distribution must be on a business need-to-know basis. Not authorized for information unless the information is encrypted using Oracle-approved encryption.	Need to set encryption rule during TDE
Confidential Highly Restricted	It is essential that this information remains confidential to Bank and that access within bank is restricted on a need-to-know basis.	Internal access/distribution must be very limited and is on a stringent business need-to-know basis. Not authorized for information unless the information is encrypted using Oracle-approved encryption.	Need to set encryption rule during TDE

All tables in OBP are classified based on above classification and columns of those tables are marked based on sensitivity.

14.2 Installation

This section explains the installation process.

14.2.1 Prepare Scripts to Encrypt Sensitive Data

Database administrator needs to create alter script to encrypt sensitive data. The utility tool (obpencryption.sh) is used to create this alter script for TDE. To run the tool, the following prerequisites are required.

Prerequisites

- Create a folder "obpencryption" where user wants to run the tool.
- Upload Sensitive_Data_List.xlsx, obp-encryption-script-gen.jar, obpencryption.sh, DB_RESOURCEBUNDLE.properties. These files are available in maskingencryption.zip. The maskingencryption.zip is part of host.zip available in installer.
- Update database details in DB_RESOURCEBUNDLE.properties file before running the script.
- Update value "encryptLocation" variable with obp encryption path in obpencryption.sh at line 6.

For example: `encryptLocation="/scratch/app/product/obpencryption"`

Run Encryption Tool

- Create update scripts for all the tables containing sensitive data. Run obpencryption.sh with TDE and ENCRYPT.

For example: `/obpencryption.sh TDE ENCRYPT`

14.2.2 Create TDE Keystore

Perform these steps to create keystore which is required for encryption and decryption. Perform the following steps.

- Create keystore location with `mkdir -p <location>`.

For example: `mkdir -p /scratch/app/admin/TDE/encryption_keystore/`

- Log in to database with `sysdba`.

For example: `sqlplus / as sysdba`

- Run the following sql instruction:

- ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '{Keystore loaction}' IDENTIFIED BY {Password}

For example: `SQL>ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/scratch/app/admin/TDE/encryption_keystore/' IDENTIFIED BY myPassword`

- ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY welcome1 CONTAINER=ALL;

For example: `SQL>ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY welcome1 CONTAINER=ALL;`

- ADMINISTER KEY MANAGEMENT CREATE KEY using tag 'KEY5' IDENTIFIED BY welcome1 WITH BACKUP CONTAINER =all;

For example: `SQL>ADMINISTER KEY MANAGEMENT CREATE KEY using tag 'KEY5'
IDENTIFIED BY welcome1 WITH BACKUP CONTAINER =all;`

- **ADMINISTER KEY MANAGEMENT SET KEY using tag 'KEY5' IDENTIFIED BY welcome1 WITH BACKUP CONTAINER=ALL**

For example: `SQL>ADMINISTER KEY MANAGEMENT SET KEY using tag 'KEY5'
IDENTIFIED BY welcome1 WITH BACKUP CONTAINER=ALL;`

- Check the encryption keys generated.

For example: `SQL> SELECT con_id, key_id FROM v$encryption_keys;`

- Check the wallet status.

For example: `SQL> SELECT * FROM v$encryption_wallet;`

14.2.3 Edit sqlnet.ora file

Perform this step to enter the TDE wallet location.

