

**Oracle Business Process Management**

**Tutorial Lab Project (INN696-1, Semester 1/2014)**

**Loan Assessment Scenario**

Student: Hoang Huy Nguyen, Student No.: 8651523

Supervisor: Marcello La Rosa

**Contents**

[1 Introduction 3](#_Toc391555087)

[2 Overview of Oracle BPM 3](#_Toc391555088)

[3 Tutorial Prerequisites 3](#_Toc391555089)

[4 The Tutorial Project: Loan Assessment 3](#_Toc391555090)

[5 Oracle BPM Studio environment 3](#_Toc391555091)

[6 Process Modelling 11](#_Toc391555092)

[7 Initial Configurations 14](#_Toc391555093)

[8 Process Organization Model 16](#_Toc391555094)

[9 Implementation of the Business Process 18](#_Toc391555095)

[9.1 Creating Data Objects 18](#_Toc391555096)

[9.2 Implementing Interactive Tasks 20](#_Toc391555097)

[9.3 Implementing Service Tasks 24](#_Toc391555098)

[9.4 Implementing Business Rules 24](#_Toc391555099)

[10 Deploying process to SOA and BPM servers 24](#_Toc391555100)

[11 Common Errors 24](#_Toc391555101)

# Introduction

This project is to deploy Oracle BPM environment on a lab computer and demonstrate process implementation with a Loan Assessment process based on *Fundamentals of Business Process Management* book by Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.

|  |
| --- |
| Here is a very brief summary of the process. The process is initiated when a loan applicant submits an application form. A loan officer will receive the application form and check for its completeness, either manually or automatically with a system automatic check. If the application is not complete, it is sent back to the applicant for supplement. In case the application is not completed by the applicant within 5 days, the system will terminate the process. Once it is complete, it will be forwarded to a financial officer and a property appraiser for assessment. The financial officer checks credit history of the applicant while the property appraiser evaluates the value of the property. The financial officer gives the application a credit grade (A, AA, B, BB, etc.) which will be translated into a rating mark by the system. Once both financial and property assessment are done, the application is transferred to a loan officer who will assess its overall eligibility. If it is not eligible, the process sends email notification to the applicant and stops there. If it is marked as eligible, the loan officer will then prepare an application pack which includes a repayment agreement with detailed terms/conditions and send to the applicant. In the next step, if the application includes a request for home insurance, it will be forwarded to an insurance sales representative (ISR) who will provide a quote for the home insurance. Once the loan officer receives the repayment agreement returned from the applicant (via email, hard-copy) and the home insurance quote from the ISR (if applicable), the loan officer will verify the repayment agreement and forward all information to another loan officer for the final decision. In case the repayment agreement verification is pending for more than 14 days, the system will terminate the process. In the final step, the loan officer makes a decision to approve or reject the application. In either cases, an email notification will be sent to the applicant for their information and the process stops. |

In this tutorial, we will implement the above process with Oracle SOA/BPM using its key components such as Oracle BPMN, BPEL, Business Rules, Human Workflow, and Oracle ADF. System installation for Oracle SOA/BPM and Oracle JDeveloper must be in place to do this tutorial.

It will be easier for readers to conduct this tutorial if they are equipped with basic knowledge and expertise of BPMN, BPEL, Java programming, XML, SOA, Web services and Java servlet/JSF technologies.

The tutorial is structured with the following sections.

|  |  |
| --- | --- |
| Background of Oracle BPM | Explains underlying Oracle BPM concepts and technologies used in this tutorial |
| Install Oracle SOA/BPM | Outlines main installation steps along with sharing experience during the installation process. Readers should still refer to Oracle installation document for technical instructions |
| Navigate Oracle JDeveloper environment | Explain essential UI editors used in Oracle JDeveloper environment to help readers familiarize themselves more quickly with this complex environment |
| Create a new BPM application project | Step by step guide on how to create a BPM application project from the ground up |
| Model the Process | Step by step guide on how to model the process scenario in BPMN notation with Oracle BPM studio |
| Implement the Process | Step by step instructions on how to implement the process with Human task, rules, server script, timer event |
| Deploy the Process | Deploy the process to SOA/BPM servers |
| Run the Process | Run the process as a process user and track the process run trace as process administrator |

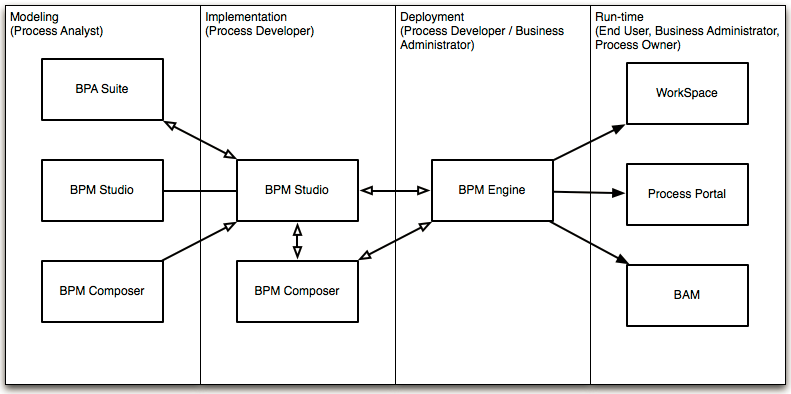
# Background of Oracle BPM

This section is to equip learners with key components of an Oracle BPM process-oriented application and some architecture concepts. This knowledge is required for understanding the tutorial practice in section X which has more focus on procedural steps.

Oracle offers a comprehensive BPM environments based on service component architecture. We can look at this environment from different views: business analyst, BPM developer, system administrator or process user. In this tutorial, we will engage in all of these roles; therefore this section would provide a quite complete overview picture.

**Oracle BPM Application Development Lifecycle (http://docs.oracle.com/cd/E23943\_01/user.1111/e15175/bpmug\_intro\_bpm\_suite.htm)**

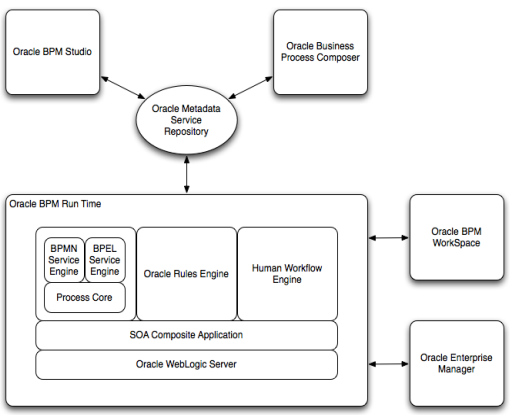
The stages of the development life cycle of an Oracle BPM application are outlined below.



**Oracle BPM Architecture**

**(**[**http://docs.oracle.com/cd/E23943\_01/user.1111/e15175/bpmug\_intro\_bpm\_suite.htm**](http://docs.oracle.com/cd/E23943_01/user.1111/e15175/bpmug_intro_bpm_suite.htm)**)**

The core of Oracle BPM product is Oracle BPM Run-Time as shown in Figure XX.



*Oracle BPM Runtime*

Oracle BPM Runtimeor“service infrastructure” provides the internal message transport infrastructure for connecting components and enabling data flow. The service infrastructure is responsible for routing messages along the wire connections between services, service components, and references.

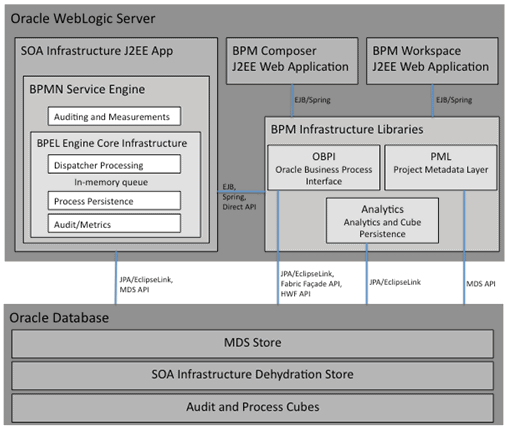
*BPEL Service Engine*

For process orchestration of synchronous and asynchronous processes

*BPMN Service Engine*

For creating and modeling business processes using Business Process Management Notation and Modeling (BPMN). The BPEL, BPMN and Process Core (provided shared services for both BPMN and BPEL engines) are commonly called as BPM engine.

Since we are going to work extensively with BPMN Engine, let’s take a deeper look in Figure XX.



BPMN Service Engine is actually an extension of the existing BPEL Service Engine and as such it leverages the core infrastructure of the BPEL. The BPMN Service Engine leverages JPA/EclipseLink to store/recover the state of a process instance in the SOA Infrastructure dehydration store maintained by a database and to persist audit records that are created in the course of running a process. MDS APIs are used to retrieve metadata Information about the BPMN Process Model and other BPM project artifacts like the Business Catalog.

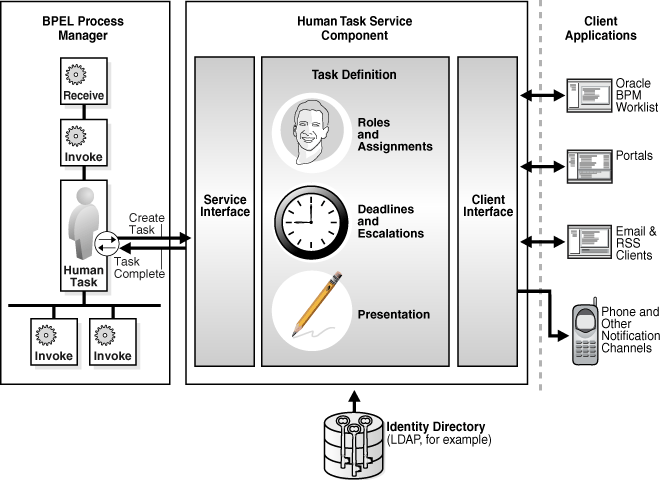
*Business Rules Engine*

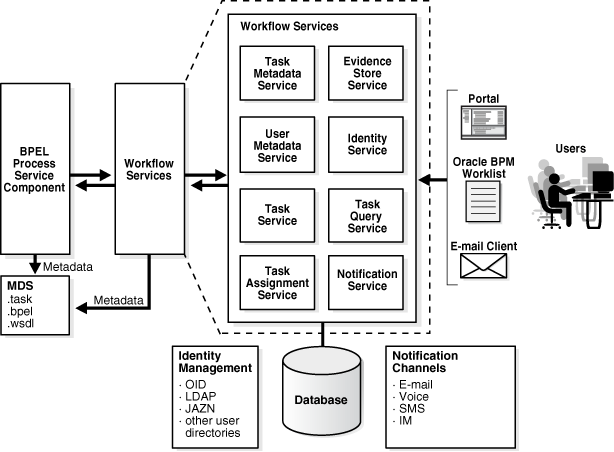
For making a decision or for processing based on business rules.

*Human Workflow Engine*

For modeling a human task (for example, manual order approval) that describes the tasks for users or groups to perform as part of an end-to-end business process flow.

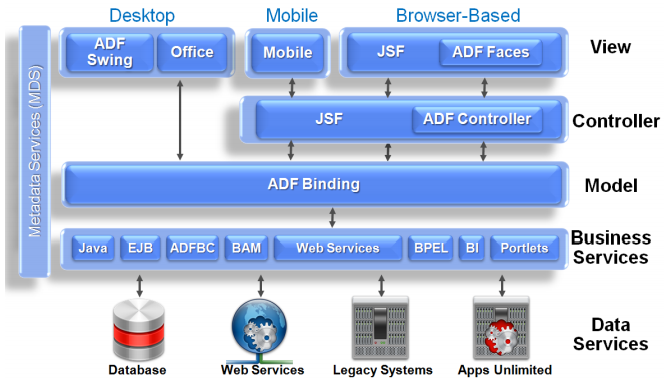
This engine is also used extensively in this tutorial. The details of its service are shown below.





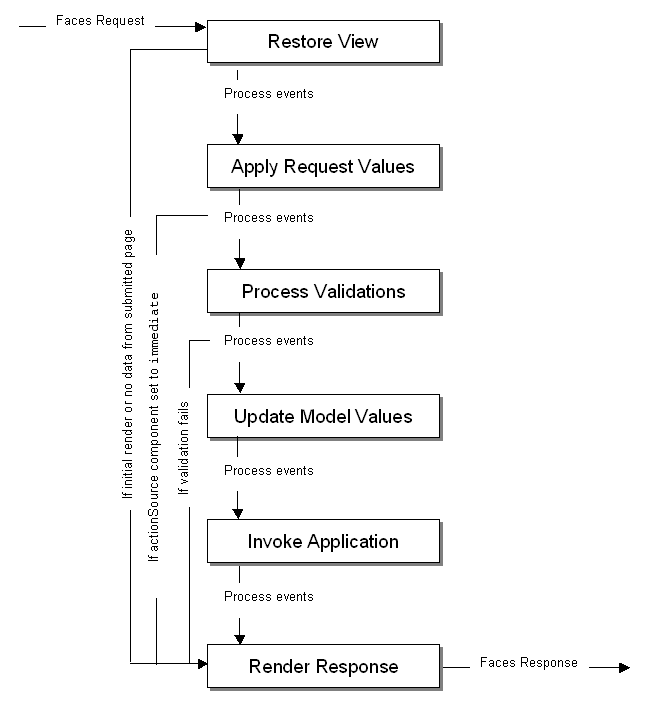
*User Interface and Data Control*

Another important aspect of BPM application is the interaction with users which is implemented by Oracle ADF. Oracle ADF runs on top of Java Server Faces (JSF) which abstracts HTML and Java Servlet technologies with web components.



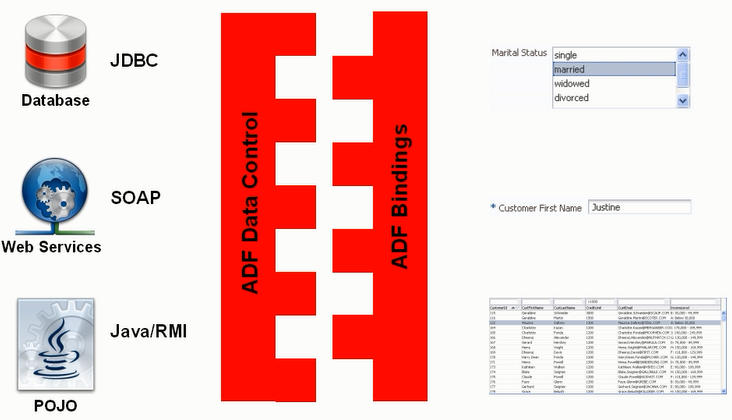
In this tutorial, we are going to implement a number of web pages in Oracle ADF. However, we will introduce the most common functions of Oracle ADF to facilitate our practice in the tutorial. Many other features of this complex framework will not be covered here. The tutorial user should refer to other materials available for Oracle ADF if they want to gain a good understanding.

Figure XX depicts main phases of ADF Face Request. The key point to us here is the “Invoke Application” phase where we can our custom codes will be processed.



Oracle ADF added the following new features to JSF:

* A wide range of new Ajax-based UI faces (components)
* Data bindings
* Data controls



The prominent contribution of Oracle ADF is the ADF Model layer which includes Data Control and Bindings as shown in Figure XX. Other mechanism is primarily based on JSF infrastructure.

The main idea of data control and data bindings is they simplify the process of associating UI elements with various data source in a declarative manner.

For BPM technology, Oracle provides a special data control called BPM data control which interacts with both BPMN and Human Workflow Engine. It hides the complexity in dealing with these two engines.

We will use data bindings and data control extensively in the tutorial.

**Other technologies**

In addition, Oracle provides other BPM related environments but they out of scope of this tutorial:

* Business Activity Monitoring
* Business Process Composer
* Oracle Service Bus
* Oracle WebCenter.

**What is Oracle BPM Application?**

Now, we’ll narrow down to some understanding of Oracle BPM application or process oriented application in terms of Oracle technologies, specifically the BPMN engine.

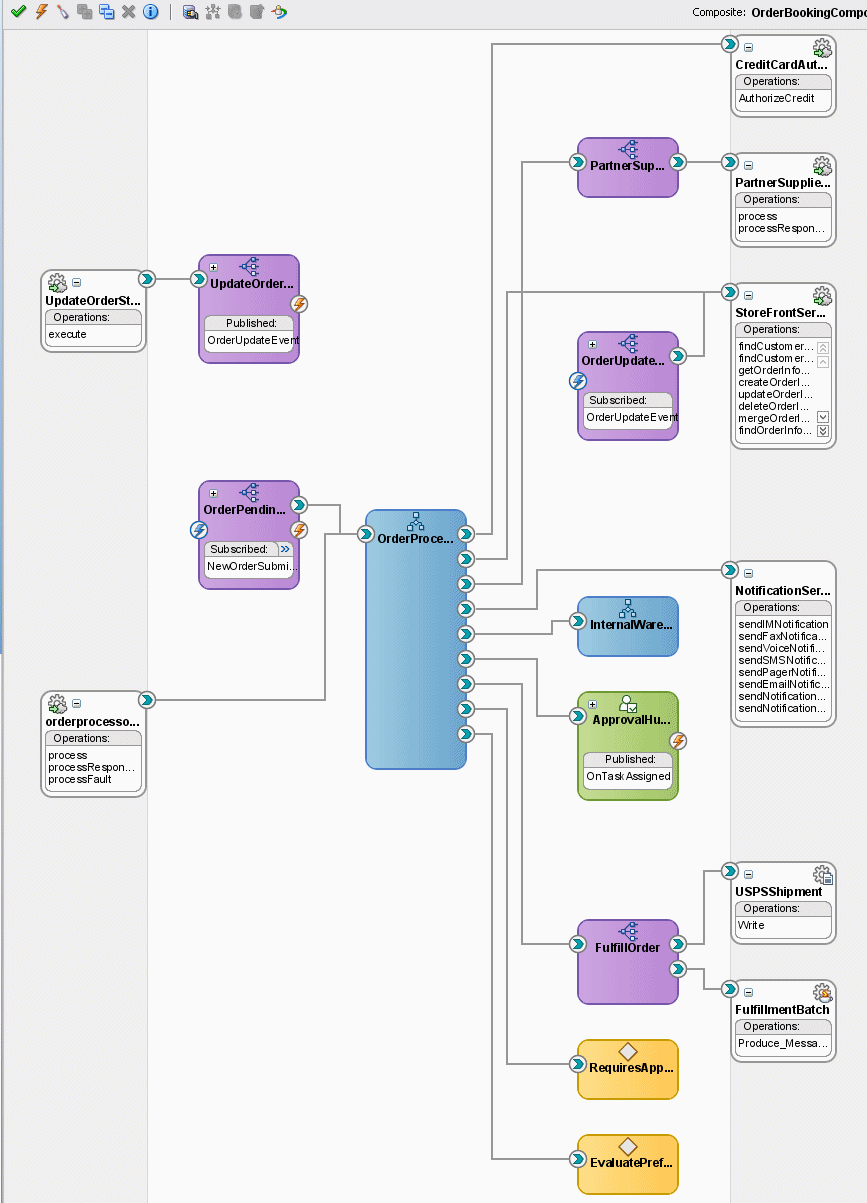
Two technologies were implemented long before BPMN engine is SOA infrastructure and BPEL. A BPM project is the basic unit of deployment to the BPM runtime. A BPM project is comprised of service components (such as BPMN Process, Business Rules, Human Workflow) and references organized as a SOA Composite, organizational data (such as Roles and Business Calendars), business indicator metadata and dashboard data. Components are targeted to service engines during deployment while services and references are enabled using the binding components. The metadata for organizational data, business indicators and dashboards is persisted and evaluated by appropriate components at runtime. At runtime, messages are received by the binding component or the BPM Workspace and are then routed to the appropriate service engine(s) by the Service Infrastructure.

It should be noted that every BPM application is actually a SOA Composite application in Oracle BPM implementation.

SOA composite applications such as those shown in the Deployed Composites page in [Figure 1-1](http://docs.oracle.com/cd/E28280_01/admin.1111/e10226/soasuite_intro.htm#CEGJGCIF) consist of service components. Note that each component is nothing but web applications created by Oracle using JSP,Servlet, EJB and JMS etc. These service components are installed as JEE application.(http://docs.oracle.com/cd/E28280\_01/admin.1111/e10226/soasuite\_intro.htm#CEGCCAEB).

These components are assembled into a single SOA composite application. Service components are the basic building blocks of SOA composite applications. Service components implement a part of the overall business logic of the SOA composite application.

Figure XX shows an example of a SOA composite application with various components: BPMN process, BPEL process, Human Tasks, Decision service (business rules) and Bindings.



An Oracle SOA composite application can consist of a variety of service components, binding components, and services.

