IBM BPM – Loan assessment process lab

This tutorial refers to the “loan assessment” process example used in Chap. 9 of M. Dumas, M. La Rosa, J. Mendling, H.A. Reijers, *Fundamentals of Business Process Management*, Springer 2013. This example can be downloaded from http://fundamentals-of-bpm.org/wp-content/uploads/LoanAssessmentProcessDescription.zip.

The loan assessment process (also called mortgage application process in this document) is a mixture of modern and archaic. Loan applications can be completed either at branch offices or over the Web. However, once applications are received, they are printed out and assigned to inboxes.

Processing these loan applications is a highly variable, non-standard process with little to no visibility on status of any particular loan, nor how the loan company is performing overall on its key metrics and targets. Once a loan is approved, the actual creation of the customer and loan information in the systems of record is a “swivel-chair” action, requiring Loan Officers to enter the loan information in a legacy application or database manually.

In this lab, we will design an automated process in IBM BPM that will provide better automation, process consistency, and visibility. You will see how the process will be created by reusing some existing assets to achieve quicker time to delivery. You will see how the process is designed in multiple iterations, adding first the process steps, then user screens, then a decision point and branches, and at each iteration the process is “played back” to show progress to that point. This is a key benefit of IBM® Business Process Manager (BPM), that the solution can be developed layer by layer, with each iteration in the development process able to be shared with the rest of the team to show progress, gain concurrence and refine requirements as the solution is built.

Let’s get started!

2.1 IBM BPM Introduction

IBM supports a broad range of BPM use cases, but the predominant use case for our current customers has been “Rapid composition and proactive management of process-based applications”. Most of our customers have selected IBM BPM to create new, flexible applications that orchestrate specific critical business processes that span work group boundaries and multiple back-office systems.

IBM BPM is a complete platform for composing model-driven process-oriented applications, including the following capabilities:

- Full-function run-time environment with specialized engines for application execution and monitoring.
- Graphical programming model based on BPMN, where activities are mapped to “services”. Most services are defined graphically (microflows), with access to scripting (Javascript, XML/XSLT) and Java APIs if needed.
- Built-in support for complex data-object definition, persistence, searching, and transformation.
- Built-in support for defining adapters to external applications or web services.
- Built-in support for exposing process applications as web services.
- Built-in support for event-driven execution.
- Run-time engines execute within J2EE app server tier, with clustering for high scalability and availability.

IBM BPM is unique in that its model-driven architecture is based on a single shared model of the process. The BPMN flow diagram, the underlying physical implementation details, the in-flight process state, and the historical performance data are all aggregated together in the same process model. Consequently, the “round-tripping” between design, implementation, and analysis views of the process across its lifecycle is straightforward – with a single shared model there is no translation or information-loss like you see in competitive suites that have multiple independent tools with multiple underlying process data representations.

Authoring Environment:

The IBM BPM Process Designer is a visual process definition tool. The Process Designer is an application that allows Business Analysts and process designers to visually specify discrete process activities, assign them to various participant roles, and implement the rules, events, and split/merges that govern the flow between those activities.

The Process Designer is a simple, Visio-like, drag-and drop application for building Business Process Diagrams (BPDs). A Business Analyst can build these diagrams rapidly because they are in swim-lane format, the most common format for process definition. IBM BPM was the first commercial vendor to implement BPDM, the Business Process Definition Metamodel. BPDM is the official "serialization format" for BPMN, and was ratified by the Object Management Group (OMG) Architecture Board in 2007. IBM is a member of the OMG committee responsible for defining the Business Process Definition Metamodel (BPDM). The alignment and synchronization of the BPMN and BPDM standards is ensured,

since they are driven by the same organisation. The standard format ensures maximum portability of your existing process diagrams into and from tools like Visio, CaseWise, ProForma, IDS Sheer, etc.

IBM BPM imports, exports, and exchanges process models using BPDM. IBM BPM can also import dialects of BPEL, the standard language for process orchestration and automation of web-service execution.

IBM BPM's architecture is 100% data-driven. No coding is required to define behavior. Most components can be built graphically or with textual wizards. IBM BPM includes property sheets and graphical wizards for the business process diagram (BPD), activities, forms, integrations, rules, reports, events, etc. Javascript is optional for the development of user interfaces, business logic, or reusable script components. Generally, lower level coding is not required in IBM BPM, but minor .Net or Java coding may be needed for exposing external system APIs for IBM BPM consumption.