- Take a backup of sqlnet.ora file before update for TDE.
- Add entries of sqlnet.ora file as follows:

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE =(METHOD = FILE)(METHOD_DATA =
(DIRECTORY = {Keystore location})
```

For example:`ENCRYPTION_WALLET_LOCATION =
(SOURCE = (METHOD = FILE) (METHOD_DATA =
(DIRECTORY = /scratch/app/admin/TDE/encryption_keystore/)`

14.2.4 Run Created Alter Script

- Get TDE_Encryption.sql script from obpencryption/generatedScript/tde.
- Log in to database.
- Run TDE_Encryption.sql.

15 Masking Customer Private Data

This chapter describes the configuration, installation, and policy setup to mask customer private data categories as sensitive or Personally Identifiable Information (PII).

15.1 Configuration

The following is the classification of information related to OBP. This information is used to drive TDE configuration.

Table 15–1 TDE Configuration

Classification	Details	Access and Distribution	Action
Public	This information is not sensitive, and there is no value with it remaining confidential to Bank.	No restrictions	No Encryption
Confidential Internal	It is important that this information remains confidential to Bank.	May be accessed by and distributed to all support persons. Distribution to third parties must be authorized by the information owner and requires that an appropriate confidential disclosure agreement is in place.	No Encryption
Confidential Restricted	It is very important that this information remains confidential to Bank and that access within bank is restricted on a need-to-know basis.	Internal access/distribution must be on a business need-to-know basis. Not authorized for information unless the information is encrypted using Oracle-approved encryption.	Need to set encryption rule during masking Tables containing this type of data will be accessed through view for RO user. Synonym needs to be created for the tables and views containing this type of data for RO and ERO user.
Confidential Highly Restricted	It is essential that this information remain confidential to Bank and that access within bank is restricted on a need-to-know basis.	Internal access/distribution must be very limited and is on a stringent business need-to-know basis. Not authorized for information unless the information is encrypted using Oracle-approved encryption.	Need to set encryption rule during masking. Tables containing this type of data will be accessed through view for RO user. Synonym needs to be created for the tables and views containing this type of data for RO and ERO user.

All tables in OBP are classified based on above classification and columns of these tables are marked based on sensitivity.

15.2 Installation

This section explains the installation process.

15.2.1 Prepare Scripts to Encrypt Sensitive Data

Database administrator needs to create the following script for masking sensitive data.

- View creation script of the tables containing sensitive data and mask them for RO (Read only) user.
- Synonym creation script of created view of the containing sensitive data for RO (Read only) user.
- Synonym creation script of tables containing sensitive data for ERO (E Read only) user.

The utility tool (obpencryption.sh) is used to create above script. To run the tool, the following prerequisites are required.

Prerequisites

- Create a folder "obpencryption" where user wants to run the tool.
- Upload Sensitive_Data_List.xlsx, obp-encryption-script-gen.jar, obpencryption.sh, DB_RESOURCEBUNDLE.properties. These files are available in maskingencryption.zip. The maskingencryption.zip is part of host.zip available in installer.
- Update database details in DB_RESOURCEBUNDLE.properties file before running the script.
- Update value "encryptLocation" variable with obp encryption path in obpencryption.sh at line 6.

For example: `encryptLocation="/scratch/app/product/obpencryption"`

Run Encryption Tool for View Creation script and mask data

- Create view creation scripts for all the tables containing sensitive data after mask. Run obpencryption.sh with MASK and VIEWCREATE as parameter.

For example: `/obpencryption.sh MASK VIEWCREATE`

Run Encryption Tool for Synonym Creation script for RO user

- Create synonym creation scripts for all the created containing sensitive data. Run obpencryption.sh with MASK and SYNONYMRO as parameter.

For example: `/obpencryption.sh MASK SYNONYMRO`

Run Encryption Tool for Synonym Creation script for ERO user

- Create synonym creation scripts for all the tables containing sensitive data. Run obpencryption.sh with MASK and SYNONYMEERO as parameter.

For example: `/obpencryption.sh MASK SYNONYMEERO`

15.2.2 Create Schema for RO and ERO User

To create schema for RO and ERO user, execute the following steps.

- Create Read-Only (RO) and E Read-Only (ERO) user for accessing masked data from view and table.
- Grant for proper access.

15.2.3 Execute Created Scripts through Encryption Tool

Run all created scripts through the encryption tool for the following task.

- Mask sensitive data for RO user.
- Create view for tables contain sensitive data.
- Create synonym to access the view.
- Create synonym to access the table for ERO user.

To do the above tasks, perform the following steps.

- Get all view creation scripts from obpencryption /generatedScript/masking/viewforRO location and run after logging in to database.
- Get synonym creation script (MaskingSynonymForRO.sql) for RO user from obpencryption/generatedScript/masking/synonymForRO and run after logging in to database.
- Get synonym creation script (MaskingSynonymForERO.sql) for ERO user from obpencryption/generatedScript/masking/ synonymForERO and run after logging in to database.

16 Configure ODI for Inbound Document Upload

This chapter provides the steps to configure ODI for Inbound Document Upload.

For document upload ODI execution, complete the following configurations:

1. Configuring the Input directory:

a. For example, if input directory is `/scratch/odi/InboundDocument/Upload/lendingZone/`

b. Update the configuration in the table using the following SQL:

```
update FLX_FW_ODI_SUB_INTERFACE_TYPE set IN_FILE_PATH=
/scratch/odi/InboundDocument/Upload/lendingZone/ ' where SUB_INTERFACE_ID='9551';
```

c. Note that the `SUB_INTERFACE_ID='9551'` should not be changed.

2. Configure the Schema directory:

a. Provide the directory where all the schemas are present.

Framework configuration:

```
update FLX_FW_ODI_SUB_INTERFACE_TYPE set SCHEMA_FILE_PATH =
'/scratch/odi/InboundDocument/Upload/schema/' where SUB_INTERFACE_ID ='7002';
```

b. Copy all the schema for ODI mediapack zip from directory schema to the new directory which is configured for framework.

c. Provide schema file for Document Upload.

```
update FLX_FW_ODI_SUB_INTERFACE_TYPE set schema_file_path='D:\work\odi\inDocUpload\InboundDocument\Upload\schemas\scan_images_request.0.1.XSD' where SUB_INTERFACE_ID='9551';
```

3. Configure Temporary directory:

```
update FLX_FW_ODI_SUB_INTERFACE_TYPE set temp_file_path=
'/scratch/odi/InboundDocument/Upload /lendingZonetmp/' where SUB_INTERFACE_ID='9551';
```

4. Configure Archive directory:

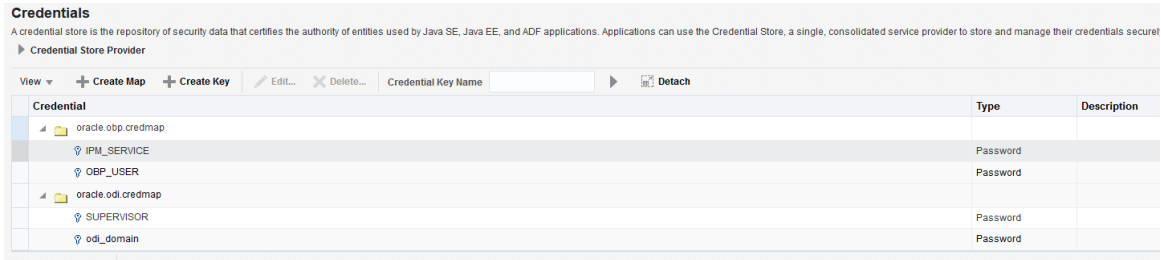
```
update FLX_FW_ODI_SUB_INTERFACE_TYPE set archive_file_path=
/scratch/odi/InboundDocument/Upload/archieve/' where SUB_INTERFACE_ID='9551';
```

5. Create users in connector: Create two credential maps:

- oracle.obp.credmap: This has two keys.
 - IPM_SERVICE: It has the username and password of IPM. It is used to upload the documents to IPM.
 - OBP_USER: It is required to make web service call to OBP.

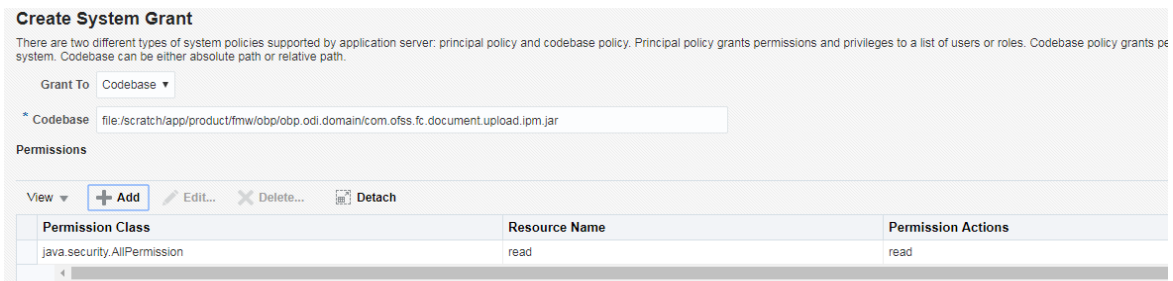
- oracle.odi.credmap
 - SUPERVISOR: It has supervisor username and password.
 - odi_domain: It has domain username and password.

Figure 16–1 Credentials



6. Provide permission to the java project for fetching the user credentials. Provide read permission to Java binary com.ofss.fc.document.upload.ipm.jar from EM.

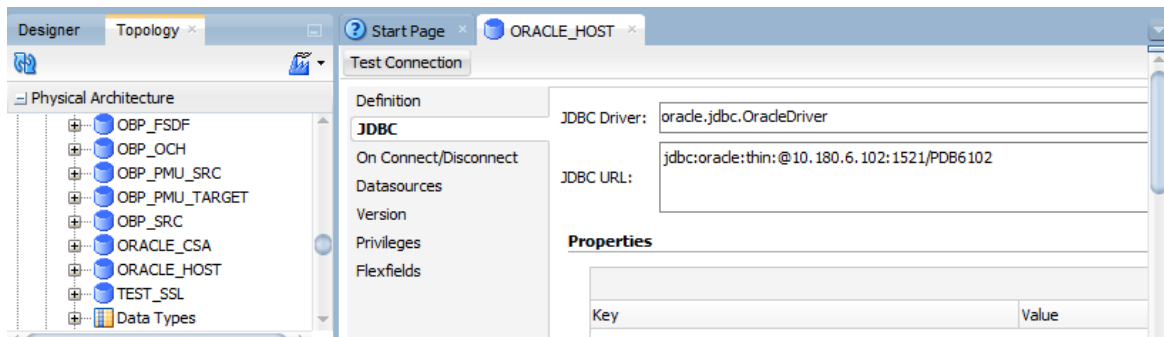
Figure 16–2 Example of permissions



7. Configuring IPM URL:
 - a. update FLX_FW_ODI_SUB_INTERFACE_TYPE set GEFU_IN_FILE_PATH='http://\${IPM-HOSTNAME}:\${IPM-PORT}/imaging/ws' where SUB_INTERFACE_ID='9551';
 - b. Replace \${IPM-HOSTNAME} with IPM Hostname or IP address.
 - c. Replace \${IPM-PORT} with IPM Server port number.
8. Configure config/properties/OutboundWebserviceConfig.properties to provide OBP Host web service configuration.
 - a. Replace \${OBP-HOST-IP} with OBP Host IP address or hostname.
 - b. Replace \${OBP-HOST-PORT} with OBP Host managed server port.

-
9. Provide FJ Connection Details in ODI Topology data server ORACLE_HOST.

Figure 16–3 Connection details



17 Additional Recommendations