* BPEL processes
* BPMN processes (if Oracle BPM Suite is installed)
* Human workflows
* Oracle Mediator
* Decision services (Oracle Business Rules)
* Spring
* JCA adapters
* HTTP binding
* EJB service
* Direct binding service
* Oracle Application Development Framework (ADF) Business Component service
* Oracle BAM
* Oracle B2B
* Oracle Healthcare
* Business events
* Oracle User Messaging Service

Note that Oracle SOA Composite Application complies with Service Component Architecture standard. Therefore, those interested can read more details in the SCA specification.

From a process user perspective, the application is an automation of a process model. However, looking insight of implementation, an Oracle BPM application is in nature a SOA Composite application with added components such as: BPMN Engine, BPM Workspace, BPM Process Composer. (<http://docs.oracle.com/cd/E25054_01/doc.1111/e15176/soa_composite_bpmpd.htm>)

<BPMN process picture>



Implementing a BPM process in Oracle BPM environment is actually creating an SOA composite application with BPMN process as a main component.

**How an Oracle BPM application executed at run-time**

When we run a BPM project, the SOA engine creates a SOA composite instance which is identified by a unique instance ID. The SOA composite instance contains an instance of each of the components defined in the SOA composite. For example, if our SOA composite defines a BPMN process and a human Task, then the SOA composite instance contains a BPMN process instance and a human task instance.

When the Oracle SOA Service Infrastructure application starts, it initializes all the service engines (BPMN, BPEL, Human Task, etc) and loads the composites from the MDS repository. If the composite contains any BPMN processes, it targets those individual components to the BPMN service engine, similarly for BPEL, Human Task and others. Once the process is loaded and its BPM specific metadata persisted in the database, the system is available to receive requests.

(<http://docs.oracle.com/cd/E14571_01/core.1111/e10106/ha_soa.htm>)

A detailed startup and shutdown lifecycle of BPMN process is as follows (similarly for other service engines)

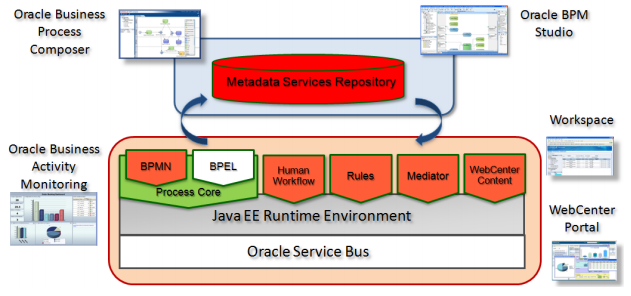
1. Start BPM Server.
2. Start BPMN service engine.
3. BPM project composites are loaded from the MDS repository by the SOA Service Infrastructure.
4. BPMN components are dispatched to the BPMN service engine to be loaded.
5. The BPM project metadata, such as organization data and audit/measurement metadata, is persisted in the infrastructure database.
6. Composite binding components are activated.
7. The BPMN engine services requests.
8. The shutdown signal is received by the SOA Service Infrastructure.
9. The SOA Service Infrastructure starts unloading composites.
10. Composite binding components are disabled.
11. BPMN components are dispatched to the BPMN engine to be unloaded.
12. The BPMN service engine shuts down.

Detailed service lifecycle at step 7 is elaborated as follows:

1. There are two ways to initiate a BPMN process instance, either using Initiate Task via BPM Workspace or from a Web Service client. In the former, a user would log into BPM Workspace and click on a visible process link that they are granted to start a process instance.
2. The BPMN Service Engine starts executing the process. The process activity may initiate different types of service components included in the process execution: BPEL process, web services, human task, etc.
3. If it is a BPEL process, the process will transfer control to BPEL process engine.
4. If it is a User Task, the BPMN Service Engine creates a user task and transfer control to Human Workflow Engine. After the human workflow is complete, control is passed back the BPMN engine. Any required data objects are passed back to the user task. Notably human tasks are independent from BPMN processes. Therefore, if we terminate a BPMN process while it runs a user task, the associated human tasks keeps running independently. Normally human task will interact with human users via web pages, mobile devices or desktop application. Oracle ADF is used to implement this interaction. ADF is an extension of Java Server Faces which in turn runs on java servlet technology. At this step, a HTTP request for ADF Face resource will be sent to web server which initiates ADF request processing lifecycle (refer to xxx).
5. If it is a server script, it will be executed directly in BPMN engine
6. If it is a web service or adapter, BPMN engine will call to SOA Infrastructure to execute external web service.
7. If it is a business rule, BPMN engine will transfer control to Business Rules Engine.
8. If it is an email task, BPMN engine will transfer it to Human Workflow Engine.

At step 4a, the integration between Human Task Engine and the Oracle ADF server is through ADF Data Control.

**Tooling view**



*Oracle BPM Studio*

This component is implemented in Oracle JDeveloper and used as integrated environment for modelling business processes (business analyst) as well as implementing the process flow (BPM developer).

*Oracle Business Process Composer*

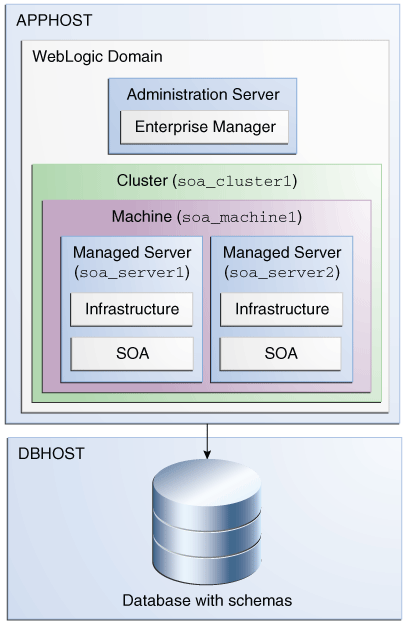
This is a lightweight tool which allows business analysts or power users to customize the process within their permissions without having to redevelop or redeploy the process.

*Oracle BPM Workspace*

*Oracle Enterprise Manager*

**Deployment Architecture**

The deployment of Oracle BPM on lab environment follows the topology shown in Figure X which covers the key products including Oracle BPM, BAM, MDS and Oracle Service Bus.



BPM Consoles

* Enterprise Manager
* BPM Workspace
* BPM Composer
* Weblogic Server Console

BPM Tooling Architecture

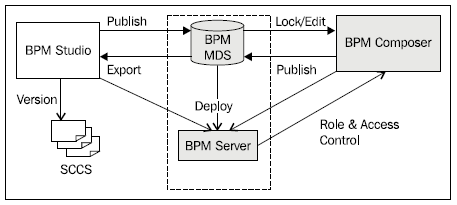


Figure 1: Oracle BPM Tooling Architecture

# Installation of Oracle BPM

This section outlines main installation steps for Oracle BPM/SOA version 11.1.1.7.0. It won’t contain all step-by-step instructions but readers should follow Oracle online installation document at the following link: <http://www.oracle.com/technetwork/middleware/soasuite/overview/quickstartguidesoasuite11gr1ps4-459545.pdf>

It is noted that Oracle installation manual is well-explained. This section thus only highlights critical steps and some important preparations based on practical experience to ensure a smooth process with less frustration.

Note that there may be multiple versions of Oracle SOA/BPM Suite available for download, but this paper is for Oracle SOA 11g and the version number is 11.1.1.7.

**Hardware configuration**

Oracle BPM/SOA is a large-scale installation which consume huge capacity of hard disk, RAM and CPU. My personal experience showed that

**Software requirements**

We have installed Oracle BPM successfully on the systems

* Windows 7 Professional or Windows 2008 Server
* Oracle Database 11g Release 2
* Internet Explorer 8 and 9
* Java Development Kit 7 64-bit server or IBM JRockit 64-bit server

There may be some confusion in terms of Oracle BPM Suite and Oracle SOA Suite packages for download. Oracle BPM Suite is actually Oracle SOA Suite with added BPM components. Oracle does not publish on their web site a separate product for BPM Suite 11g, but they provide SOA Suite 11g package and some BPM-related components to run on the SOA platform. Therefore, the installation process includes SOA Suite installation as a major set up work and some BPM components.

**Access privileges for installation**

The logged in user for Oracle SOA Suite installation should be the default Administrator of Windows, otherwise there may be numerous error messages relating to lack of system rights. This is because the installation script requires some critical system privileges in it chain of commands. Although you can login as a user with granted Administrator rights (belongs to Administrators group) and/or trigger the installation script with “Run as administrator” option, the chain of commands called by the script may lose this access identity during its execution and encounter privilege lacking errors during the process.

Again, it is vital to run Oracle Weblogic and Oracle SOA Suite installation script by logging in Windows as the default Administrator user.

It does not mean we have to always log in as Administrator user to be able to start Oracle BPM/SOA server after installation. We can do it with a different user and “Run as administrator” option.

**Installation Parameters**

It is a good practice to keep a note of all parameters and options set throughout the installation process. You will definitely need some of them later on or get frustrated because of not having them in hands. For example, they should include: installation folder for every component, port number, version number, domain name, server name in use, chosen username and password, service name, system ID, database name, and many others.

Now, once we are aware of the pre-installation notices, the following sections will explain key installation steps so that you can confidently follow Oracle installation manual. You can also relate the steps here with the architecture background in section 2.

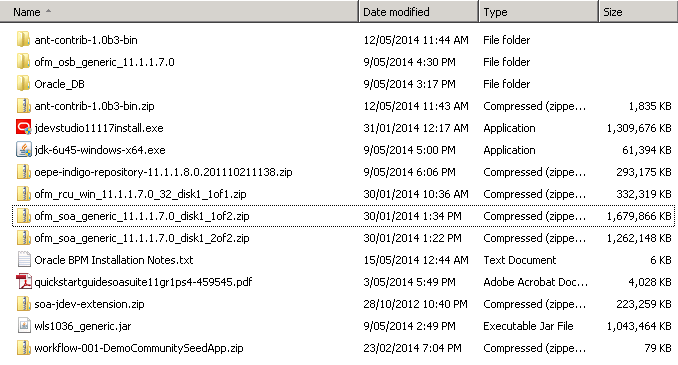
**Step 1: Download all necessary packages for installation**

Oracle SOA Suite can be downloaded from the link below.

<http://www.oracle.com/technetwork/middleware/soasuite/downloads/soasuite11gdownload-2210918.html>

JDK and IBM JRocket can be download from Oracle JDK web site.

The downloaded files can be saved into one folder as shown below.



It is important to make sure all packages to install are of the same version. For example, the above components have the same version number as 11.1.1.7. Incompatible components might not work with each other, for example JDeveloper 11.1.1.6 may have issues to run smoothly with SOA 11.1.1.7.

The key packages in the installation folder are summarized in the following table.

|  |  |  |
| --- | --- | --- |
|  | Installation | Installation files |
| 1 | SOA Suite | Ofm\_soa\_generic\_11.1.17.0\_disk1\_1of2.zip  Ofm\_soa\_generic\_11.1.17.0\_disk1\_2of2.zip |
| 2 | Repository Creation Utility | Ofm\_rcu\_win\_11.1.1.7.0\_32\_disk1\_1of1.zip |
| 3 | JDeveloper | jdevstudio11117install.exe |
| 4 | Oracle Weblogic Server | wls1036\_generic.jar |
| 5 | Oracle OSB (optional for this tutorial) | Ofm\_osb\_generic\_11.1.1.7.0 |
| 6 | JDK Server edition 64 bit | Jdk-6u45-windowx-x64.exe |
| 7 | Oracle Database Server 11g R2 | Oracle DB |
| 8 | Demo Community (to create demo users) | Workflow-001-DemoCommunitySeedApp.zip |

These components will be explained in the next steps.

**Step 2: Install Oracle Database 11g R2**

Oracle Database server is needed to store schema for Oracle BPM/SOA, meaning they are server application with their own database. Thus they need a DBMS for data management.

Follow standard installation for Oracle Database Server.

Note that the character set for database should be chosen as Unicode UTF8 since it is required by Oracle SOA/BPM schema.

**Step 3: Install JDK**

The JDK should be 64-bit and server edition to be able to accelerate the performance of BPM/SOA engines by taking advantage of more than 4GB RAM space.

It is recommended to install Oracle JDK 6 or IBM JRockit, both should be 64-bit server edition. There might be compatibility issues with different Java versions.

After JDK installation, make sure you can set up the default java home on your system to the new java installation folder because the default call “java” in installation script may point to a different java installation on your system.

**Step 3: Install Weblogic Server**

Oracle SOA/BPM is JEE application to be run within a JEE container. Oracle Weblogic server is used as a JEE container.

The installation process for Oracle Weblogic server is normally standard and smooth.

**Step 4: Install Database Schema**

Note that Oracle SOA/BPM is nothing but a JEE server application and it uses database to store it design-time and run-time data. The data can be process instances, process runtime data, configuration parameters, audit data, business activity monitoring data, etc.

In particular, you should note that the MDS database is installed. MDS is used as a common database store for process data in Oracle SOA platform.

The Repository Creation Utility (RCU) is used for database installation. A database server must have been installed and a database server instance has been started before running this step. Also make sure that you can provide the right database connection parameters when asked: server name, instance id, username.

**Step 6: Install Oracle SOA**

Once the Weblogic server and database tables have been in place, we now can deploy Oracle SOA as a JEE application to Weblogic Server in this step. It is a large-scale server application including SOA Infrastructure, BPEL engine, BPMN engine, Human Workflow Engine, Business Rules Engine, etc., including the console applications used to administer these engines (Enterprise Manager).

This is often referred to as BPM run-time architecture.

**Step 7: Create SOA Domain**

Referring to the deployment architecture in section 2, Oracle SOA manages its infrastructure logically with host, domain, clusters and managed server concepts. This step is to generate these administration data in the management database of Oracle SOA. There is no software to install in this step but only configuration data to create on the basis of web logic server, database server and SOA server application.

You select a specific way you want SOA server to organize BPM/SOA applications in the future: by domain, cluster, standalone server or managed servers, etc.

**Step 8: Install JDeveloper**

After the SOA infrastructure has been established, now it’s time to install Oracle JDeveloper as a design and development environment.

Noted that JDeveloper is offered by Oracle as a sole tool for all types of Java-based development projects, not only BPM application.

For BPM application development, you should add some design components: SOA Composite editor and BPMN editor. Otherwise, you will not find out where to design the BPMN process model and links to key process-related concepts: business objects, human resources, human tasks, rules, etc.

**Step 9: Create Demo Users**

This step is to create some demo users for our tutorial. Oracle BPM allows us to define logical roles in a process. At run-time, these roles are translated to real users on the system by connecting to a LDAP server provided with Oracle Weblogic Server. This role-user mapping can be created in the process at design-time.

In order to have users available for the design-time mapping, Oracle provides a script to add a list of users of a fictitious company to Weblogic Server.

The script can be found at the following link.

[http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/obpm/11g/r1/install/files/DemoCommunitySeedApp.zip]DemoCommunitySeedApp.zip](http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/obpm/11g/r1/install/files/DemoCommunitySeedApp.zip%5dDemoCommunitySeedApp.zip)

You download the script and follow the instructions to install users.

**Management console and tools**

After successful installation, you can access Oracle SOA/BPM consoles and tools as follows.

Oracle Enterprise Manager

<http://localhost:7001/em>

Oracle BPM Workspace

<http://localhost:8001/bpm/workspace>

Oracle BPM Composer

<http://localhost:8001/bpm/composer>

Oracle Business Activity Monitoring

<http://localhost:9001/OracleBAM/>

Oracle Service Bus

<http://localhost:7001/sbconsole>

Oracle Admin Server Weblogic Administration

<http://wflab02.qut.edu.au:7001/console>

Oracle WebLogic for each server: run from the Enterprise Manager console.

# Navigating Oracle JDeveloper environment

Oracle JDeveloper is designed as an integrated development environment for various types of projects: Java desktop, web, mobile, BPM project, etc. It is claimed as a strength of Oracle BPM environment when different roles (business analysts, developers, designers and others) can use one tool only for their duties. However, due to its richness, it turns out to be highly complicated tool to quickly grasp and easily become overwhelming to a new user.

This section explains main elements of this environment so as to help you get acquainted to this tool. The essential skill is to understand the meaning of every structural element and where to find them when you need to.

First of all, readers of Oracle BPM documentation may be confused with many design editors such as BPM Studio, SOA Composite Editor, BPEL Process editor and so on; however, they will not be able to find any of them as a software. Actually they are parts of Oracle JDeveloper installation and exist in one tool only, not multiple tools. You open Oracle JDeveloper and it contains all those design editors as interface components.

The main screen of Oracle JDeveloper is shown in Figure XX with some main elements.

Design space

Component Palette

Application Navigator

BPM Project Navigator

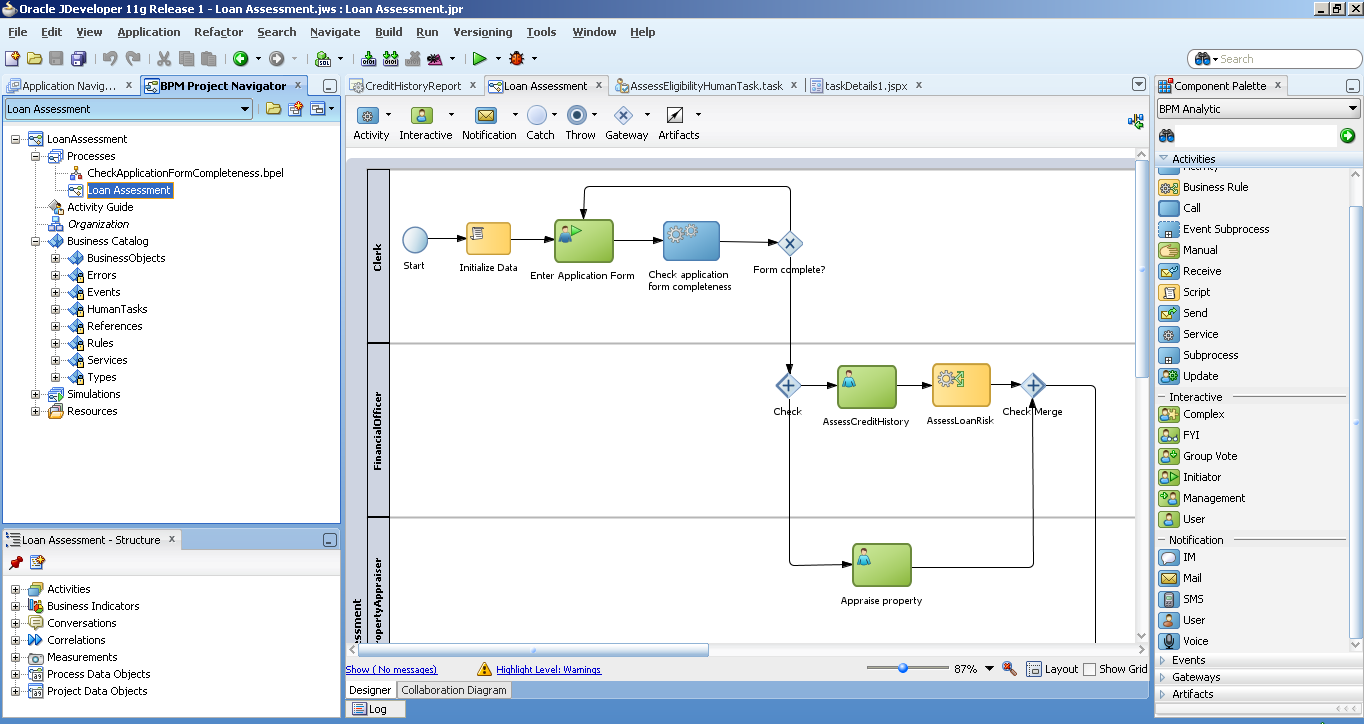


Figure 1: JDeveloper environment

We’ll focus on the primary design panes as listed below in this section. Other supporting panes will be explained in details during the tutorial.

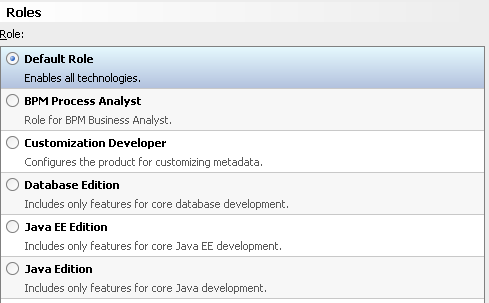
* BPM Project Navigator
* Application Navigator
* BPMN Editor
* BPEL Editor
* ADF Page Editor
* Component Palette
* Property Inspector

BPM Project Navigator and Application Navigator are two main explorers to open the design artefacts which will be shown in other design panes.

The design panes above are arranged by default into docking windows as displayed in Figure 1. The window has basic facilities for a normal IDE: moving, floating, docking, pin, unpin, minimize, maximize. You should spend a little time to familiarize yourself with these window features. You can also use View menu to find and unhide a particular pane.