IBM BPM does not require any proprietary languages or scripting. Graphical modeling is fully BPMN Compliant, including the constructs for events. Web based forms are built in a WYSIWYG "Coach Designer" (the modeled form definitions are represented internally as XML). If scripting is needed, IBM BPM utilises standard Javascript. Standards are a core focus of IBM BPM, from being 100% J2EE to providing a unified Authoring Environment in Eclipse to all the underlying representation of components.

IBM BPM relies on 100% standard, proven RDBMS systems to host its process and performance databases. This ensures that process data can be easily consumed by non-IBM BPM tools, report writers, and data warehouses.

IBM BPM is a full-function BPMS platform specifically designed to support the development and execution of “end-to-end processes” that may be long-lived, and may span multiple sub-processes, applications, user groups, or functional organisations. In IBM BPM, sub-processes and activities may be implemented using multiple existing applications or different BPMS tools, but still enables the end-to-end process to be monitored and governed as a single distributed entity. This is in contrast to application-specific “workflow” modules / add-ons that are only designed for automation of the application in which they run.

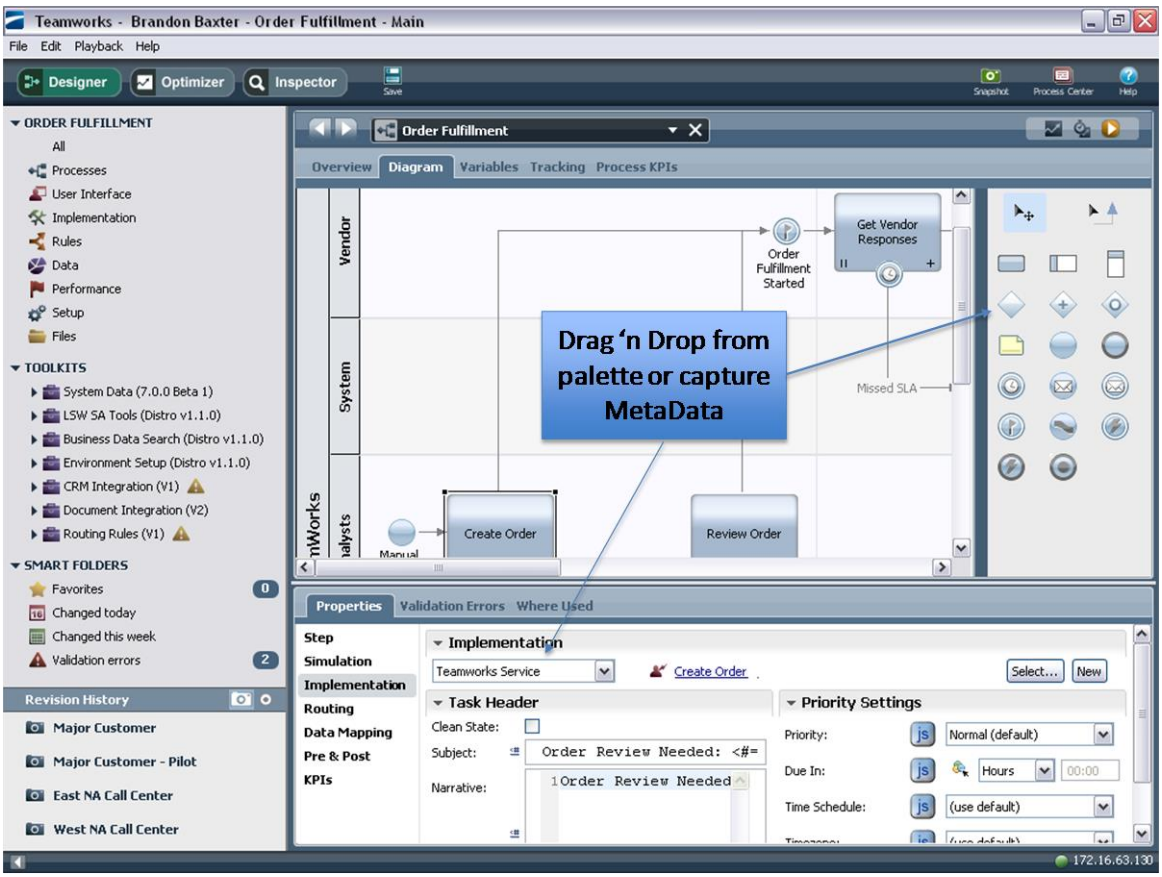


Figure 1. IBM BPM Authoring environment

Performance and KPI Tracking

IBM BPM tracks Key Performance Indicators (KPIs) – i.e., process performance metrics – during process execution. IBM BPM includes several out-of-the-box KPIs that reside in the IBM BPM Library, such as Total Time, Rework, Cost, and others. IBM BPM also enables you to create custom KPIs and add those to the IBM BPM Library for reuse. IBM BPM KPIs can be used for reporting, for Service Level Agreement (SLA) rules that trigger real-time alerts, and for process optimization analyses. KPIs can be tracked for individual activities, as well as for entire processes.

In addition, IBM BPM "Tracking Groups" and "Timing Intervals" provide the tracking context for process metrics and KPIs across process boundaries. These constructs are unique to IBM BPM, are defined explicitly in the IBM BPM process models, and are managed automatically by IBM BPM' patented Performance Server. Timing Intervals are especially powerful, as they map logical time-based KPIs (like "Approval Duration") to multiple underlying process paths, all transparently to the report author. This allows reports to be authored in terms of the logical KPIs (for example, "what is the average Approval Duration?") without requiring the report queries to enumerate all possible paths, or requiring the queries to be rewritten as paths change with the process implementation.