This chapter provides specific recommendations to be considered for implementation:

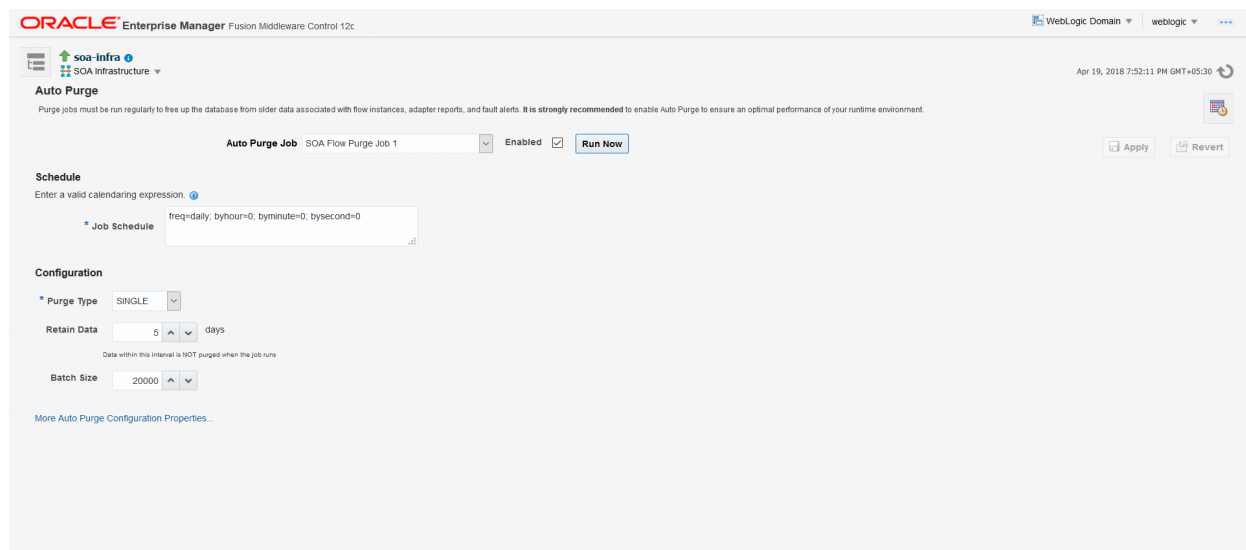
17.1 SOA Related

This section mentions certain recommendations for setting up the properties on SOA server.

17.1.1 Enable Auto Purge Job

- Oracle SOA Suite team strongly recommends periodic purging of composite instances. Purge instances as soon as they are available for purge.
- SOA suite 12c comes up with default purge job enabled with retention period of 7 days.
- It is recommended to keep this default job enabled in the production.

Figure 17–1 Auto Purge

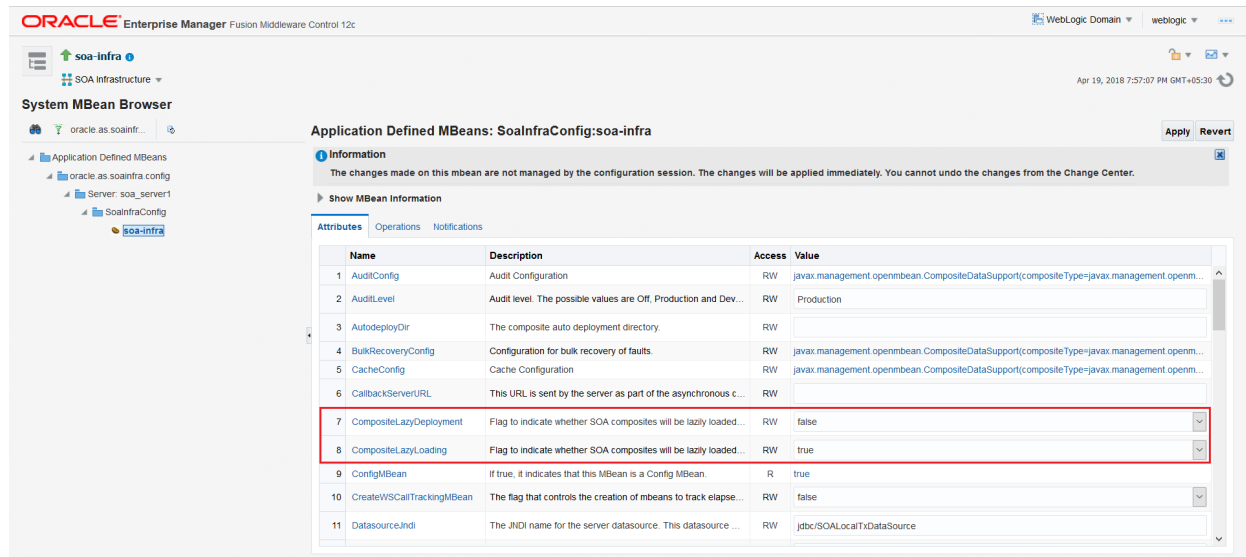


17.1.2 Enable Lazy Loading

- 12c supports lazy loading of composites on deployment as well as server startup.
- **CompositeLazyDeployment:** Loads the composites lazily on deployment.
 - More useful for non-production environments where there is frequent deployment of all the composites.
- **CompositeLazyLoading:** Loads the composites lazily on server start up.
- It is recommended to keep the default settings unchanged, that is CompositeLazyDeployment as false and CompositeLazyLoading as true in production environment.

- This will reduce the SOA server startup time.
- To confirm the settings, in SOA EM console, go to soa-infra > SOA Administration > Common Properties. Click More SOA Infra Advanced Configuration Properties link.