At times you may encounter that a pane is invisible though you have selected to show it through the View menu. In that case you can restore the main screen to its standard arrangement by selecting Window menu > Reset Windows to Factory Settings. The whole JDeveloper window will be reset and the pane can be navigable again through the View menu.

The interface is configurable for different user role to simplify the number of accessible design panes. In case you cannot find a particular pane, the reason can be it is not relevant for the chosen role. You can select to make all panes visible in Oracle JDeveloper by going to Tools menu > Preferences > Roles and select “Default Role”.



**BPM Project Navigator**

BPM Project Navigator contains logical elements of a BPM project such as process, organization, business catalog, simulations. Therefore, you should open this pane whenever you need to work with these BPM concepts.

|  |  |
| --- | --- |
| It is organized in a tree view with the top node is the project name. Pre-defined sub-nodes represent concepts of a BPM project.  The main node is Processes which allows us to create BPMN and BPEL processes.  The Organization is for people perspective of the process.  Business Catalog contains most of the elements of a process as seen in the figure. When you design a process model, these elements will be arranged into these folders.  Activity Guide, Resources and Simulations: not used in this tutorial. |  |

**Application Navigator**

A BPM application is structured as a group of projects, in which there is a BPM project with BPM resources such as BPMN process model file, BPEL model file, business rule model file, human task model file, etc., and several Web projects, each containing web page implementation for human tasks in the BPM project. Application Navigator is thus a tool to navigate and edit each project in the BPM application.

|  |  |
| --- | --- |
| As outlined in the figure on the right, Loan Assessment is the BPM project whereas others are Oracle ADF projects which implement the human tasks in the Loan Assessment process.  The BPM project has structure as:   * Application Resources * BPM Content * SOA Content   Every Oracle ADF project has three primary folders:   * Application Resources * Resources * Web Content   Application Navigator shows physical files in a project rather than a logical organization. So if you want to have a deeper look at the file format, implementation details or specification level, you definitely should explore the project with Application Navigator. |  |

If we look at files level, the application folder is shown in Figure XX and it is exactly the same structure as shown in the Application Navigator. The application master file has “jws” extension at the root folder. There is one folder for BPM project and one folder for each Oracle ADF project. A project master file has “jpr” extension.

It is not recommended to edit the project files directly (they are all text files) since most of them are auto-generated and managed via Oracle JDeveloper. Modifying them outside of Oracle JDeveloper may break the integrity of the application or project and cause errors.

|  |  |
| --- | --- |
|  |  |

**Design Editor, Structure, Component Palette and Property Inspector**

There are three main design editors used for a BPM application: BPMN editor, BPEL editor and Oracle ADF editor. Editor is used to open and edit a design (default location is in the center of the main screen). Each design, either BPMN process model or BPEL process model or ADF page, has a structure which can be viewed via the Structure pane (default location is at the bottom left corner of the main screen). Each design type is also accompanied with a set of design objects shown in the Component Palette (default location is on the right of the main screen). Each design object has a set of associated properties shown in the Property Inspector pane (default location is on the right of the main screen).

The design editor has a visual display of its elements. When we open a design editor, the Component Palette will open the corresponding component set for that editor. When we click on a component in the design editor, the Property Inspector will show properties of that component.

The figure below shows some examples of the three main editors for BPMN, BPEL and Oracle ADF.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**Working with a Design Editor**

All design editors have common and repeatable design actions:

* Support visual view
* Support drag and drop from the Component Palette
* Support Undo and Redo (shortcut key is Control-Z)
* Support a context menu with a right click on a design object

# Create a new BPM application project

When login to Oracle JDeveloper, remember to select “All Roles” so that we have all privileges in the system for project activities in the following instructions.

To open a new project for Loan Assessment process, click File > New, select General > Applications > BPM Applications (Figure XX). Click OK.

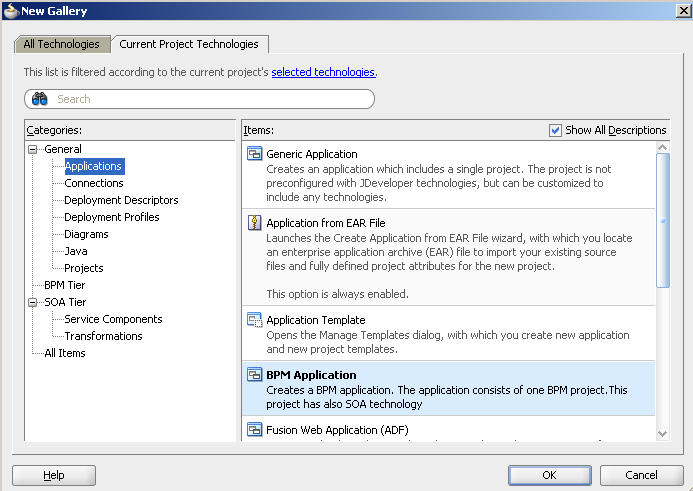


Figure 2: JDeveloper environment

In the new Wizard window, type Project Name: “LoanAssessmentProcessApp”. Select a folder for the application. Choose an application package prefix, e.g. “com.loanassessment”

Click Next.

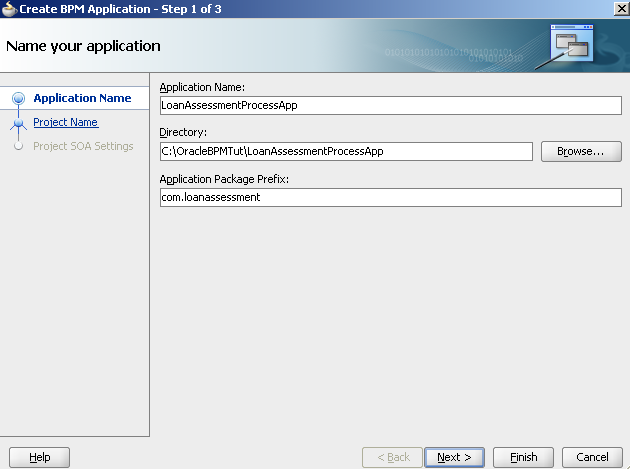


Figure 3: JDeveloper environment

In the new window, enter a name for the project (“LoanAssessmentProcessApp”). Note that a folder is selected for the project which is a subfolder of the application folder. This is a good practice because an application might have multiple projects, each for different components, particularly GUI ADF components. We’ll create more projects later on.

Predefined technologies such as BPM and SOA have been selected by default for the project.

Click Next.

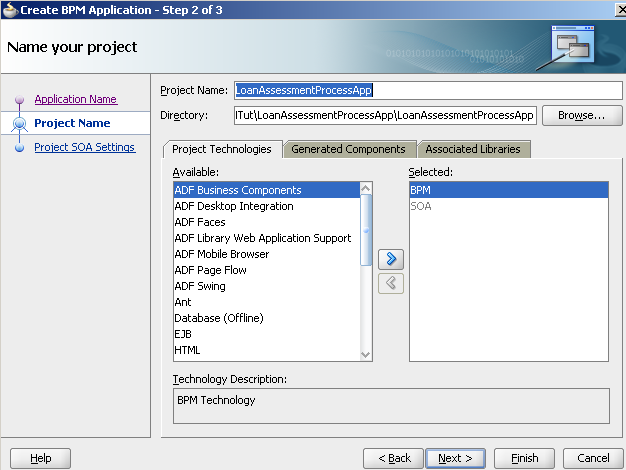


Figure 4: JDeveloper environment

In the new window, select “Composite with BPMN process” option. This means we are going to design BPMN process and create BPMN-based process application as a composite application.

Click Finish.

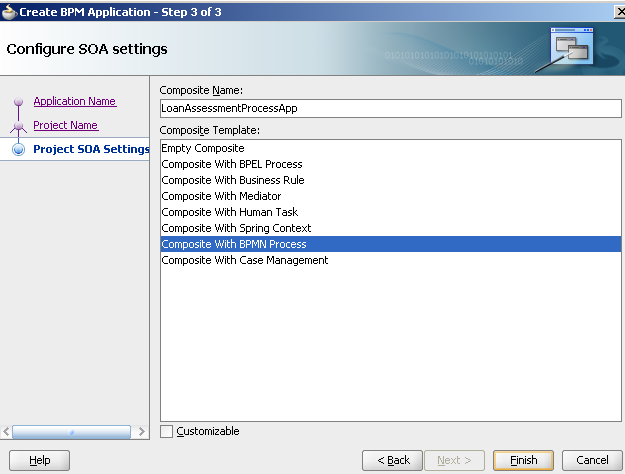


Figure 5: JDeveloper environment

A Wizard will display to select style of process we want to create first. Choose “Manual Process” since our process will have many interactive activities.

Name the process: “Loan Assessment”.

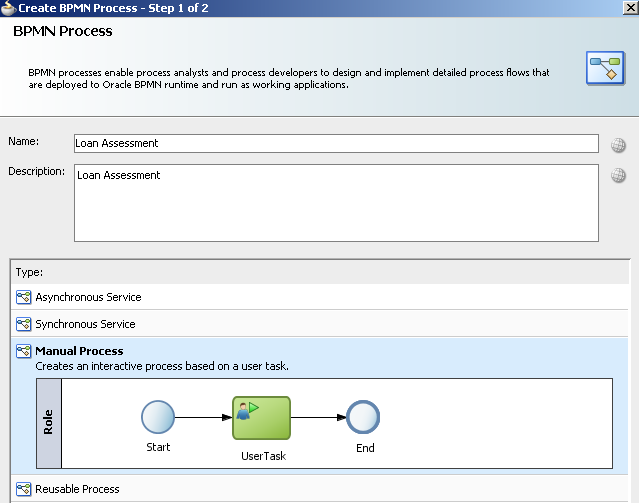


Figure 6: JDeveloper environment

Just click Finish in the next window with default values (Sampling points are defined ways of collecting data for process analytics, not the focus of this tutorial).

A new project will be created and shown in the design environment as shown in Figure XX. It has a lane called “Loan Assessment” and a default role called “Role”. There is a start event, end event and the first activity called “UserTask”.

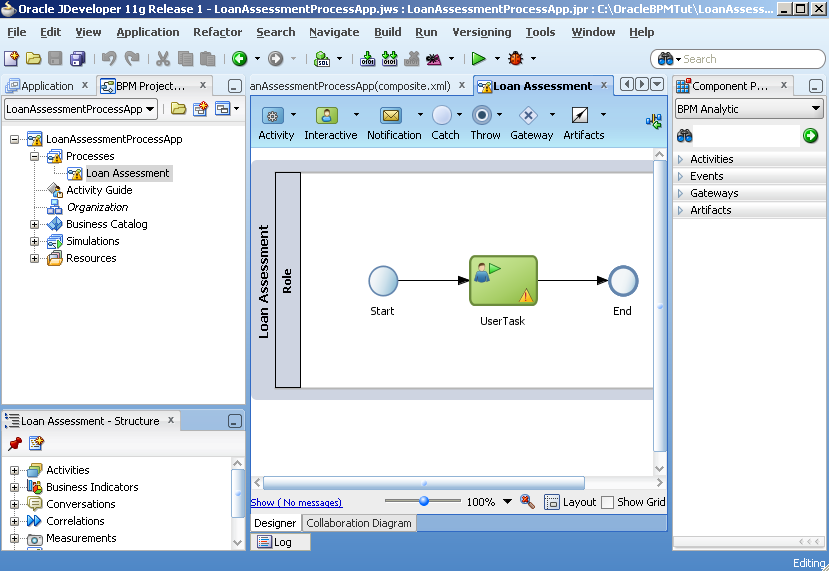
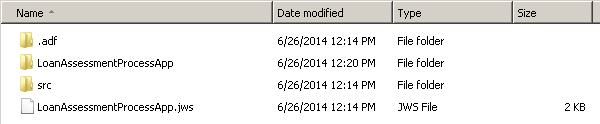


Figure 7: JDeveloper environment

In project folder, a hierarchy of files are generated as shown. File “.jws” is the project file.



# Model the Process

Starting from the initial process model, we can drag and drop modelling widgets from Component Palette onto the BPMN editor. You can see that Oracle BPM has a strong support for BPMN 2.0 notation along with additional elements such as Script, Email notification, Service call, and Business rule.

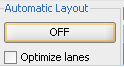
The primary stages to model a process in BPMN are as follows.

* First, we add the process lanes (also called roles) since process lanes can be quickly understood from the process scenario and help form an initial structure for the model
* Then, we add main process activities which should cover the process scenario as much as possible
* Then, we add ancillary activities to implement particular processing requirements if not covered in previous stage.

**Turn off Automatic Layout**

You should notice to turn off the automatic layout setting of the BPMN editor, otherwise it may change the layout of the model out of your control and take your much time in re-layouting the model again.

In the BPMN editor, click on the Layout button at the bottom area. If the setting in the popup is ON then turn it to OFF.

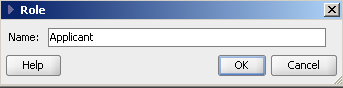


## Add process lanes

1. Open the initial process model in BPMN editor (in BPM Project Navigator, expand the project > Processes, double click on “Loan Assessment” process).
2. Select the start event, the UserTask activity and stop event. Press Delete button to delete them all. We clean them so that we focus on lanes at the moment. We’ll add activities later.
3. Right click on the current Role in the model called “Role”, select Properties. In the Role properties, select New.



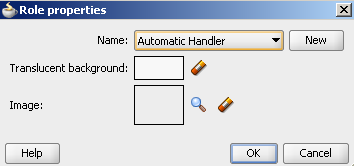
In the Role popup, type “Applicant” and click OK.



“Applicant” role will be created and replace the role “Role” in the model.



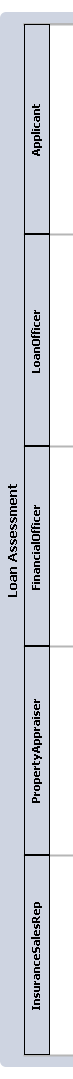
1. Right click on the empty space of the process model, select “Add Role” in the menu. In the Role properties, select New.



Type “Loan Officer” and click OK. A Loan Officer role is created and added below the Applicant role in the model.

1. Repeat the same steps to create Financial Officer, Property Appraiser and Insurance Sales Rep role. Note that the role name does not allow white space. You can drag and drop the role on each other to adjust the order of them.

The new process model with all roles (lanes) should look similar to Figure XX.



## Add main activities with sequence flows

Open the process model in BPMN editor.

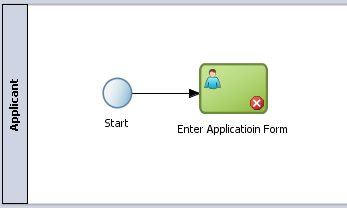
**Add “Start” event**

1. In Component Palette, expand WeEvents.
2. Drag “None” component from “Start Events” section onto the Applicant lane.
3. In “Properties – ThrowEvent” pop-up, enter “Start” in the Name field, then click OK.
4. A Start event will be added to the Applicant lane. Ignore the red cross at the moment.

**Add “Enter Application Form” activity**

1. In Component Palette, expand Activities, drag a User activity from Interactive section onto the empty space of Applicant lane.
2. In the Properties – UserTask1 pop-up, enter a value for the Name field: “Enter Application Form”. Click OK.
3. “Enter Application Form” activity is created and shown in the Applicant lane.
4. Drag Start event and “Enter Application Form” activity to adjust them to be horizontally aligned and from left to right order, respectively.
5. In Component Palette, expand Artifacts. Click on Sequence Flow to select it and then move and hover your mouse over the Start event. The mouse icon changes to a plus sign. Click the mouse on the Start event and drag it onto “Enter Application Form” activity and drop there. A sequence flow will connect the Start event to “Enter Application Form”.

Note that you can always click on the sequence flow, hold and move it to adjust the connection line.



**Add Timer event for “Enter Application Form” activity**

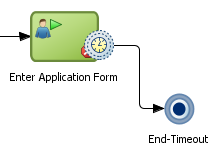
The scenario is if the “Enter Application Form” lasts for more than 5 days, the system will terminate that step and the process stops.

Modelling steps:

1. In Component Palette, expand Events
2. Drag Timer from “Catch Events” section and drop it on the edge of the “Enter Application Form” activity.
3. In the “Properties – BoundaryEvent” popup, enter “TimeOut” in the Name field then click OK.
4. The timer will attach to the edge of the activity shape as shown below.



1. Drag a Terminate event from Component Palette, End Events section to the model on Applicant lane.
2. In the “Properties – End” popup, enter “End – Timeout” in the Name field. Click OK.
3. A Terminate event will be added to the model
4. Drag a Sequence Flow to connect the Timer event of Enter Application Form activity to the Terminate event. Note that you can always click on the sequence flow, hold and move it to adjust the connection line.



**Add a Service Call for checking form completeness**

The scenario is the system will check the completeness of the application form automatically and returns a result whether the form is complete or not.

Modelling steps:

1. Drag a Service activity component from Component Palette, Default section to the Loan Officer lane but locate it after the Enter Application Form activity.
2. In the “Properties – ServiceTask” popup, enter “Check application form completeness” in the Name field. Click OK.
3. A Service Call activity will be added to the model.
4. Connect the Enter Application Form activity to the Check Application Form Completeness with a sequence flow.

**Add an Exclusive Gateway for form completeness decision**

The scenario is the process checks the result of “Check Application Form Completeness” activity. If the form is complete, it proceeds to the next activity. Otherwise, the process returns the application form to the Applicant to supplement the missing or invalid information.

Modelling steps:

1. Drag an Exclusive Gateway from Component Palette, Gateways section to the Applicant lane in alignment with the Enter Application Form activity but after the Check Application Form Completeness activity.
2. In the “Properties – ExclusiveGateway” popup, enter “Complete?” in the Name field. Then click OK.
3. An Exclusive gateway will be added to the model.
4. Connect Check Application Form Completeness activity to the “Complete?” gateway with a sequence flow.
5. Connect “Complete” gateway back to Enter Application Form activity with a sequence flow.

**Add a Parallel gateway for directing process flow to two activities at the same time**

The scenario is if the form is checked as complete, the system will forward the form to Financial Officer and Property Appraiser at the same time for assessment.

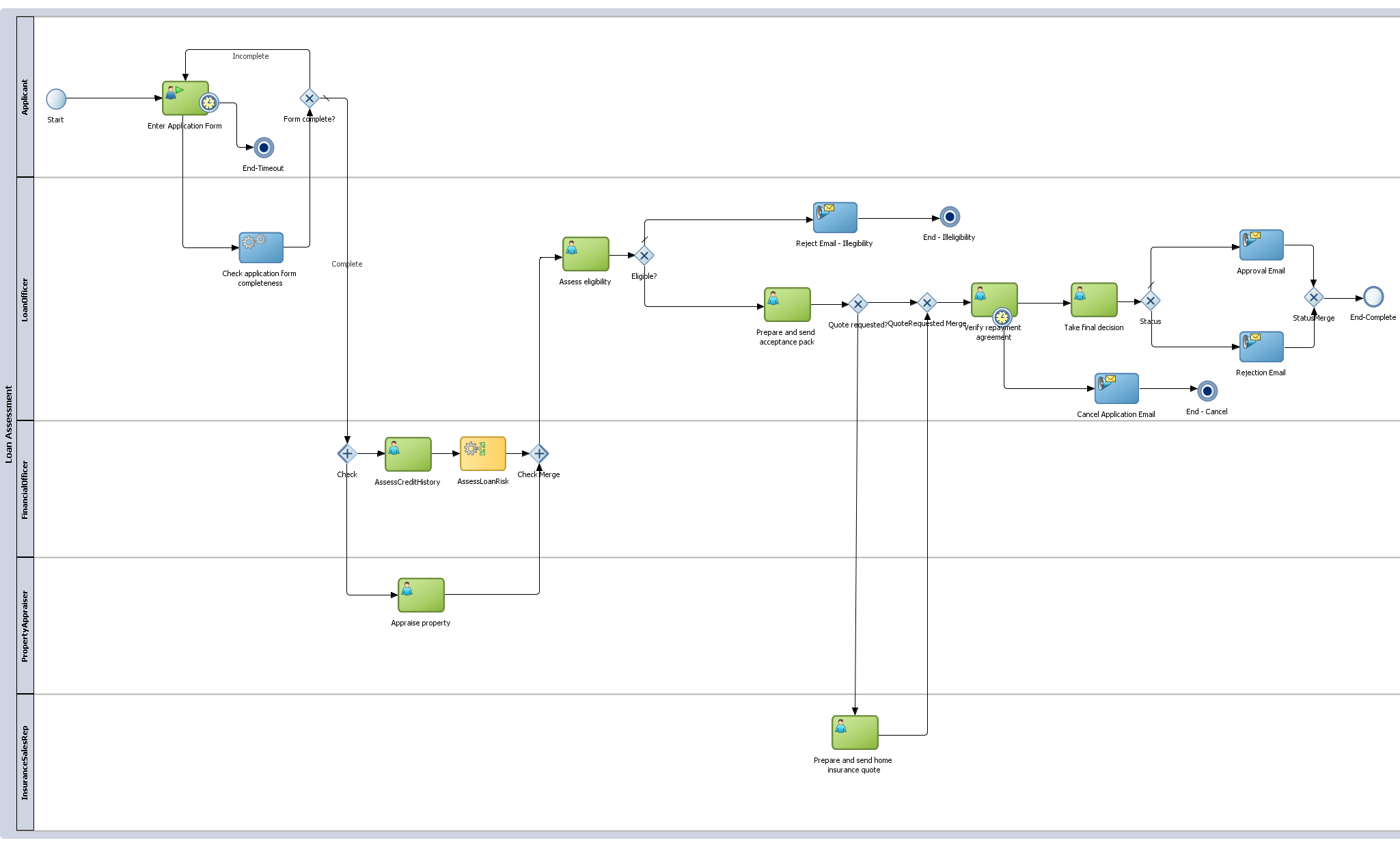
Modelling steps:

1. Drag a Parallel Gateway from Component Palette, Gateways section to the Financial Officer lane and place it after the “Complete?” gateway horizontally
2. In the “Properties – ParalellGateway” popup, enter “Check” in the Name field. Then click OK.
3. A Parallel gateway will be added to the model.
4. Connect “Complete?” gateway to the “Check” parallel gateway with a sequence flow.