Service Level Agreements (SLAs) are specified as shared, reusable IBM BPM components that can be applied to process activities. Once defined, IBM BPM automatically monitors the SLA every time the associated process activity is executed. The SLA is expressed as an English-like business “rule” that can be easily created and understood by non-technical users of IBM BPM – a syntax-directed sentence editor in the Authoring Environment helps users construct a rule properly. The SLA rule specifies the activity and KPI to monitor, the “trigger” condition, and the triggered operation – sending a notification or invoking an exception process.

SLAs can be monitored using IBM BPM reports in the real-time web ScoreBoards, and in the Optimizer view of the Authoring Environment.

Performance Optimization

The IBM BPM Optimizer has been enhanced to support analysis and optimization across multiple processes. The Optimizer displays “heat maps” that show the magnitude of a specific process KPI (like “Execution Time”, “Wait Time”, “Total Time”, “Rework”, or a custom KPI) overlaid directly on top of IBM BPM process models. The KPI measures are aggregated across multiple processes to avoid sub optimizing for just a single process.

In fact, one can perform analysis and optimization work over different “Scenarios” – a Scenario defines a specific set of processes, filtered by time and/or by business data values (for example, only processes involving customer X).

For each scenario analysis, the Optimizer shows a “Smart Start” summary of which processes and which organisational groups are providing the most to the overall performance – or bottleneck!

For each scenario analysis, the Optimizer displays interactive reports showing more details about the aggregate performance of each component of the process. For example, by clicking on an activity in the heat map view, a report is displayed showing which user executed that activity, how often, and what other processes were those users involved in.

The IBM BPM Optimizer recommends performance improvements based upon historical, in-flight, and simulated data. Those recommendations go well beyond most engines, which simply ask you to reduce an activity time or throw more resources at a step. IBM BPM understands which activities take a consistently long period of time and which have peak bottleneck issues. It also understands activities that loop back based upon a decision, such as an approval step. For those looping steps if they are an issue (a number of rejections or a bottleneck), IBM BPM will recommend automating the step by creating business rules. IBM BPM includes a Guided Optimization wizard that will suggest business rules based upon real data correlation so that you understand when a decision can be automated and when not. Thus the Optimization Wizard might tell you that 200 out of 200 times when an order was "Urgent" and from "Woolworths" that it was always approved. Not only does it recommend that (And others) as rules, it can automatically create those rules, implement them in the model, and then do a comparative analysis of the new process to the old, again using historical data. Thus IBM BPM differentiates by leveraging a greater notion of process, more powerful analysis tools, and real historical metadata to provide more meaningful recommendations to the business analyst.