Figure 17–2 Lazy Loading Settings



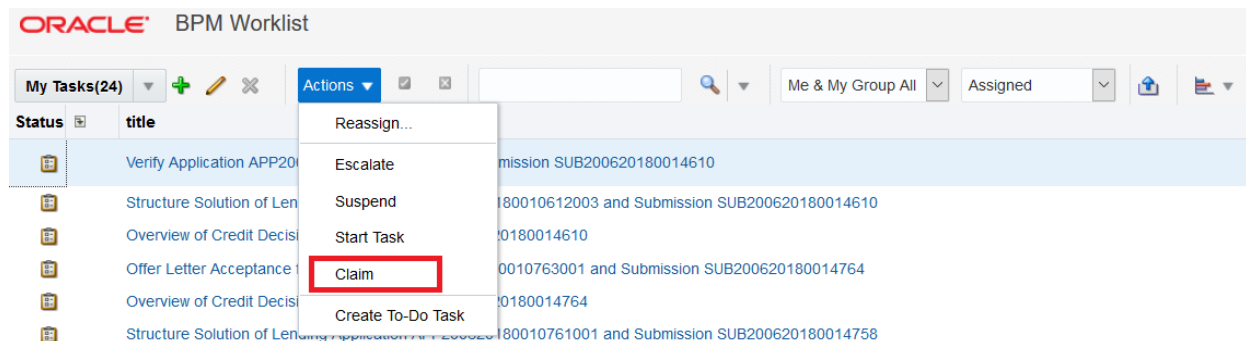
17.2 BPM Worklist Related

This section mentions certain recommendations on usage of BPM worklist.

17.2.1 Disable Claim Action from Task Details Page

- Always claim the task using worklist action menu.
- Claim option is disabled by default inside task details page.
- It can be enabled (configurable), if needed (but not recommended).
- Disabling this option helps to avoid loading of task details page twice, that is before claiming the task and after claiming the task.

Figure 17–3 Claim Action



17.2.2 Always Open Human Task Details in External Window

- BPM worklist supports two options to display the task details.
 - **Same Window:** Human task details are opened in the same browser window just below the worklist grid.
 - **External Window:** Human task details are opened in the new browser window (as a child popup).
- It is strongly recommended to make use of **External Window** option.
 - Human task details can be seen in full-screen mode.
 - Avoids unnecessary loading the task details page if user is browsing through the list of human tasks in worklist grid.
- This option can be enabled from worklist administration page.

Figure 17–4 Enable External Window option