At this point, we have been quite familiar with different types of components, Component Palette and sequence flow. Let’s apply the same for the remaining process scenarios. The process model at the end of this stage should be modelled and connected similarly to Figure XX.

Below is all remaining components listed in the order of their arrangement in the model.

* Assess Credit History activity, User activity component.
* Appraise Property activity, User activity component
* Assess Loan Risk, Business Rule component
* Parallel gateway for merging parallel flows from Assess Credit History activity and Assess Property activity.
* Assess Eligibility activity, User activity component
* Eligible? Exclusive gateway
* Reject Email – Illegibility activity, Mail activity component
* Terminate event after Reject Email – Illegibility activity, Terminate event component
* Prepare and send acceptance pack activity, User activity component
* Quote Requested? Exclusive gateway
* Prepare and send home insurance quote, User activity component
* Quote Request Merge exclusive gateway
* Verify Repayment Agreement activity, User activity component
* TimeOut timer for Verify Repayment Agreement activity
* Cancel Application Email activity, Mail activity component
* Terminate event after Cancel Application Email activity, Terminate event component
* Take Final Decision activity, User activity component
* Status exclusive gateway to check Approval status after Take Final Decision activity
* Approval Email notification, Mail activity component
* Rejection Email notification, Mail activity component
* StatusMerge exclusive gateway
* End – Complete event, “None” event component.



## Add ancillary activities

In the next modelling steps, we will add some implementation specific activities in accordance with some process scenarios.

**Initialize process data**

An arising scenarios here is Oracle BPM engine cannot initialize the process data automatically since our process data is complex with compound data objects (data object contains another data objects and so on). We have to do it explicitly in the process. Intuitively we may encounter this requirement only after we have implemented data objects for the process (explained in the succeeding section). However, for business process with potentially complex data objects, this issue can be anticipated beforehand and thus the in-charged business analyst can perform this modelling action.

Modelling steps:

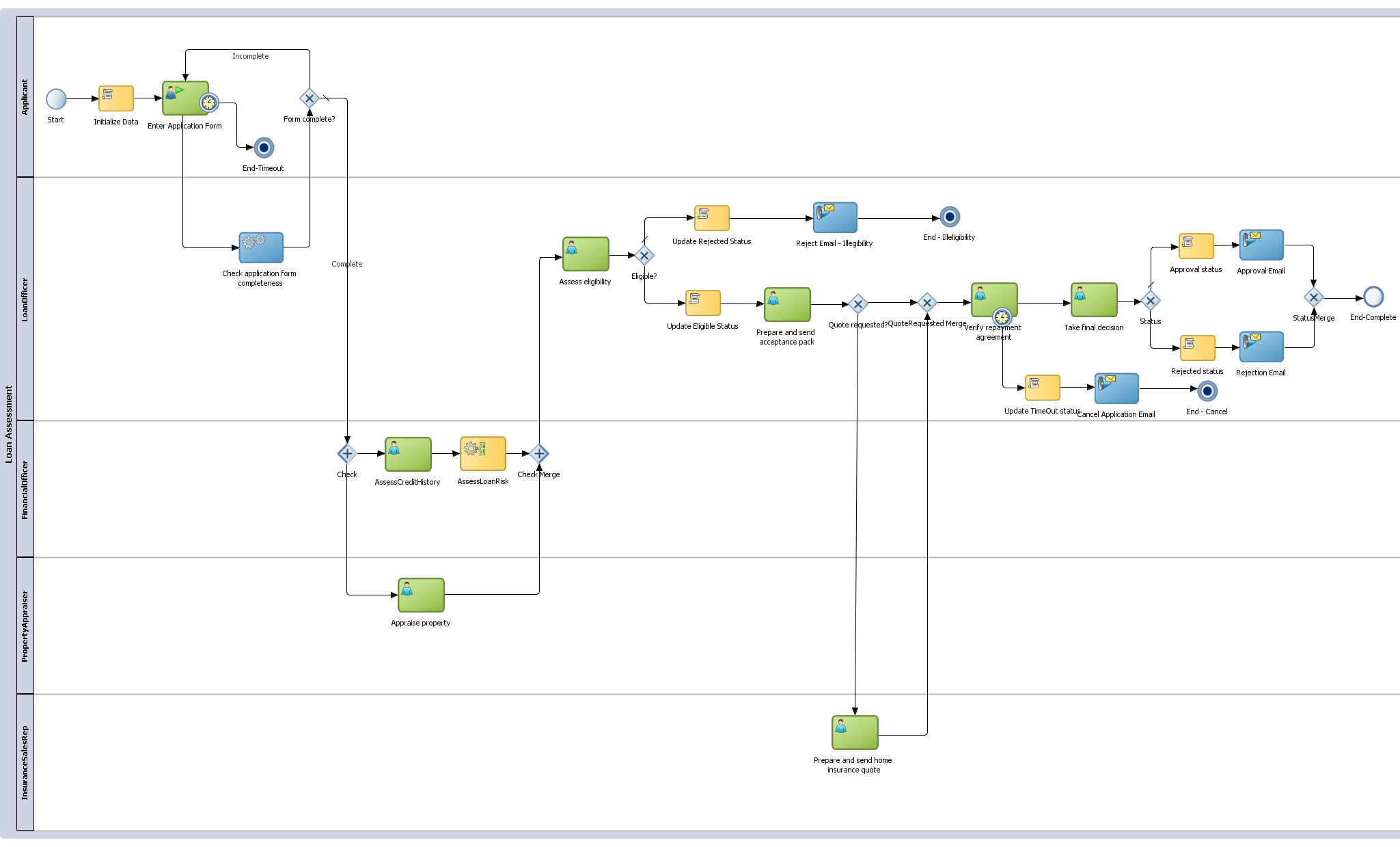
1. Drag a Script activity component from Component Palette, Activities section to Applicant lane, drop it on the sequence flow between the Start event and the Enter Application Form activity.
2. In “Properties – ScriptTask” popup, enter “Initialize Data” in the Name field. Then click OK.
3. A script activity will be added to the model with sequence flow is reconnected from Start event to the Initialize Data activity and from that activity to Enter Application Form activity.

**Update application status**

The scenario is the application status will be updated after some particular activities, such as after Assess Eligibility activity or Take Final Decision activity.

Modelling steps: we use a Script activity component and follow the same steps as those for Initialize Data activity.

The process model at the end of this stage should look like the one in Figure XX. This also completes process modelling activities and we can move on to process implementation in the next section.



# Implement the Process

In this section, we will add implementation details for all modelling components added in preceding section. However, prior to creating any implementation details, there are some vital preparation we need to do including initial setup and data objects.

## Initial set-up

Some configurations need to be set up prior to implementation activities. You might need assistance from your administrator for system parameters (username, password, port number, etc.).

### Server Connection

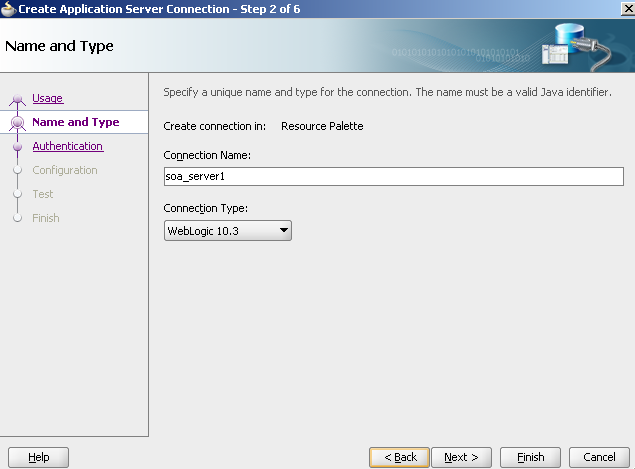
Click View menu > Application Server Navigator.

Right click on Application Servers, choose “New Application Server…” to open a Create Application Server connection dialog.



Click Next in the first screen (with default value is Standalone Server which is this tutorial installation).

Type any name without spaces in Connection Name.



Enter system information including username, password, server host name, port, domain name.

Click Test Connection.

If the test does not succeed, one of the cause may be firewall block connection to the server.

### Email Notification

In order to implement Mail activity, we need to configure the User Messaging Service (UMS) on Weblogic Server so that it can send email. One way is to install a mail server and set up connection between UMS to the mail server. Another way is we can configure UMS to connect to Google mail server as a relay server to send email. In this section we’ll use the second way for simplicity.

It is required to have Internet connection to send email from Oracle SOA server to an external email. Make sure that the connection is available on the server.

The instruction below is from an Internet resource at <http://www.soatutor.com/2012/10/setting-up-email-notification-in-oracle.html>.

There are four steps to configure email notification.

Step 1: Import certificates from gmail and add it to your server trust store

Step 2: Configure email driver properties

Step 3: Enable notification mode

Step 4: Testing the configuration

**Step 1: Import certificates from gmail and add it to your server trust store**

Any email server uses two protocols to send/receive messages. SMTP for sending mails. Either POP3 or IMAP for receiving mails. Gmail uses IMAP for retrieving mails.

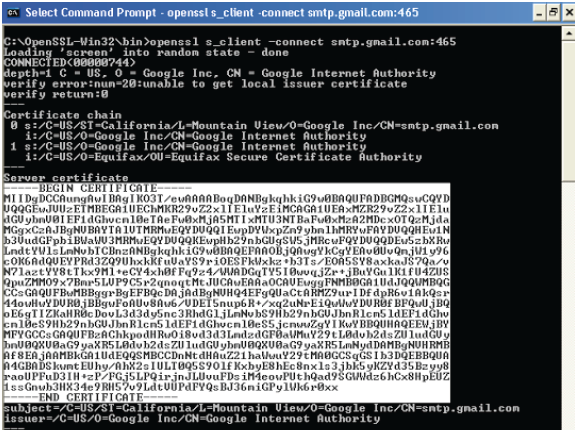
So, you need to get both SMTP and IMAP certificates connecting to the gmail server in order to send/receive mails to/from your inbox.

You can download the certificates using an open source software called openssl. First, you need to download and install it.

Open command prompt and cd to openssl\_install\_folder/bin. Give the below command to view the SMTP certificate.

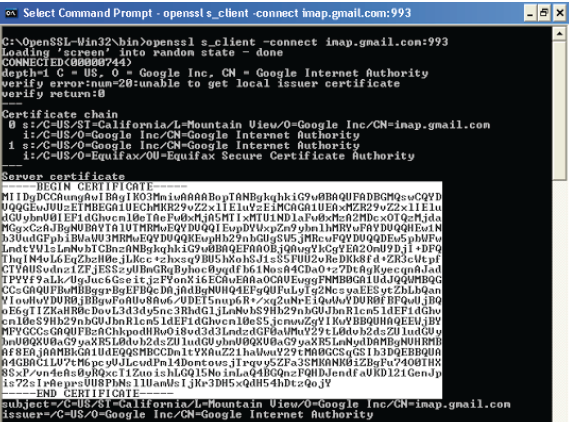
openssl s\_client -connect smtp.gmail.com:465

Copy the code highlighted in the picture above and paste it to a file and name it smtp\_gmail.cert



Similarly, issue the follwing command to view the IMAP certificate

openssl s\_client -connect imap.gmail.com:993



Now that you have both smtp and imap certificates with you, you need to import these to your server trust store

For this, open command prompt and navigate to %JAVA\_HOME%/bin. Issue the following command

keytool -import -alias smtp.gmail.com -keystore trusted-certificates.jks -file <location of smtp\_gmail.txt>

keytool -import -alias imap.gmail.com -keystore trusted-certificates.jks –file <location of imap\_gmail.txt>

Once you are done with importing the trust certificates to the keystore using the keytool, you need to tell the managed server(soa\_server1) that there is a user defined trust store from which it has to look for keystore.

This will be done by editing the %MIDDLEWARE\_HOME%\user\_projects\domains\soa\_domain\bin\setDomainEnv.cmd file

Search for *-Djavax.net.ssl.trustStore* and replace the value with the the trusted-certificates.jks file path that was generated by the keytool command. Also you need to edit -Djavax.net.ssl.trustStorePassword (if not available, create one). So finally your entries should look somewhat similar to this

-Djavax.net.ssl.trustStore=D:\oracle\Middleware\jdk160\_29\bin\trusted-certificates.jks -Djavax.net.ssl.trustStorePassword=welcome1

Once you are done with this edit, one step is pending, where you will tell the managed server that a custom keystore is setup and has to be considered.

This is done by opening the Admin Console(<adminHost>:<adminPort>/console --> Environments --> Servers --> click on soa\_server1)

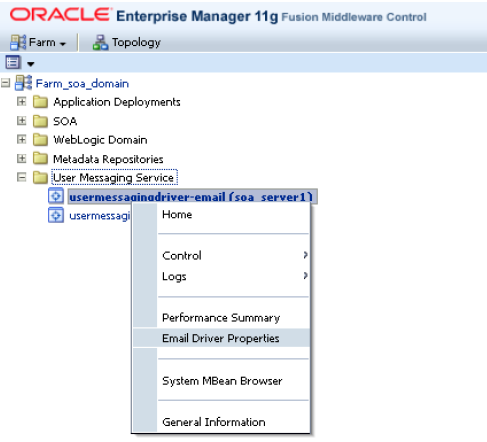
Click on Keystores, and change the Keystores to "Custom Identity and Java Standard Trust".

That finishes the certificates configuration.

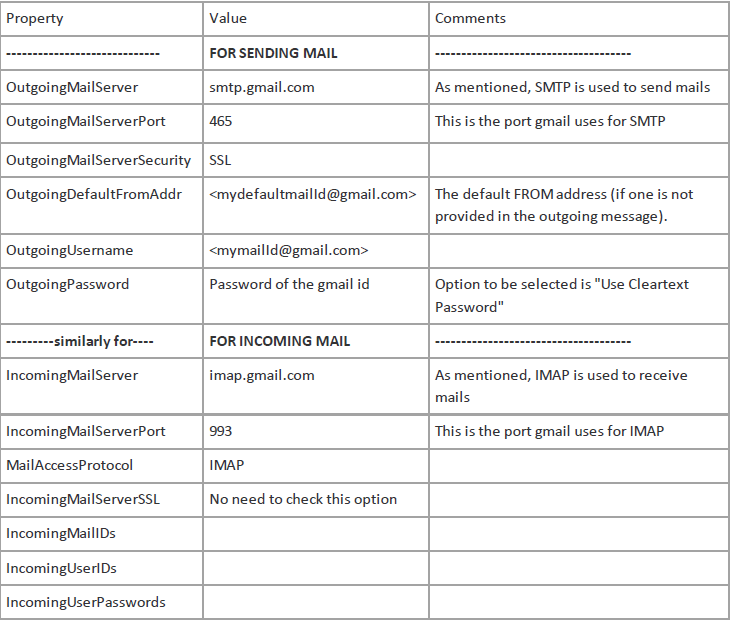
**Step 2: Configure email driver properties**

This step configures email driver properties like email server details, incoming/outgoing email, passwords, etc.

For this, open EM, traverse as shown to open Email Driver Properties Screen.



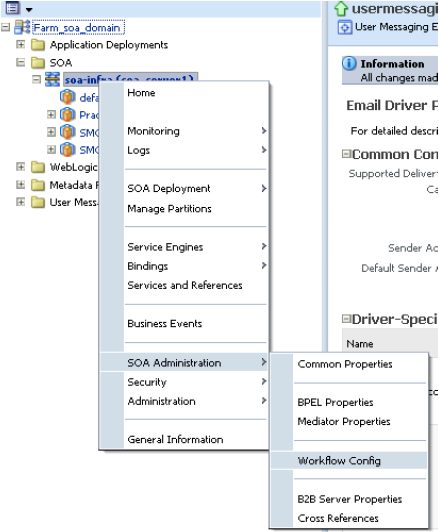
Configure the below mentioned properties



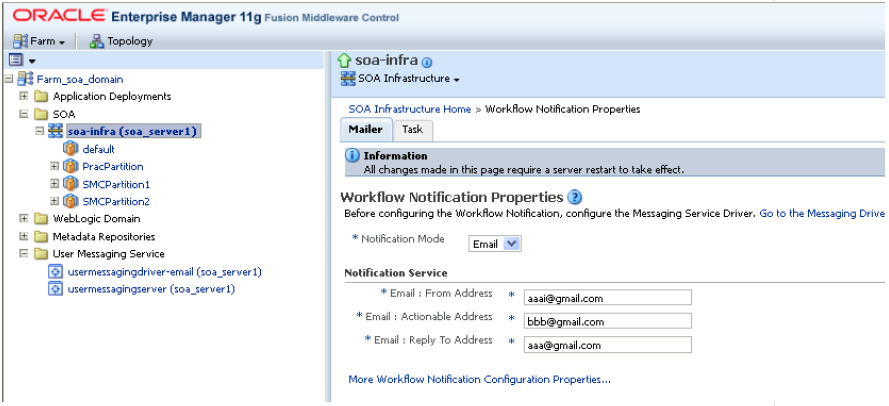
**Step 3: Enable notification mode**

This step lets the server know what mode to use for notifications. Since that we've configured email notification above, we'll enable EMAIL notification mode.

Traverse to *WorkFlow Config* in EM as shown in picture.



Setup the required values.



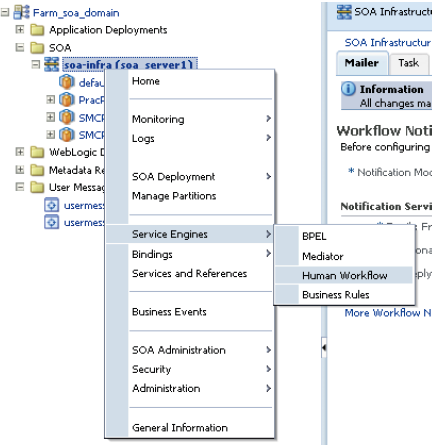
Restart the Admin (for step1) & Managed Server (for steps 2&3)

This sets up the required configurations.

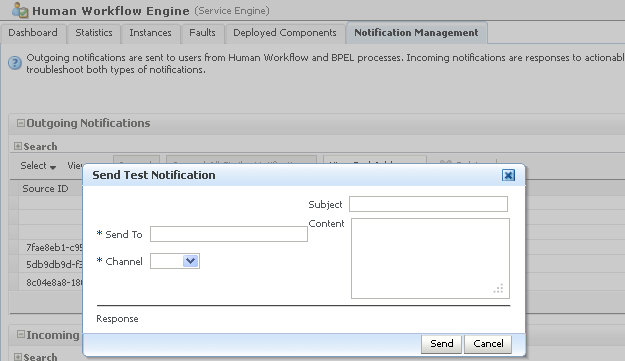
**Step 4: Testing the configuration**

The last thing that you need to do is test it.

Navigate to *Human WorkFlow* as shown



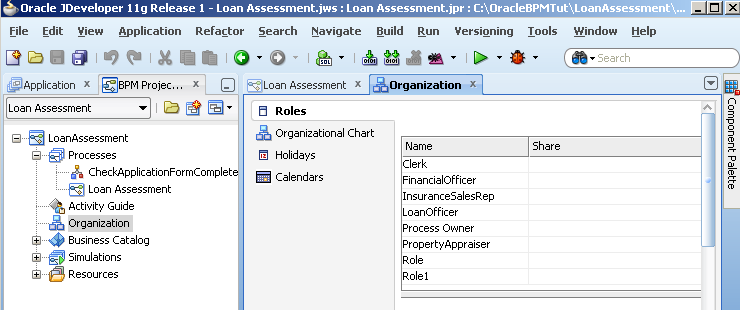
Notification Management --> Send Test Notification --> give details in the popup and check the mail.