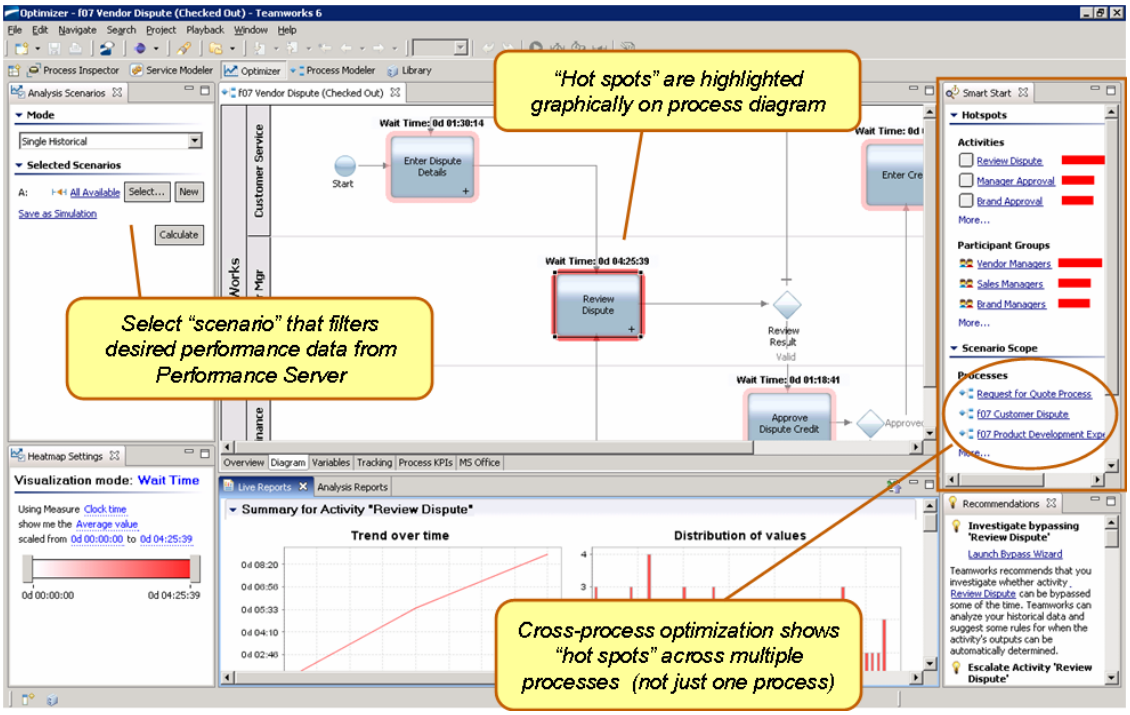


Figure 2. Performance simulation and analysis

2.2 Getting started with Process Designer

2.2.1 Start Process Center

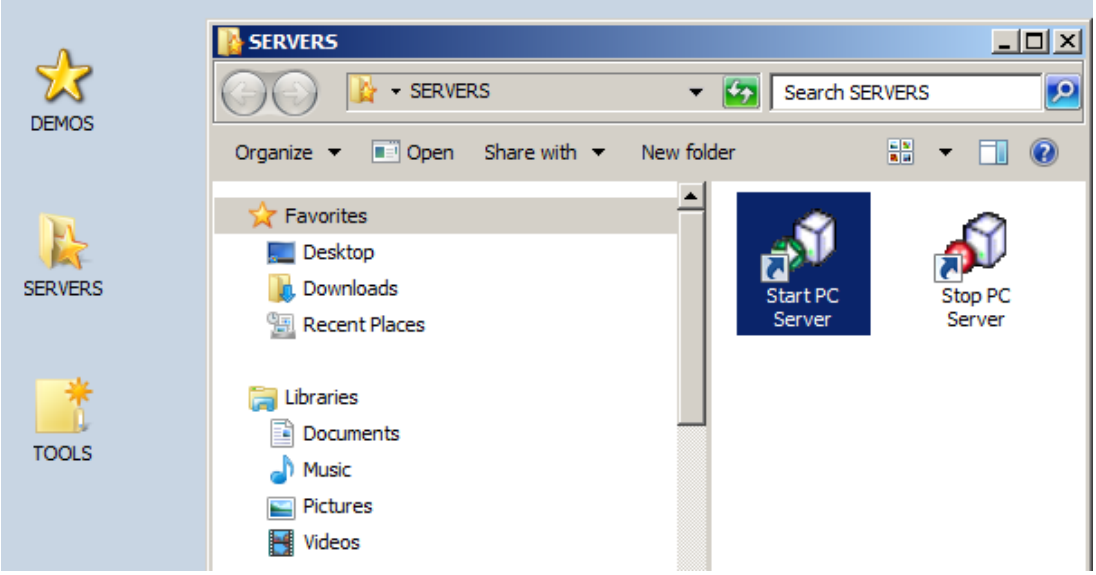
Process Center

Process Center is a component of IBM BPM that provides a shared repository for BPM development, testing, deployment, and overall BPM governance. We will be connecting to Process Center as we design the process to iteratively develop and test our process design.

__1. Locate the **Servers** shortcut folder on your desktop.

__2. Double-click the **Servers** shortcut folder to open it.

__3. Double-click **Start PC Server** as shown.



The server will take approximately five minutes to start. Please wait for the server to start completely (the command window will disappear).

2.2.2 Start Process Designer

Process Designer

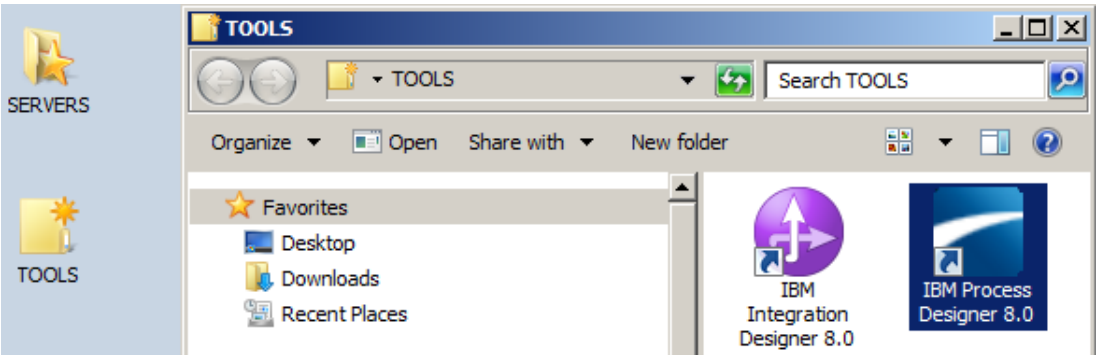
Process Designer is a component of IBM BPM that enables multiple roles to collaborate on process design. The same tool is used for designing the process diagram, user interface screens, the dashboard metrics (KPIs and SLAs), process rules, and integration services.

We will start Process Designer and connect it to Process Center to begin.

__4. Locate the **Tools** shortcut folder on your desktop.

__5. Double-click the **Tools** shortcut folder to open it.

__6. Double-click **IBM Process Designer 8.0** as shown.



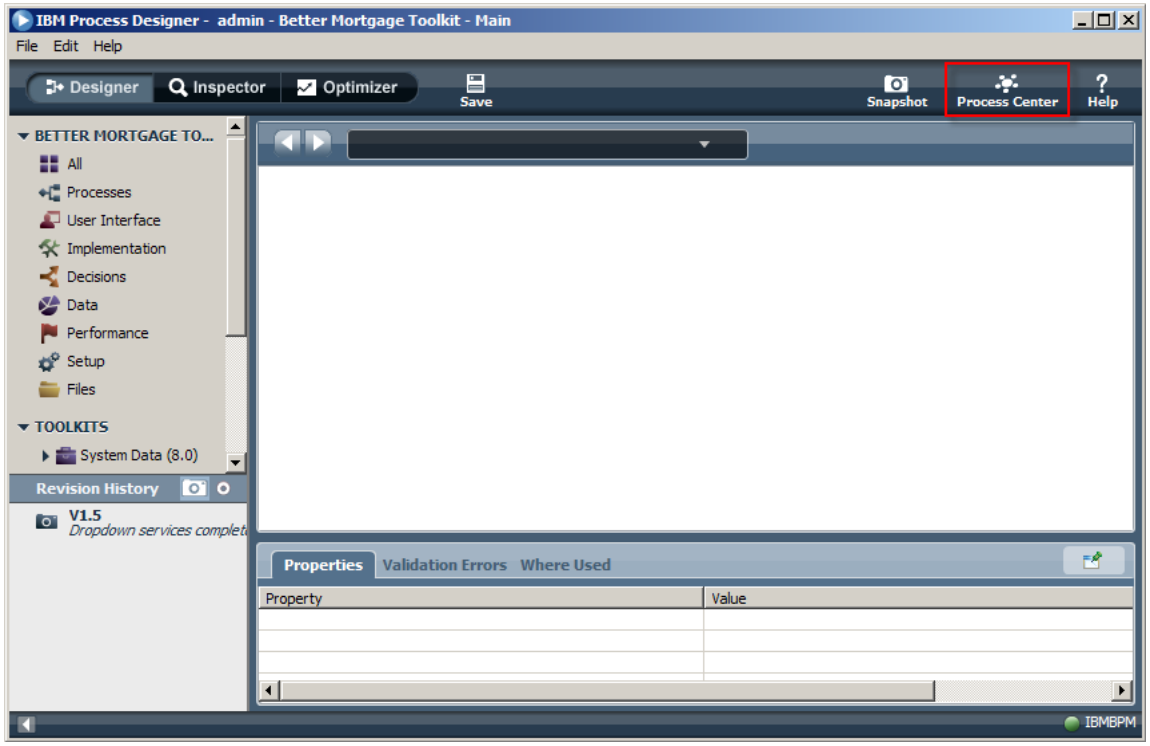
__7. When Process Designer launches, log in with **admin** for User Name and **admin** as shown:

2.3 Creating the process – Version 1

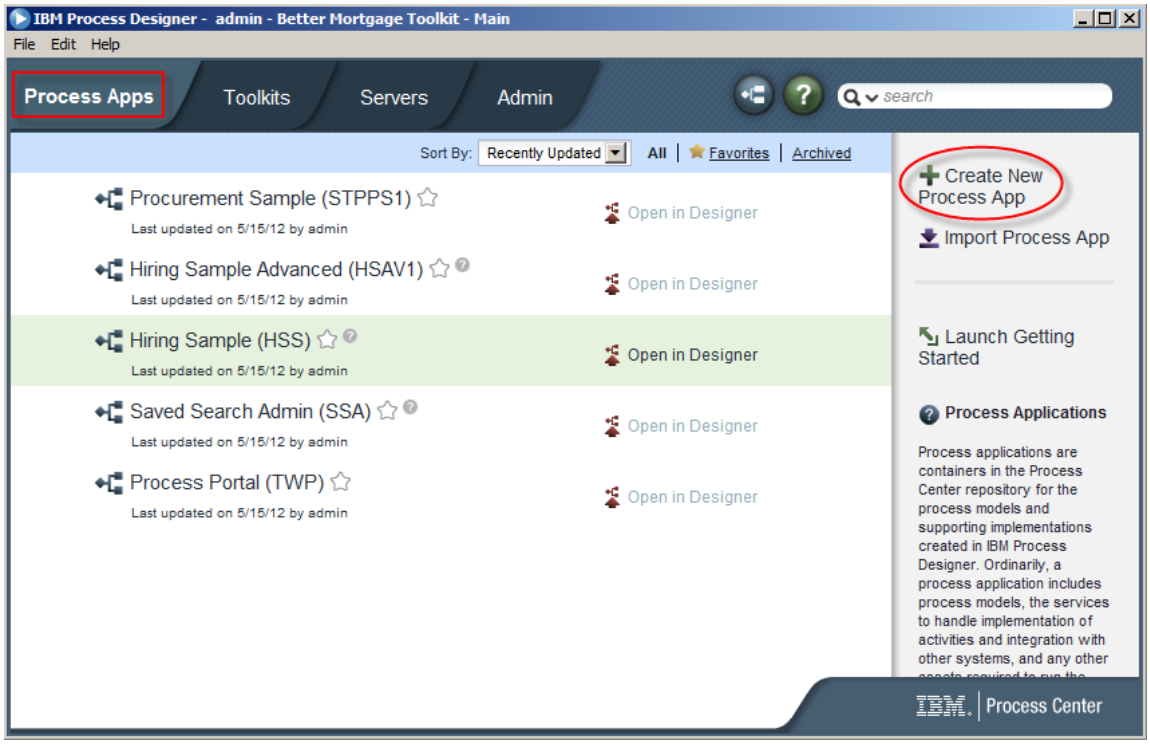
Now we will proceed to create the mortgage application process. We will start with identifying the steps within the process and the roles that will be responsible for completing each step. We will connect the steps from beginning to end. Finally, we will test the results to confirm proper process flow in our first playback.