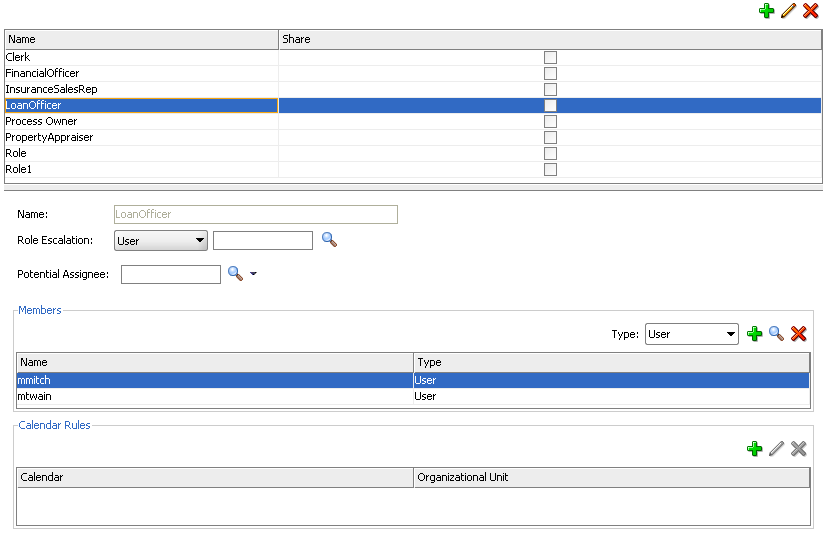


## Implement Organization

In terms of human resources to execute interactive tasks, Oracle BPM provides a mechanism to map from design-time role model to run-time user model.

In BPM Project Navigator, click on Organization. The list of roles defined in the process is shown in Figure XX. This environment allows us to add real users to each role and define availability hours (calendar) for users.



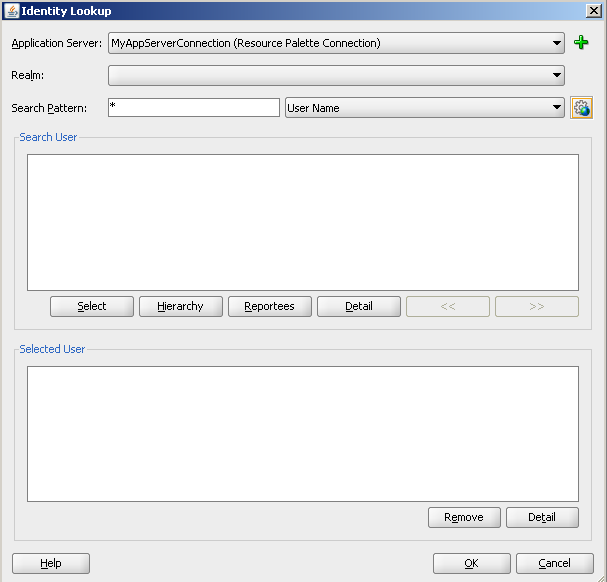


Each real user is granted a user ID in BPMS system by system administrator to access and execute human tasks on the system.

The screen in figure XX demonstrates how to map system users to design-time roles in BPMN diagram. The purpose is at run-time, these users will be routed to different activities according to routing rules. We’ll define these rules later in the Human task configuration.

**Add users to design roles**

For this tutorial, a list of users have been created in the system. Click on plus sign button in member panel will open an “Identity Lookup” window.



Select server connection as created in Section 7. Enter search keyword or search all.

List of users will be listed. These users include those demo users we have installed in section 3, step 9. Select users you want to add to the role. Repeat for each role.

My selected users for the tutorial are as follows:

|  |  |
| --- | --- |
| **Role** | **User** |
| Applicant | Jcooper |
| Financial Officer | Cdickens |
| Loan Officer | Mmitch, mtwain |
| Property Appraiser | rsteven |
| Process Owner | Wfaulk |
| Insurance Sales Representative | fkafka |

## Implement Data Objects

In BPM Project Navigator, navigate to Business Catalog > BusinessObjects. This function allows us to add data object types. Once these types are defined, we can add as many data objects as we want to the process.

Thus, there are two main steps for implementing data objects:

Step 1: Define Business Objects

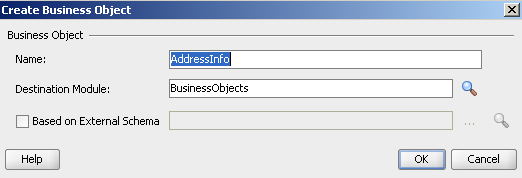
Step 2: Create Data Objects

**Step 1: Define Business Objects**

In BPM Project Navigator, expand Business Catalog > BusinessObjects.

Right click on BusinessObjects, select New > Business Objects.

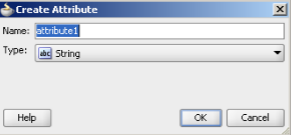
In the “Create Business Object” popup, provide business object name (without spaces), e.g. AddressInfo. Click OK.



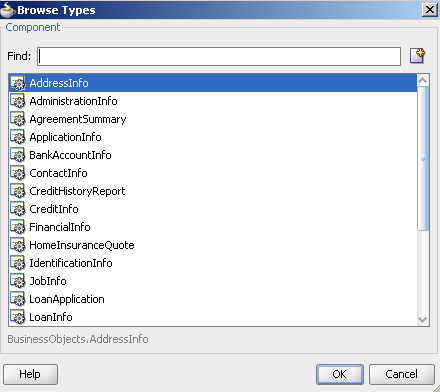
The business object will be created and shown in Business Catalog > BusinessObjects. Its detailed information is also shown as below (or double-click to open it if it is not shown).



Click on the green plus sign button to add a new attribute.

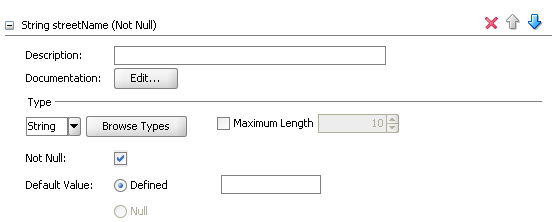


In the “Create Attribute” popup, select a name and type of the attribute. If the attribute is not primitive types (e.g. integer, string), we can select <Component> and <Array>. For Component or Array, we can select further a business object as base element as shown below. We need to select from XSD files which contain type definition for the object.



Click OK in the Create Attribute popup when you have filled in the Name and Type field.

The attribute will be added and shown in the detailed information of the business object.



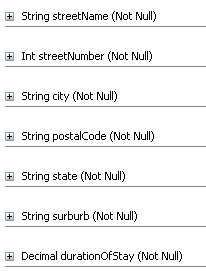
You can expand it to see more attribute properties. You can set some property values: type, maximum length, not null, default value, etc. Click Save on the menu to save the changes.

Repeat the same procedures above to add other attributes to the business object.

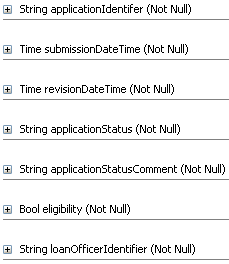
That’s the procedure. Now, let’s create all business objects needed for the process based on the process scenario.

Below is list of all business objects for this tutorial. They are listed with simple business objects first (object with primitive attributes), then compound business objects, (object with attributes can be another business object).

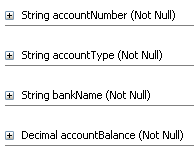
**AddressInfo**



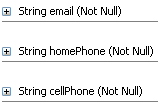
**AdministrationInfo**



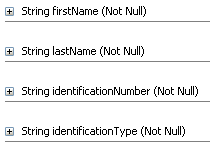
**BankInfo**



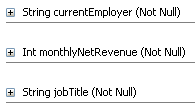
**ContactInfo**



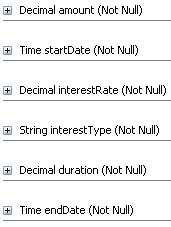
**IdentificationInfo**



**JobInfo**



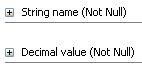
**LoanInfo**



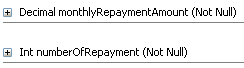
**PropertyInfo**



**SurroundingPropertyInfo**



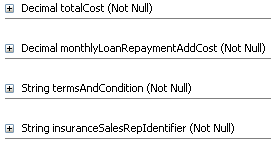
**RepaymentAgreement**



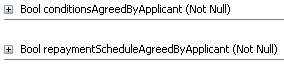
**RiskAssessment**



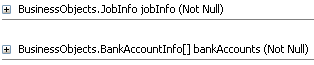
**HomeInsuranceQuote**



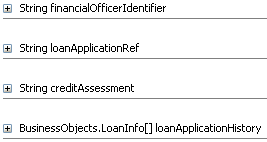
**AgreementSummary**



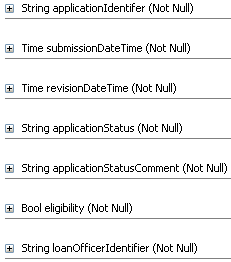
**FinancialInfo**



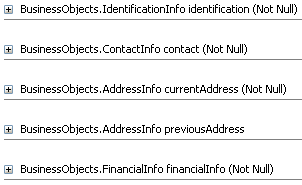
**CreditHistoryReport**



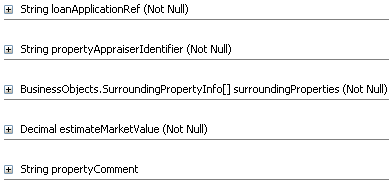
**AdministrationInfo**



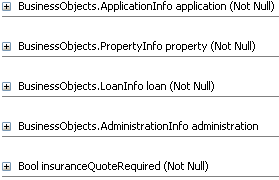
**ApplicationInfo**



**PropertyAppraiserInfo**



**LoanApplicationInfo**

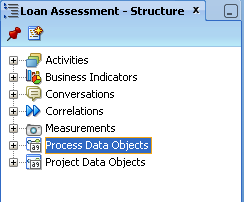


That finishes the business objects definition, i.e. the data object types. We can now create data objects for our process.

**Step 2: Create Data Objects**

In BPM Project Navigator, navigate to Processes > Loan Assessment process. Click on Loan Assessment process.

Note that the Structure pane shows the structure of this process as follows.



Right click on Process Data Objects, select New to create a new data object for the process.

In the Create Data Object popup which looks the same as the Create Attribute for business object, we can enter value for Name and Type field.



## Implement Human Tasks

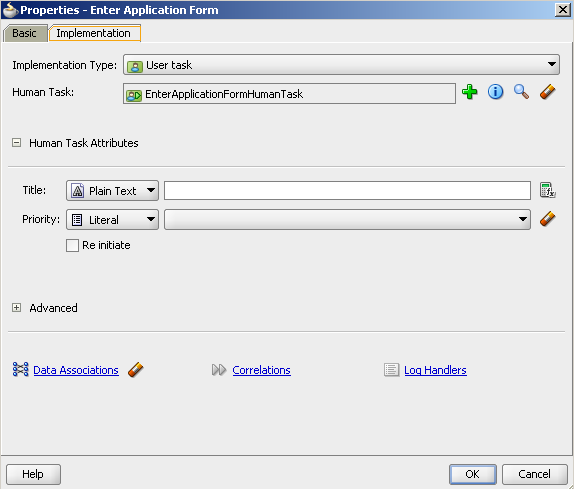
Human Task is an implementation of every user task we have modelled in the BPMN diagram. In this section, we will create human task for every user task, one by one. Note that Oracle BPM has two levels of user task implementation: human task and user interface. It is more manageable to separate them since they are at different levels of details. A user interface can be changed without need to change human task content.

For each human task, the implementation details depend on the task requirement and so we need to relate to the task scenario. Important scenarios include:

* Title
* Priority
* Outcome
* Owner
* Data
* User Assignment

### Enter Application Information

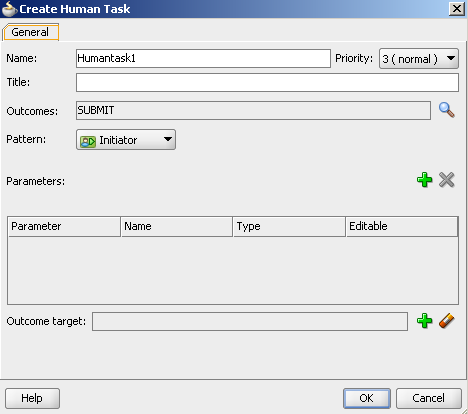
Right click on Enter Application Form user task, select Properties > Implementation



Click the green plus sign button to add a new Human Task.

In the pop-up window:

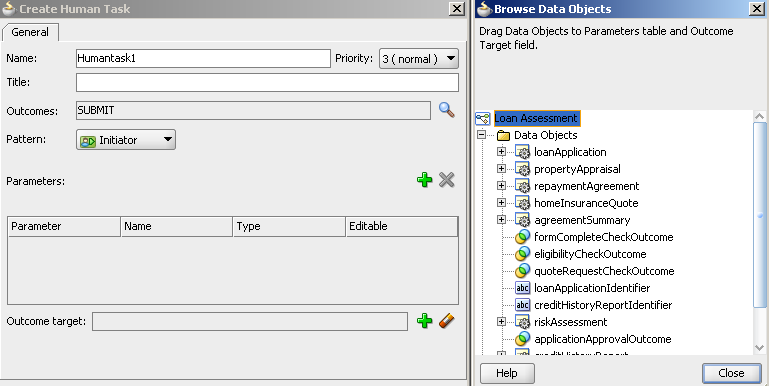
* Enter name and title of the task, leave
* Leave priority as default
* Select an outcome. Here select SUBMIT since this is the first form that user will submit.
* Patterns: select Initiator since this is the first form user will initiate in the process.



**Data Configuration and Mapping**

In this activity, we will add create data variables for task and then assign (map) the process variables to task variables and vice versa. This has to be done because task can only access task variables not process variables. This is the only way of data exchange between different tasks in a process. It is common for a task to assign process variables to its variables before its processing and then assign its task variables back to the process variables after its processing.

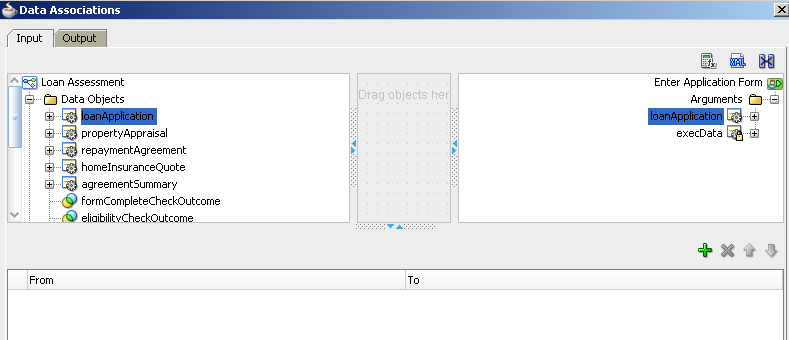
In Create Human Task window, click the green plus sign button in Parameters section which will open a Browse Data Objects window. It contains all process data objects (process variables).



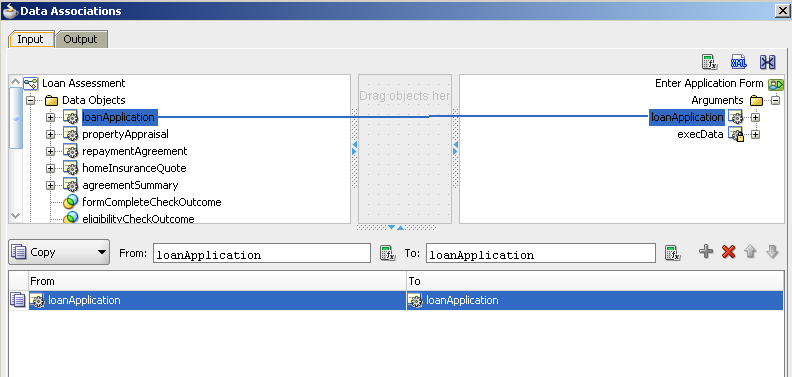
Drag “loanApplication” variable onto the Parameters table, select Editable.

Select OK in Create Human Task window and back to Properties window.

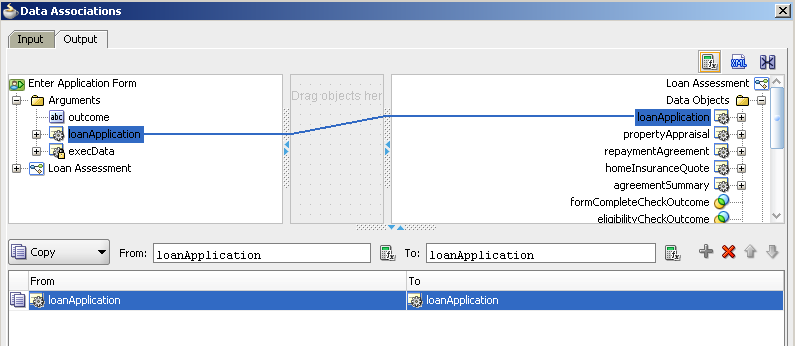
Click Data Association to open Data Association window. There are Input and Output mapping tab. In the Input tab, the process variables are on the left-handed pane and the task variables are on the right-handed pane, and vice versa in the Output tab.



To map (mean assigning) from a process variable to task variable, drag a process variable and drop it on the corresponding task variable. A line will be drawn connecting two variables to indicate the mapping, as demonstrated in Figure XX.



Perform similar mapping in the Output table.



Then select OK button in the Data Association window to finish data mapping.

Select OK in the Properties window to finish human task implementation.

### Assess Credit History

### Appraise Property

### Assess Eligibility

### Prepare and Send Acceptance Pack

### Prepare and Send Home Insurance Quote

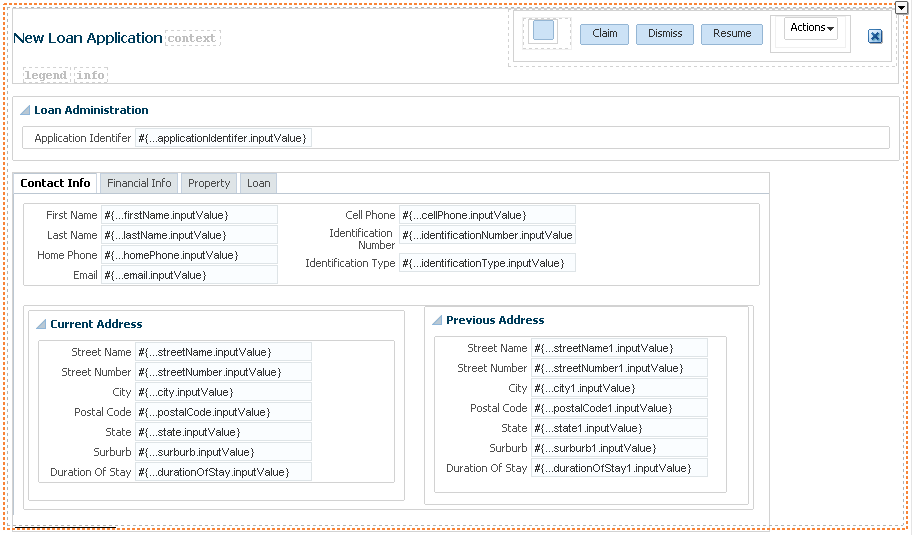
### Verify Repayment Agreement

### Take Final Decision

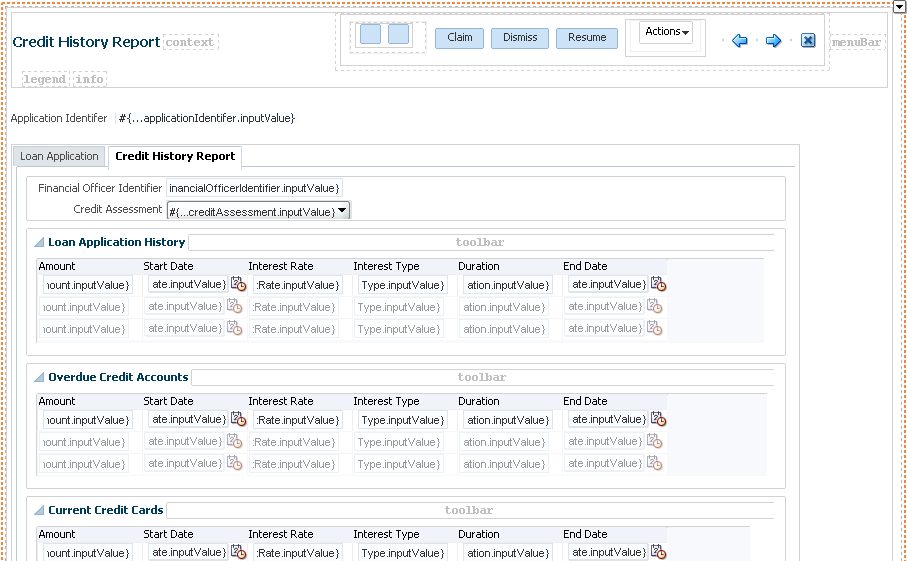
## Implement User Interfaces

In this activity, we are going to implement user interface for every user task. Oracle BPM uses Oracle ADF which is a complete and huge framework for user interface.