2.3.1 Create Process Application

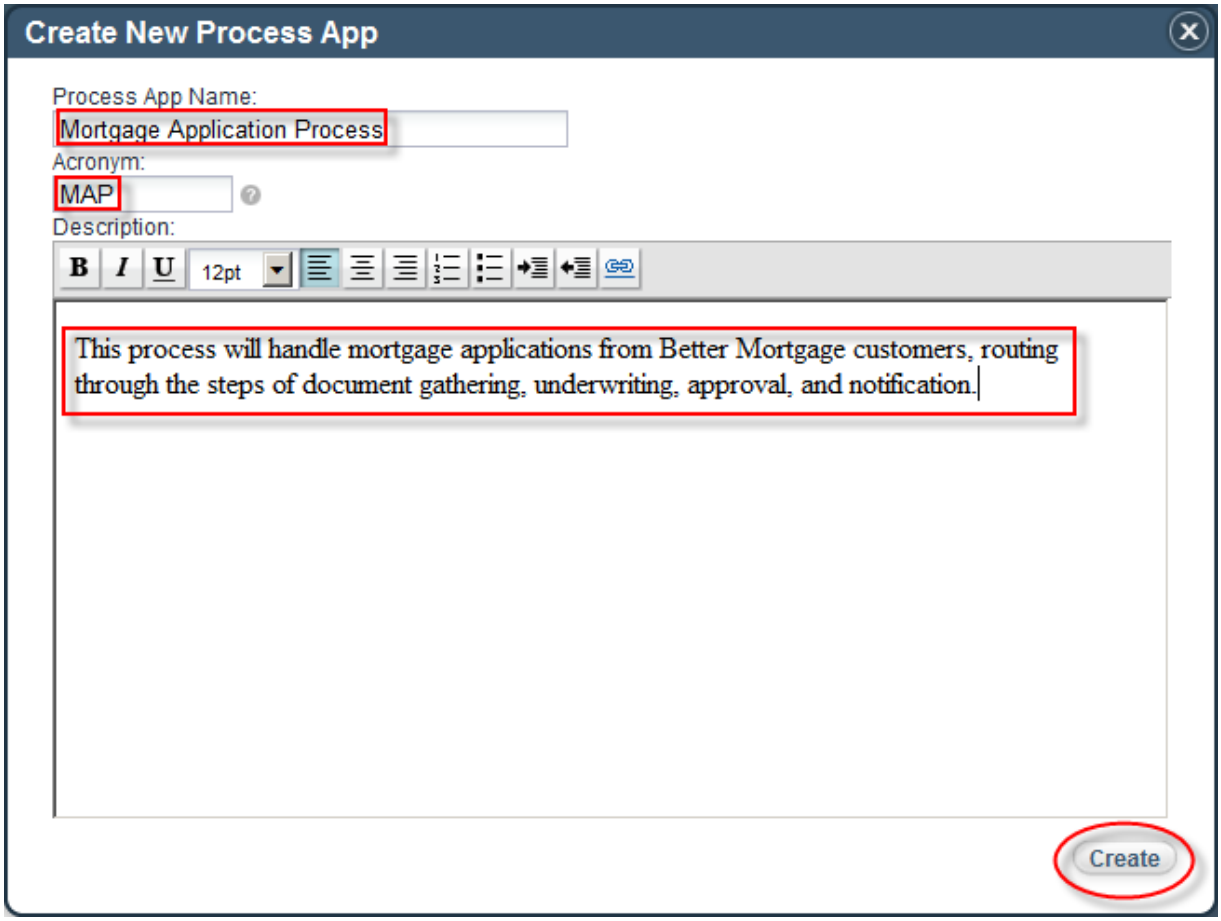
__1. Click the **Process Center** link in Process Designer.