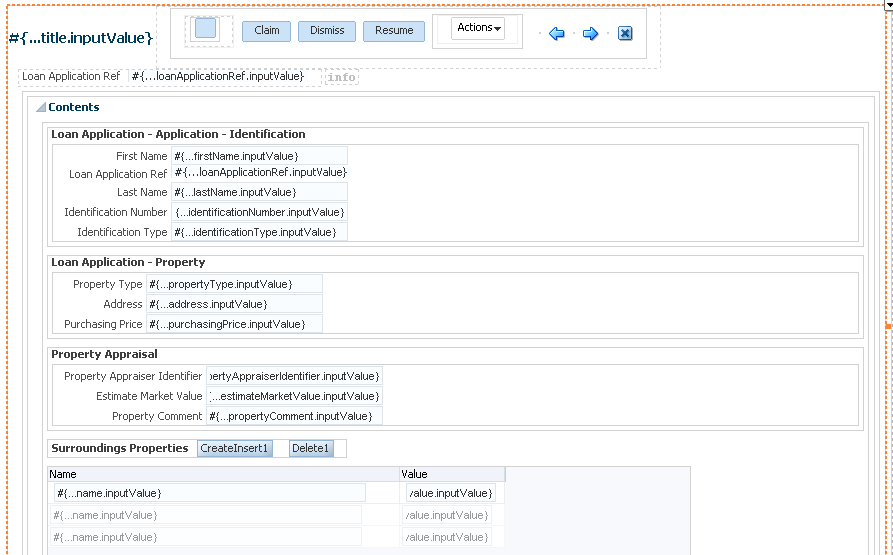
### Enter New Loan Application

Asse

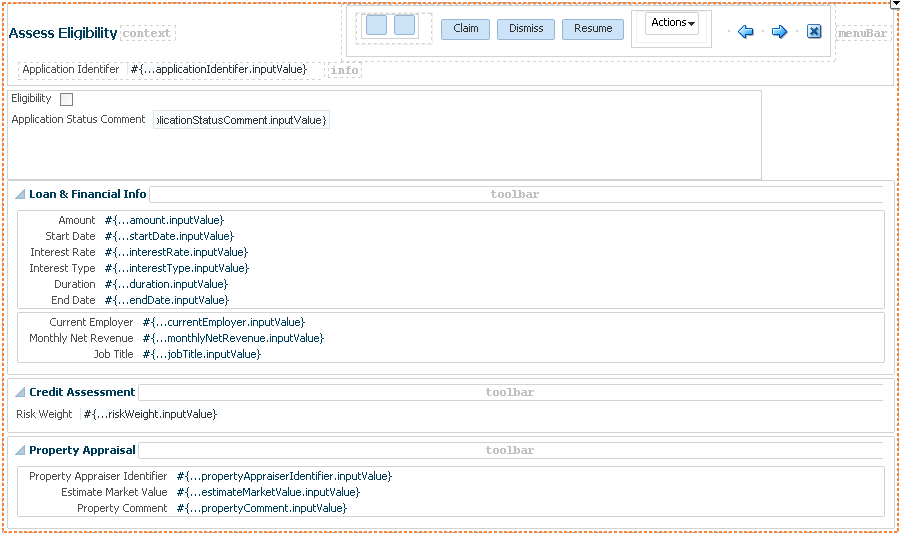
### Assess Credit History



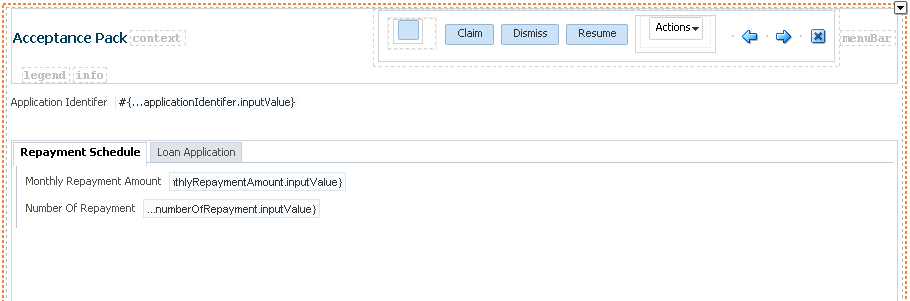
### Appraise Property



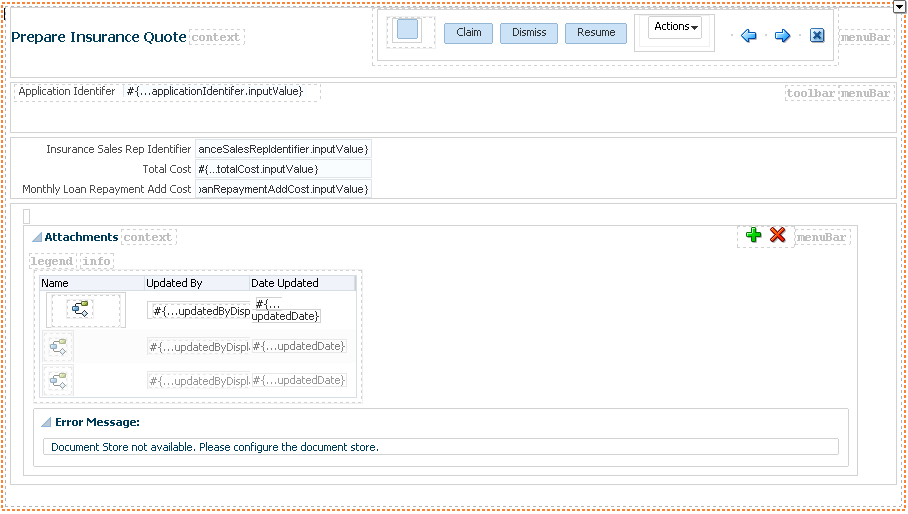
### Assess Eligibility



### Prepare Acceptance Pack



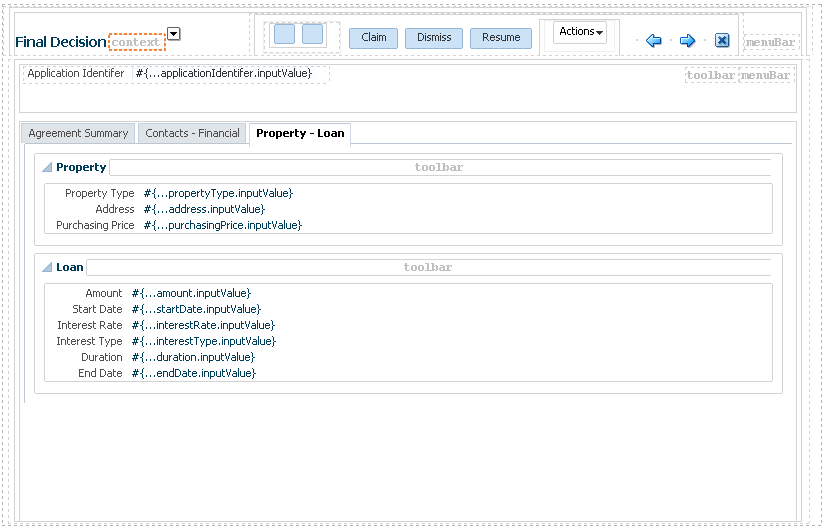
### Prepare Insurance Quote



### Verify Repayment Agreement



### Take Final Decision



## Implement Service Tasks

### Check Application Form Completeness

### Update Rejected/Eligible status

### Update Approval/Rejected loan status

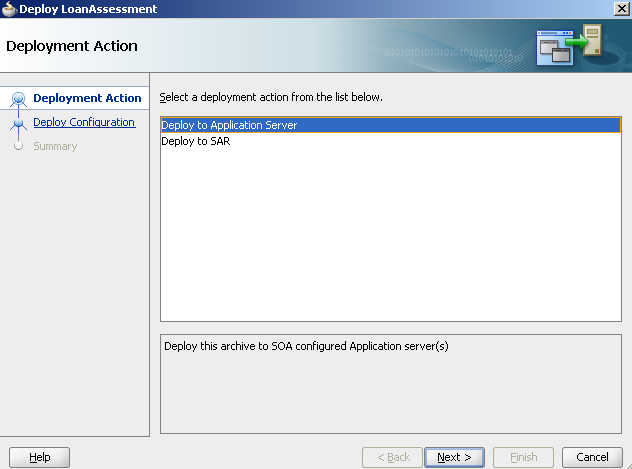
## Implement Business Rules

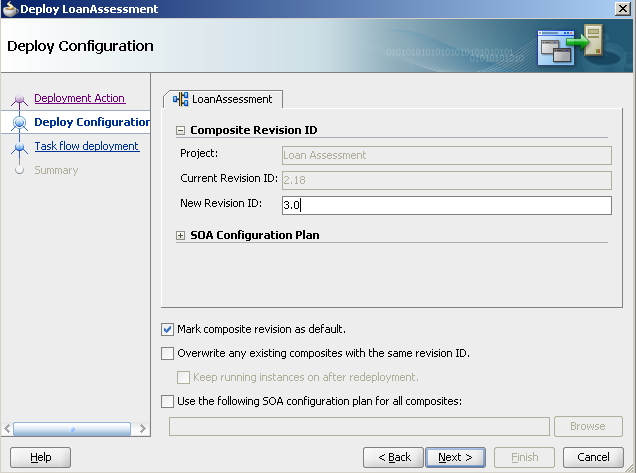
### Assess Loan Risk

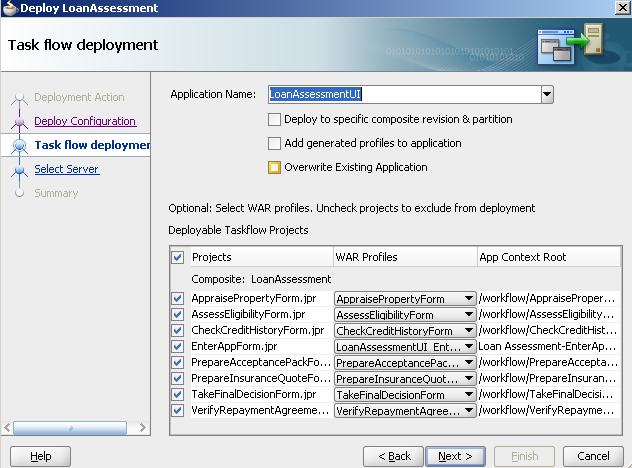
## Implement Timer

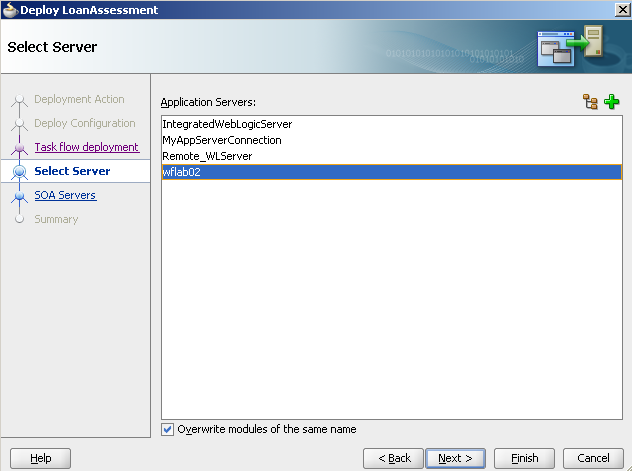
### Repayment Agreement Timer

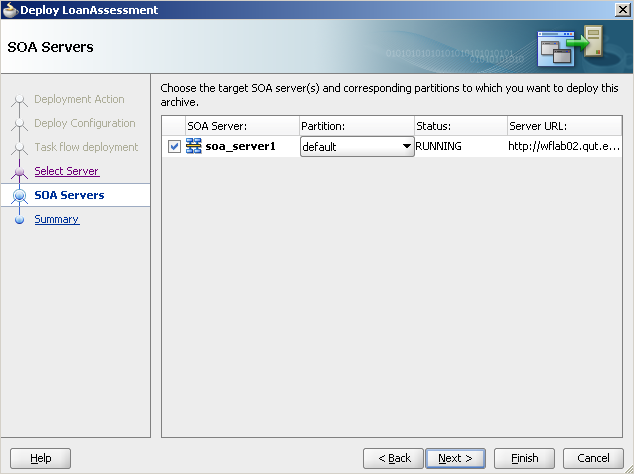
# Deploy the Process

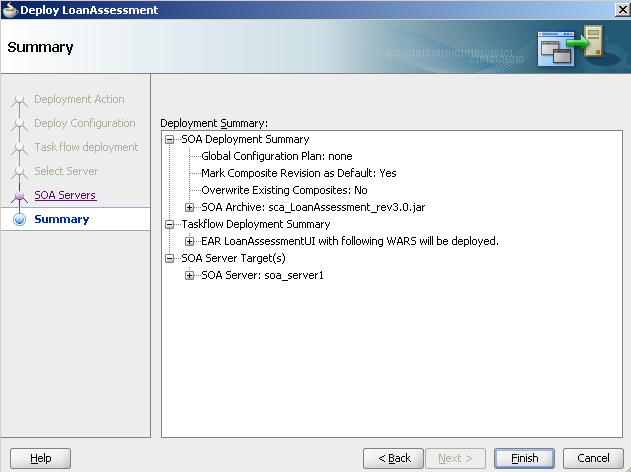




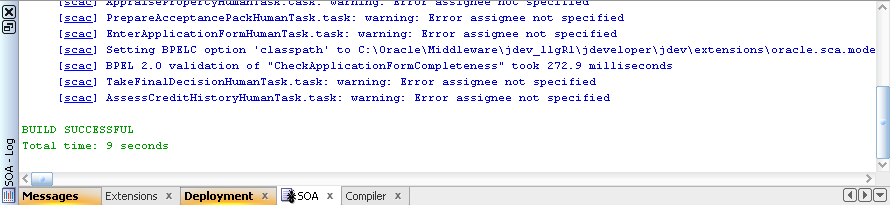


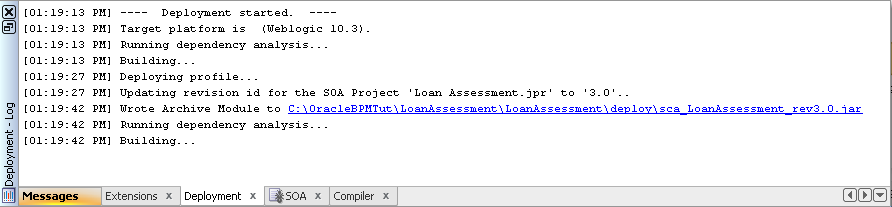


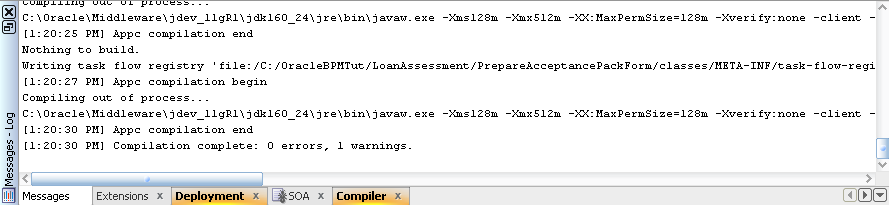


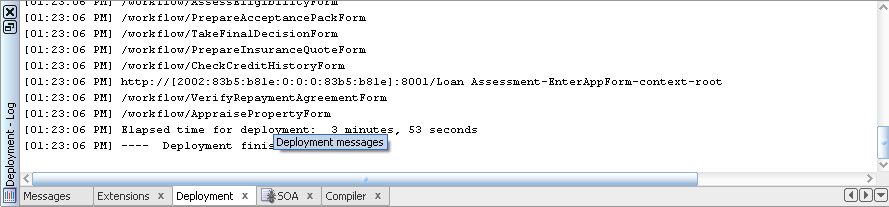


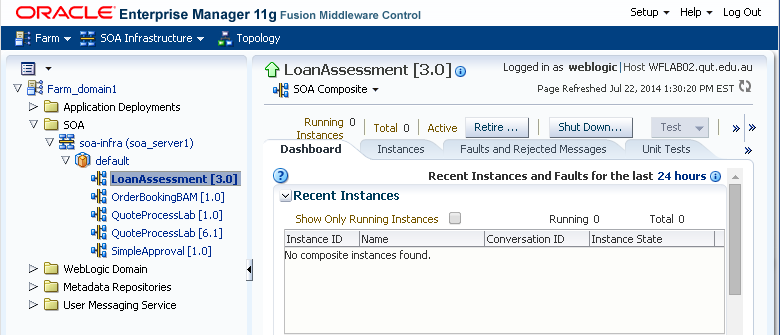


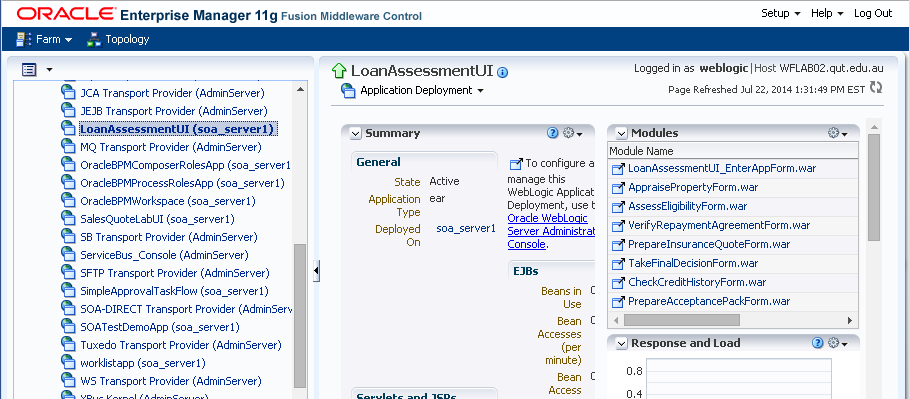






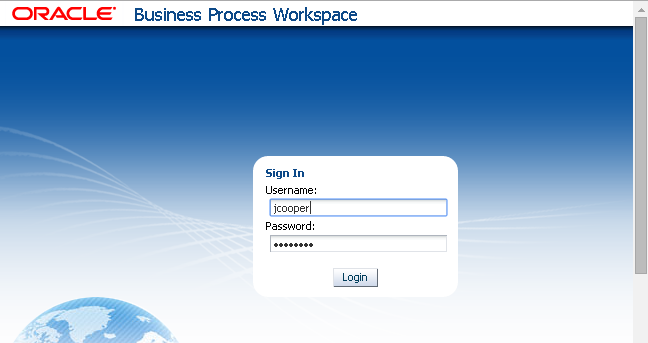


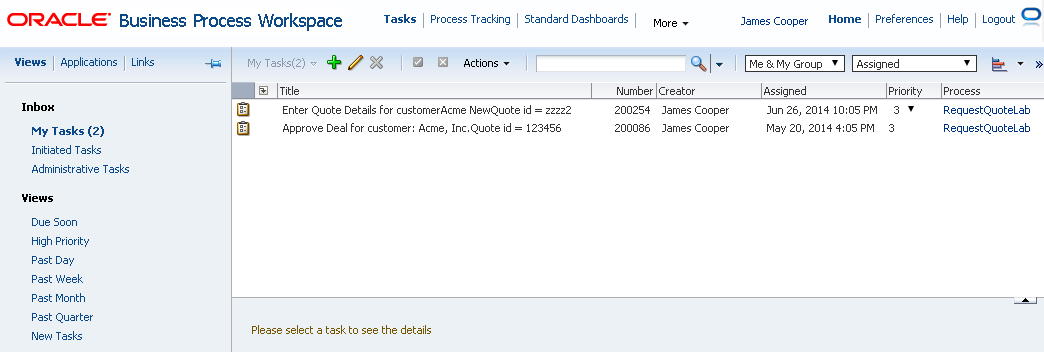


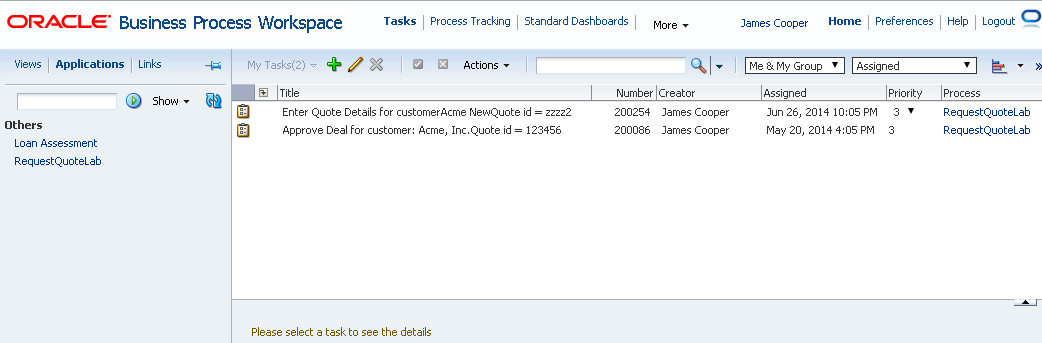


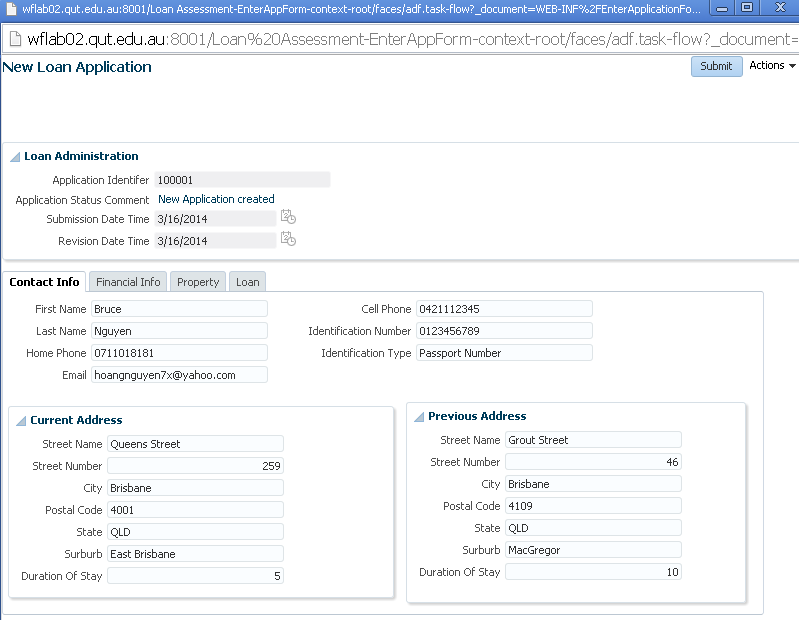
# Run the process

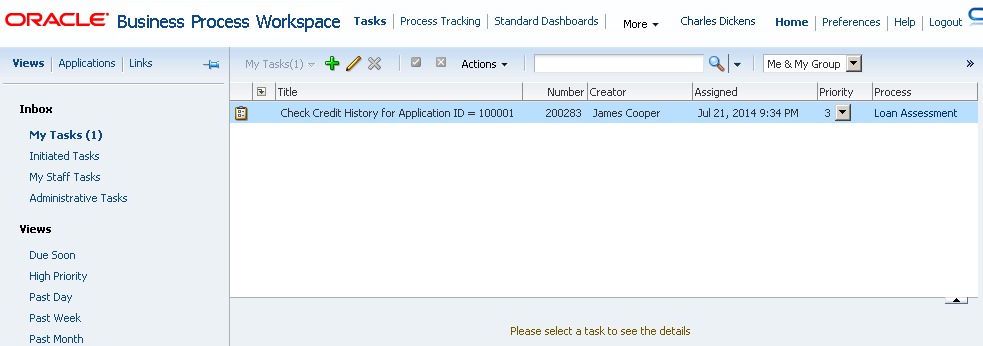
## Process user

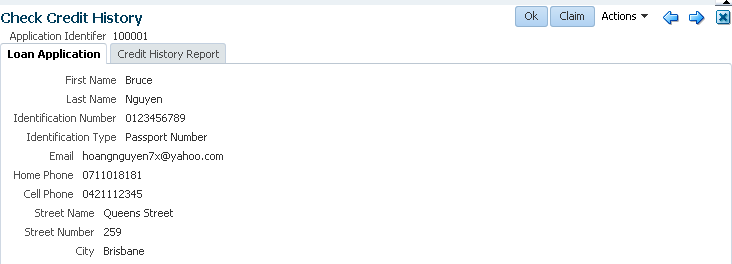


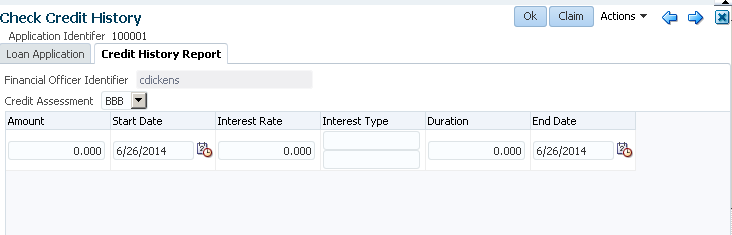


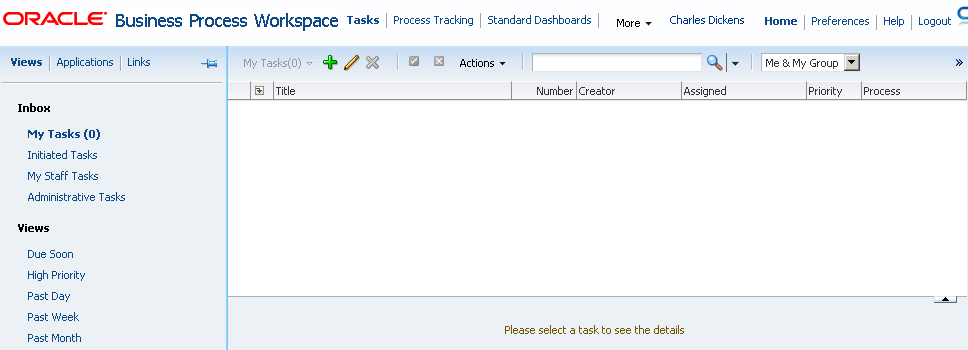


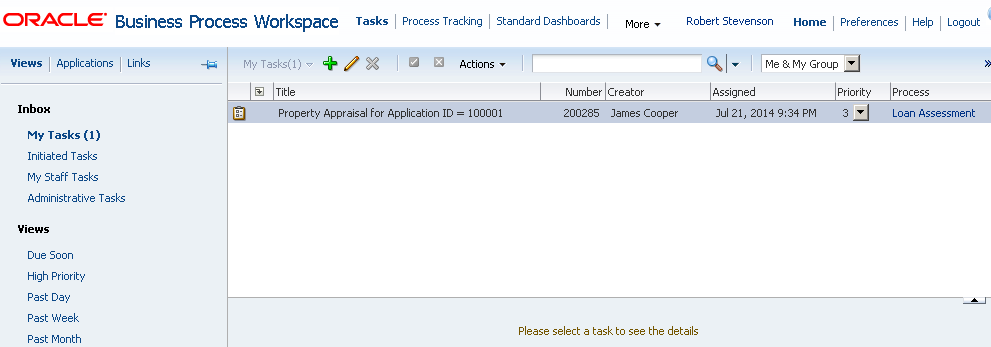




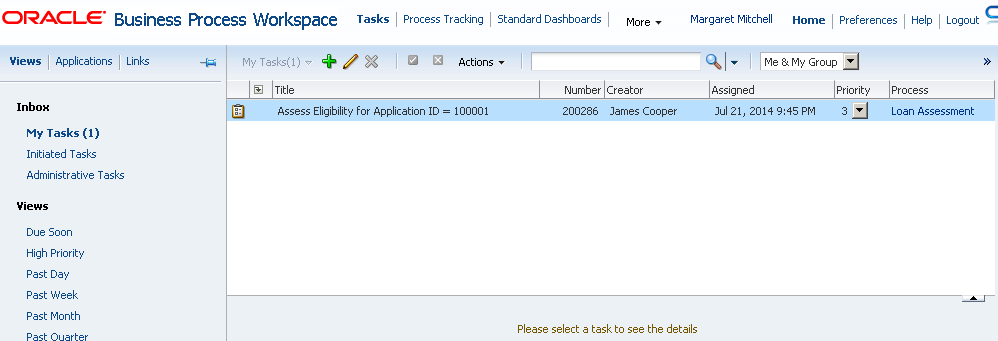


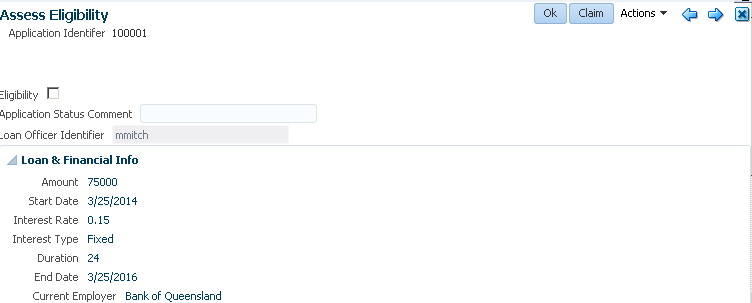


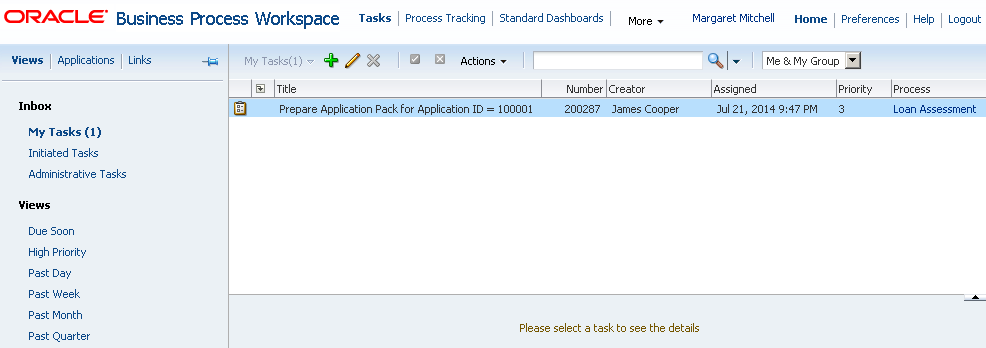


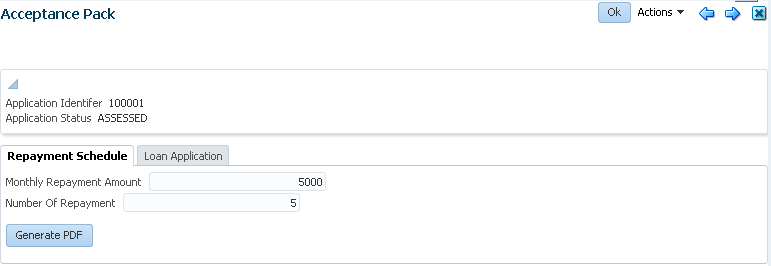


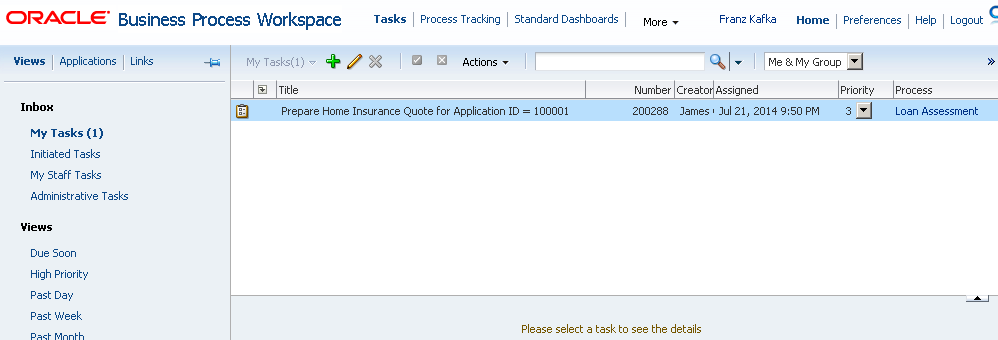


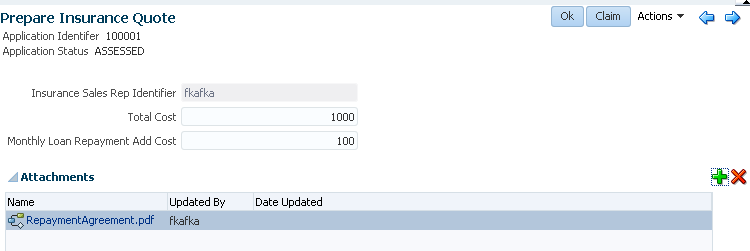


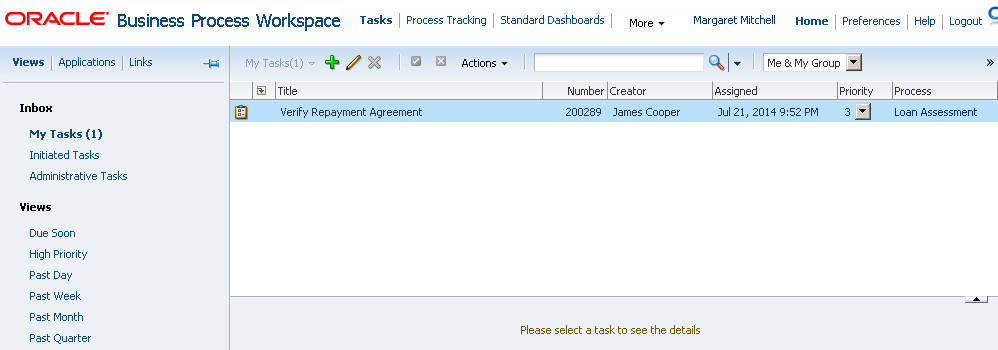


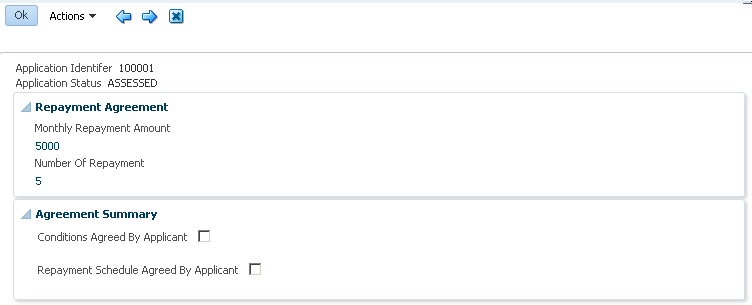


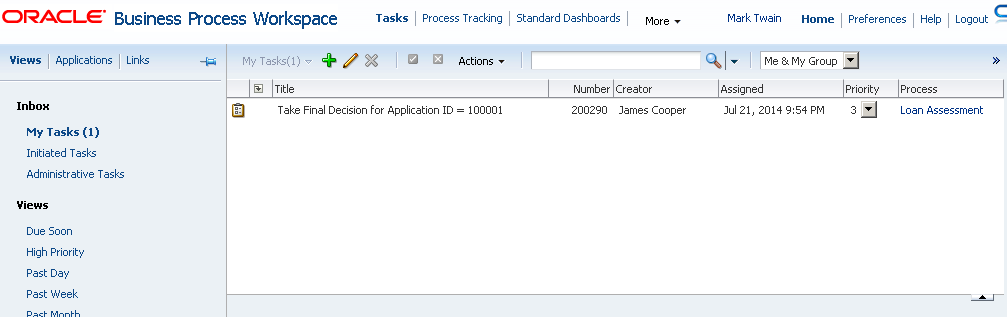


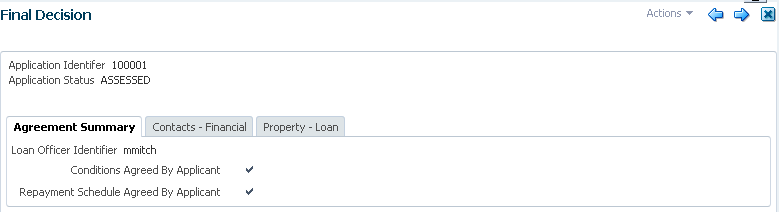


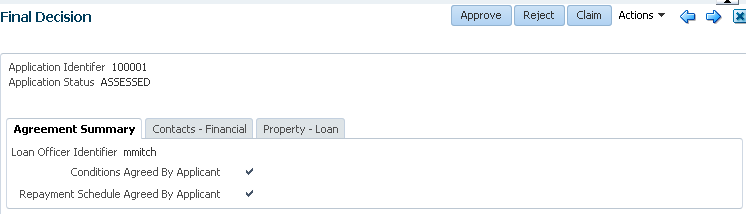




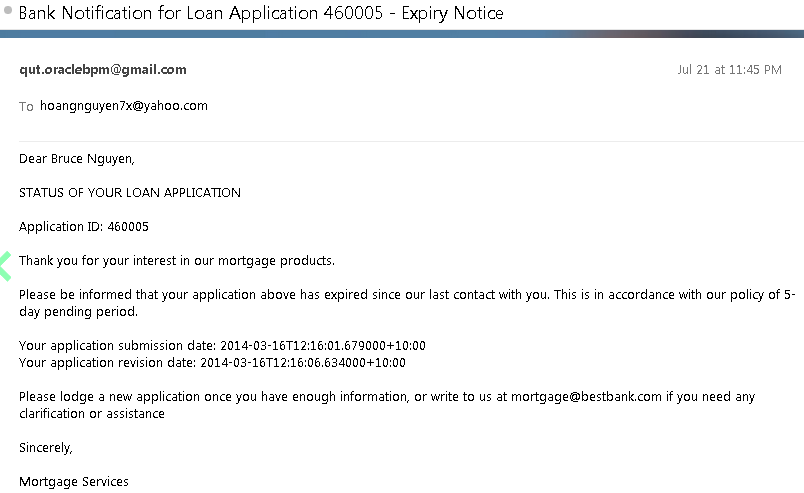


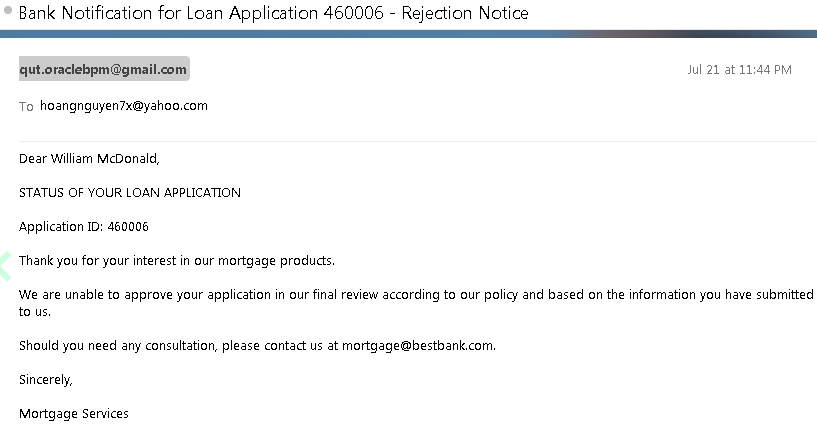


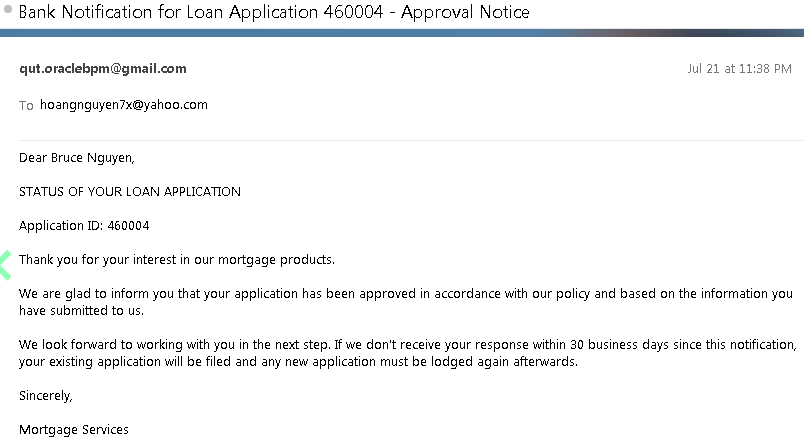






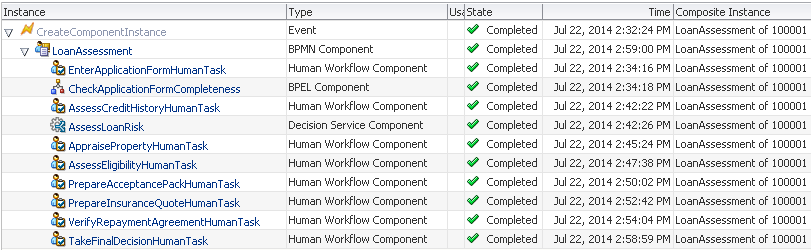


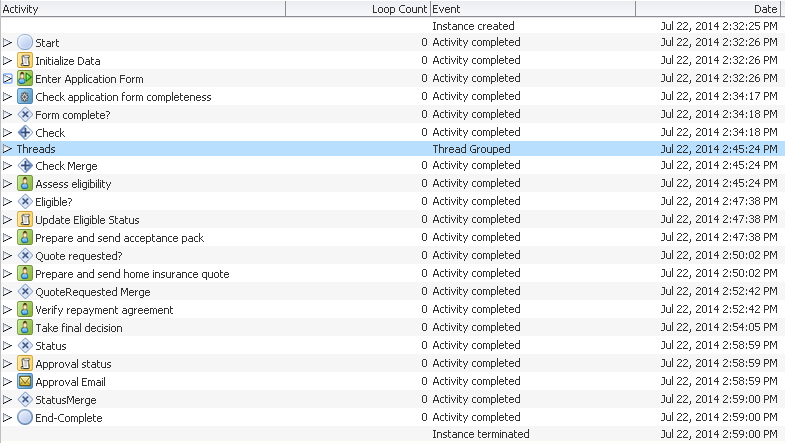


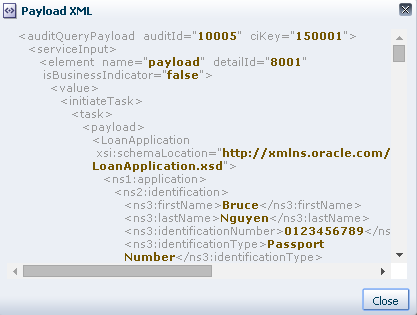




## Administrator







# Appendix

## Further Improvements

* One single UI project for all Human Task Forms
* Form Validation
* Collaboration event message
* Connect with BAM
* Database communication
* Use page template and view template to design reusable interface

## Common Errors

Below is a collection of common errors that people new to Oracle BPM would likely encounter. Refer to this list to save you huge time and headache of implementing process in Oracle BPM environment, which is a multi-layered, large scale and very complicated system.

|  | **Error** | **Resolution** |
| --- | --- | --- |
|  | User form appears with no fields for data entry although data objects have been assigned to human task and ADF form. | Cause: if process data object is compound, meaning they contain another data object as elements, then Oracle BPM cannot initialize compound object, even though “auto initialize” is selected in its properties.  Solution: either define simple data object only, i.e. only contains primitive fields (string, integer, etc), or initialize compound data object by using Script task and XML literal. |
|  | A process/project data object has been defined but not appear in human tasks or other places for reference. | Make sure that “Save” button is pressed straight after data object is added. Then, check again the data object has been added in the list of process data objects. Then, check if data object is shown in the reference place. |
|  | An error message relating to data transfer between tasks: e.g. subLanguageEvent or Exception | It is likely caused by invalid data schema. Whenever change any business object, all dependent artifacts need to be updated accordingly: process variables, task variables including variables used in rules, service tasks. For example, a LoanInfo business object is deleted and replaced with a new object called CreditInfo, or even it is replaced with new object with the same name because Oracle BPM has a mechanism to keep track of object version and strict data validation will raise an exception if the object has been replaced. |
|  | In BPM Process Workspace, Homepage > Applications > Error message: “Runtime Operation Error”. This happens after deployment of a new process version. | There is a conflict with a previous process version. Undeploy previous version would resolve this error. |
|  | Compilation error - XML-20108: (Fatal Error) Start of root element expected. in rdf files. | Open rdf file. Add:  <?xml version="1.0" encoding="UTF-8" ?>  <test>  </test>  Save |
|  | Error in running Ant tasks to create seed demo users.  Error message: The name is undefined….. | "Cause: The name is undefined" means the task is not installed in your ant environment. <foreach> is not a task of vanilla ant but needs the ant addon [antcontrib](http://ant-contrib.sourceforge.net/) available for ant. After installing antcontrib you should use <taskdef resource="net/sf/antcontrib/antlib.xml"/> to activate all antcontrib tasks.  Don't use <taskdef resource="net/sf/antcontrib/antcontrib.properties"/>as mentioned on <http://ant-contrib.sourceforge.net/> as net/sf/antcontrib/antcontrib.properties contains only tasks for ant versions before Ant 1.6.x  Download ant\_contrib library (.jar file) from <http://ant-contrib.sourceforge.net/> and save to ant/lib directory. |
|  | [12:04:19 PM] Weblogic Server Exception: weblogic.application.ModuleException:  [12:04:19 PM] Caused by: java.lang.ClassNotFoundException: oracle.bpel.services.datacontrol.types.Number  [12:04:19 PM] See server logs or server console for more details.  [12:04:19 PM] weblogic.application.ModuleException:  [12:04:19 PM] Taskflow deployment failed to deploy to server. Remote deployment failed  [12:04:19 PM] #### Deployment incomplete. ####  [12:04:19 PM] Remote deployment failed (oracle.jdevimpl.deploy.common.Jsr88RemoteDeployer)  com.sun.faces.config.ConfigurationException: CONFIGURATION FAILED! oracle.bpel.services.datacontrol.types.Number  It happened to UI projects, not the process project. | Login to BPM Workspace > Administration  Assign sufficient members to the roles without any members.  Add Approval Group, assign member to the group.  Redeploy the project. |
|  | When logging into BPM Workspace with a user to start a process, there is no corresponding process under “Application” section. | Log in BPM Workspace > Administration with administrator role. Go to Roles. Check the Role to start the first activity of the process is there. One possible error is it is not created during deployment process. Create the role manually, add users.  Log in with the starting user again. |
| 1. 1 | No credential mapper entry found for password indirection user=sys for data source quote. | Go to your Application Properties (in JDev) -> Deployment and uncheck the option "Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment", rebuild and deploy the application.  Explain: Its a WLS feature (Password Indirection). While generating ear file for an application from JDev, it will generate a **-jdbc.xml file for each DB connection in the application resources, set the indirect password attribute, update weblogic-application.xml to add each**-jdbc.xml file as a module and update web.xml (if it exists) to add a resource reference to each jdbc jndi name. However, since there is no server to deploy to, Jdev will not place the passwords in the ear file. The EAR file will not deploy as is. The passwords for the data sources must be setup on the server before the application will run correctly.  Check out [this link](http://download.oracle.com/docs/cd/E12839_01/webcenter.1111/e10148/jpsdg_deploy.htm#BGBEHJAJ) to get some more info and steps to achieve this (http://download.oracle.com/docs/cd/E12839\_01/webcenter.1111/e10148/jpsdg\_deploy.htm#BGBEHJAJ) |
| 1. 2 | Error when running application | One cause maybe the ADF Framework is not used properly. |
| 1. 3 | Performance wio lib | Update to WIN64 Lib |
| 1. 4 | Error when re-deploying UI components for process forms | We must undeploy the componen on the server before deploying it again. Cannot override in this case |
| 1. 5 | When deploying: warning: Error assignee not specified |  |
|  | Error file XSD while compiling the project | Open the Business Object property, select option to view all XML structure. There will be more detailed messages to locate the error. |
|  | Weblogic Server Exception: weblogic.application.ModuleException: Context path '/workflow/EnterApplicationForm' is already in use by the module: /workflow/EnterApplicationForm application: LoanAssessmentUI | Must undeploy on server before deploying again with override option |
|  | Server Error: XPath processing error | Check to make sure the initialized data through XML Literal or XPath Expression is well-formed and qualified. Should check with an XML Validation Tool like XML Spy rather than manual check (very error-prone). |
|  | oracle.bpm.bpmn.engine.model.runtime.microinstructions.TrappableException: faultName: {{http://schemas.oracle.com/bpel/extension}subLanguageExecutionFault} messageType: {{http://schemas.oracle.com/bpel/extension}RuntimeFaultMessage} cause: {faultName: {{http://schemas.oracle.com/bpel/extension}subLanguageExecutionFault} messageType: {{http://schemas.oracle.com/bpel/extension}RuntimeFaultMessage} cause: {XPath expression failed to execute. An error occurs while processing the XPath expression; the expression is bpmn:getDataOutput('loanApplication'). The XPath expression failed to execute; the reason was: ORABPEL-77005 Uninitialized data element. DataOutput loanApplication is not initialized in flow element Enter Application Form. Make sure to initialize DataOutput loanApplication before using it in flow element Enter Application Form. Contact oracle support to resolve the issue. . Check the detailed root cause described in the exception message text and verify that the XPath query is correct. } } | No data input is created for Loan Application (Input flow failed) |
|  | Displayed form has blank data. The form’s labels and non-editable elements are shown but there is no editable elements with data. | It shows that data initialization failed due to some reason during the human task or form creation.  One fix is: remove the current human task of the form, remove the associated ADF form. Re-create a new human task and then a new ADF form. |
|  | Displayed form is blank: no editable and non-editable elements are displayed, no data. Just a completely white blank page. | There is an error with the server: shut down and restart the SOA server. |
|  | BPEL Exception: Mismatch Assign. cannot set a nonmessage value to a message-based variable. An attempt to assign a nonmessage value to a message-based variable failed. Verify the BPEL source for invalid assign activities. ORABPEL-09225.  Log file might reports:  Model.xml file not found  Model.xml file not found for component XX  Model.xml file was not generated for the component  If component is BPEL 2.0, Model.xml file is not needed and thus does not get generated and this error is expected | This is due to the wrong setVariable method.  For example:  setVariableData("outputVariable", "payload", "/client:processResponse/client:result", "complete");  The outputVariable is a message type variable, but if we call setVariableData("outputVariable", "complete"), this means set it to a non-message value.  Another possible error is the Initialize dialog of the variable property. Can accidentally click on the wrong node. May have to modify directly in the BPEL XML file to remove the initialization.  Another note is setVariableData would raise exception if the assigned data was not initialized. Initialization can be done via assigning any value from any value subnode of inputVariable to the concerned subnode of outputVariable. |
|  | BEA-000000> <<.> Error while setting task display, this can happen with app loading issue, trying to load for 5> | Redeploy the task form project associated with a human task to only Admin and SOA server, not to other servers (BAM, OSB). This error appears when these servers are not running. |
|  | WARNING: Structure is not serialized for: CheckCreditHistoryForm\_CheckCreditHistoryHumanTask | Review the data variable in the payload, check if they exist in the Business Objects under Business Catalog or have any errors?  Check the XSD files in the businessCatalog/BusinessObjects and XSD folder.  In some cases, to fix this error we have to remove and re-create the business object. |
|  | Approve, Reject buttons are not available though they are the selected actions in the human task | Check the Human Task to make sure APPROVE and REJECT are among task actions.  Refresh the Data Control to make sure the actions from Human Task consistently transferred to Data Control (right click, select Edit Definition, Re-create….., and then select Refresh in the Data Control palette. We have to re-deploy the UI form project to server after this change. |
|  | Getting error while deployment of the above saying  The XML-Schema file for the fact com.oracle.xmlns.bpm.bpmobject.hellomodule.reviewobject.ObjectFactory could not be found in the composite. The XML-Schema file xsd/ReviewObject.xsd could not be found in the composite. The XML fact com.oracle.xmlns.bpm.bpmobject.hellomodule.reviewobject.ObjectFactory of target namespace assumes the existence of the schema file in the composite. Check the composite for the schema xsd/ReviewObject.xsd and make sure it has target namespace. The schema is expected in the project xsd or businesscatalog folder. If the error persists, contact Oracle Support Services. | Need to delete the Business Object and re-create it, repeat the business rule creation, this may fix the error. |
|  | Login BPM Workspace and error message: “Runtime Operation Error”  ORABPEL-30017  Invalid task definition. The task definition at default/PrjTaskReminderNotification!1.0\*soa\_5cb0ae06-9485-488f-91b1-1b2872276fb9/Initiator could not be read. The task definition is associated with workflow default/PrjTaskReminderNotification!1.0\*soa\_5cb0ae06-9485-488f-91b1-1b2872276fb9/Initiator. Make sure that the task definition is available at the specified URL and that it is a valid XML document.  If it does not work, the error message in server log is:  Could not locate composite for workflow component XXX  Ensure composite has been successfully deployed, and that the SOA server has completed loading composites | This happens when there are more than one active record in BPM\_CUBE\_PROCESS table. You just have to make sure that there is only one active record per process.  Execute the following query by connecting to the SOA\_INFRA schema of the database  select processId, processName, compositeName, revision, status, from BPM\_CUBE\_PROCESS;  If it returns multiple records for the same ProcessName, make the status of all those entries to -1,commit and redeploy the process. Now, you can see the process in the workspace without any problem    If the above still does not work, we might need to re-deploy the same process with the version number as indicated in the error message. Redeploy and run the process again. The status of the process now should be 1 and it should run as normal. |
|  | Form display does not have the expected data object elements | Check Data Control, under the corresponding data control, Task > Payload, the data object has been there. If it is not been there (likely), check if it has been added to data configuration of the Human Task associated with the activity. Add if it is not there. If it has been there, refresh data control to make sure it is added to the data control. |
|  | The approval and reject action of the form: form does not disappear after pressing these buttons. When reopening the form, the approva/reject button have already been inactive. |  |
|  | Cannot connect BPM MDS to the Server  Caused by: oracle.bpel.services.workflow.client.WorkflowServiceClientException: javax.naming.CommunicationException [Root exception is java.net.ConnectException: t3://2002:83b5:b81e:0:0:0:83b5:b81e:8001: Destination unreachable; nested exception is:  java.net.SocketException: Permission denied: connect; No available router to destination]  at oracle.bpel.services.workflow.client.WorkflowServiceClientContext.createInitialContext(WorkflowServiceClientContext.java:686)  at oracle.bpel.services.workflow.client.WorkflowServiceClientContext.getJNDIInitialContext(WorkflowServiceClientContext.java:341)  at oracle.bpm.client.BPMServiceClientContext.getJNDIInitialContext(BPMServiceClientContext.java:228)  at oracle.bpm.client.impl.BPMServiceRemoteClient.getHistoryService(BPMServiceRemoteClient.java:339)  ... 14 more  Caused by: javax.naming.CommunicationException [Root exception is java.net.ConnectException: t3://2002:83b5:b81e:0:0:0:83b5:b81e:8001: Destination unreachable; nested exception is:  java.net.SocketException: Permission denied: connect; No available router to destination]  at weblogic.jndi.internal.ExceptionTranslator.toNamingException(ExceptionTranslator.java:40)  at weblogic.jndi.WLInitialContextFactoryDelegate.toNamingException(WLInitialContextFactoryDelegate.java:788)  at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext(WLInitialContextFactoryDelegate.java:366)  at weblogic.jndi.Environment.getContext(Environment.java:315)  at weblogic.jndi.Environment.getContext(Environment.java:285)  at weblogic.jndi.WLInitialContextFactory.getInitialContext(WLInitialContextFactory.java:117)  at javax.naming.spi.NamingManager.getInitialContext(NamingManager.java:667)  at javax.naming.InitialContext.getDefaultInitCtx(InitialContext.java:288)  at javax.naming.InitialContext.init(InitialContext.java:223)  at javax.naming.InitialContext.<init>(InitialContext.java:197)  at oracle.bpel.services.workflow.client.WorkflowServiceClientContext.createInitialContext(WorkflowServiceClientContext.java:682)  ... 17 more  javax.mail.MessagingException: connect failed;  nested exception is:  java.net.SocketException: Permission denied: connect  at com.sun.mail.pop3.POP3Store.protocolConnect(POP3Store.java:161)  at javax.mail.Service.connect(Service.java:288)  at javax.mail.Service.connect(Service.java:169)  at com.myapp.MailboxConnection.connect(MailboxConnection.java:66)  caused by: java.net.SocketException: Permission denied: connect | When trying to connect to the WebLogic Server Administration Server from WLST using localhost as the host name, the following message may be displayed if the listen-address attribute of the Administration Server has been restricted to certain IP addresses:  javax.naming.CommunicationException [Root exception is  java.net.ConnectException : <t3://HOST:PORT> : Destination unreachable;  nested exception is: java.net.ConnectException: Connection refused; No  available router to destination  **Workaround**  Use either of the following workarounds:   * Check that the listen-address attribute of the Administration Server has been set correctly in the domain configuration file. You can either remove the listen-address line or simply comment it out. Oracle recommends that you comment it out in case you need to know the value at a later time. For example, in the domain configuration file: * <server> * <name>AdminServer</name> * <ssl> * . * . * . * </ssl> * <machine>*machine\_name*</machine> * <!-- listen-address>*machine\_ip\_address*</listen-address --> * </server> * Use the host name of the Administration Server, instead of localhost, in the WLST connect command.   Ok, it doesn't look like a security manager issue. It looks like something in the operating system on that machine is preventing your application from connecting to that host. Try the tips in the JavaMail FAQ for debugging connection problems  Problem solved by uninstalling Norton AntiVirus. |
|  | java.net.MalformedURLException: Unsupported protocol: t3  or  BPM MDS cannot establish connection with the server | Add wlclient.jar and wljmxclient.jar to the server java classpath to process t3 protocol. |
|  | Look at the diagnostic log:  oracle.adf.controller.activity.ActivityLogicException: ADFC-06014: An exception occured when invoking a task flow finalizer  ADF\_FACES-30200:Fatal exception during PhaseId: RESTORE\_VIEW 1. The UIViewRoot is null, this is usually caused by previous exceptions, for more complete debugging information turn the logging level to fine. | “The UIViewRoot is null, this is usually caused by previous exceptions” |
|  | ADFC-12000: State ID in request is invalid for the current session  ADF\_FACES-60096:Server Exception during PPR, #3 | Review this link: <http://one-size-doesnt-fit-all.blogspot.in/2011/10/pageflowscope-with-unbounded-task-flows.html>  **How does ADF technically solve identifying the separate tabs**  **“…**Side note: Behind the scenes the server is smart enough to check the session parameters against the previous known connection/session to stop intruders impersonating another user's session ... you can test this by intercepting the next request before it goes out and changing the \_adf.ctrl-state parameter before it hits the server. ADF will complain displaying the following error message "ADFC-12000: State ID in request is invalid for the current session."  **For this error: open BPM/Workspace in another browser window, maybe clear the cache of current window, and conduct the transaction in the new window.** |
|  | There was an error deploying the composite on soa\_server1: Deployment Failed: Error in getting XML input stream: oramds:/deployed-composites/default/LoanAssessment\_rev5.0/SCA-INF/classes/xsd/InitializationData - FULL.xml: Illegal character in path at index 96: oramds:/deployed-composites/default/LoanAssessment\_rev5.0/SCA-INF/classes/xsd/InitializationData - FULL.xml. | Reason: InitializationData - FULL.xml filename contains spaces. This is invalid for MDS upload. This error will not occur if this project is not yet linked to MDS.  Fix: rename file without spaces.  In addition, Oracle BPM might automatically use previous settings/files if the new deployment reuses a previous revision number, which makes this error still occur. This happens even there is no such XML file in deployed .jar file (*deploy subfolder of project folder*) or server cache (*user\_projects\domains\domain1\servers\soa\_server1\dc*). In order to avoid this error, new deployment must use an unused revision number or a number without having this error before. |
|  | Process fault occurs without any user error message. If viewing process instance in Enterprise Manager console, the instance shows a system fault at Java embedding activity (RuntimeFault). | This may be caused by java codes in Java Embedding activity.  One debugging method is insert addAuditTrailEntry() in codes to investigate sources of error. |
|  | Cannot create MDS connection to server. Test connection always return “Test Failed”. | One reason is firewall on local computer or remote server. |
|  | When refreshing Data Control in BPM Studio, there is warning: “Structure is not serialized for <data control>” | There is an error with the payload associated with the human task in terms of schema validation.  There is a tricky error: Oracle BPM only allows one array element business object. If more than one array are defined in the business object, it would not be able to be parsed successfully. |
|  | In the Project schema files list pop up from selecting data for a human task, Oracle BPM displays both obsolete and current schema files, i.e. there are duplicate items in the list, only one of them is current to be valid for selection, other may have parsing error. | Notice to scroll down the list to select the right one. |
|  | the exception reported is: java.lang.RuntimeException: failed to compile execlets of | This is a compilation error of Java Embedding code. One issue may be the missing “import” statement for one of required classes used in the code.  Open BPEL process, switch to Source view, add “<import location="<java path to package and class>" importType="http://schemas.oracle.com/bpel/extension/java" />” to the XML source.  This applies to even native Java classes, e.g. java.util.regex.Pattern. |