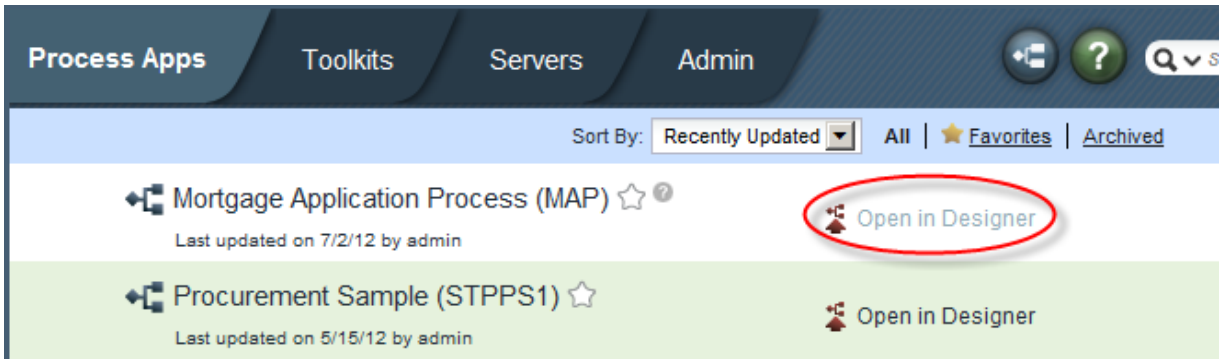
__2. Click **Process Apps** and **Create New Process App**.



__3. In the dialog box, enter the **Process App Name**, **Acronym**, and **Description** as shown:



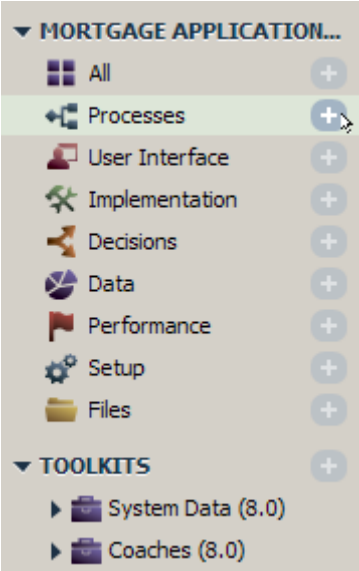
__4. Click **Open** in Designer.



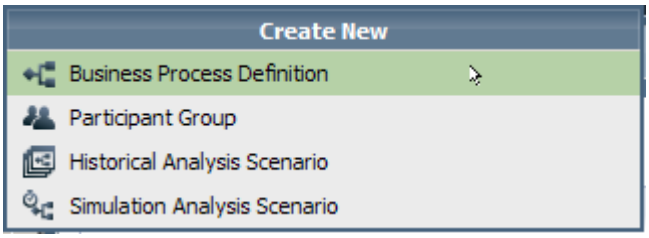
2.3.2 Create Business Process Diagram

__1. Create the Business Process diagram.

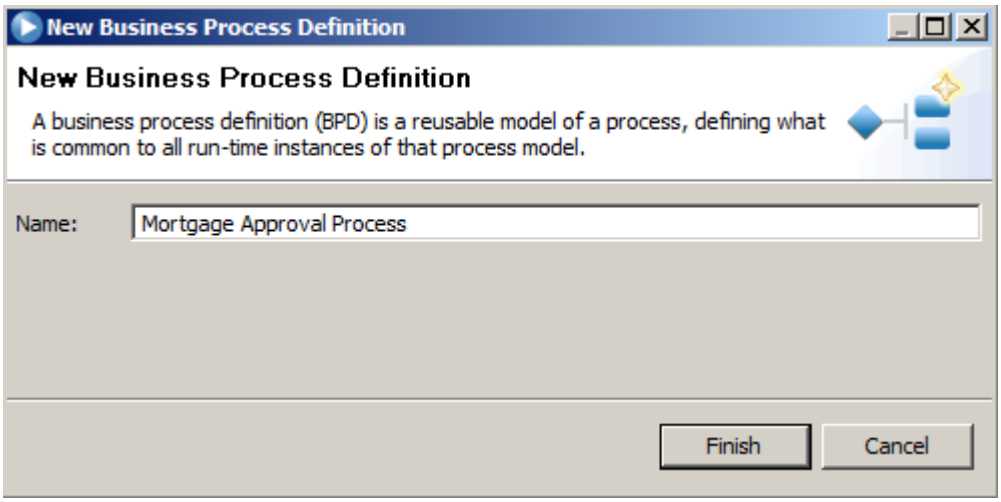
__a. Click the **+** sign next to **Processes**.



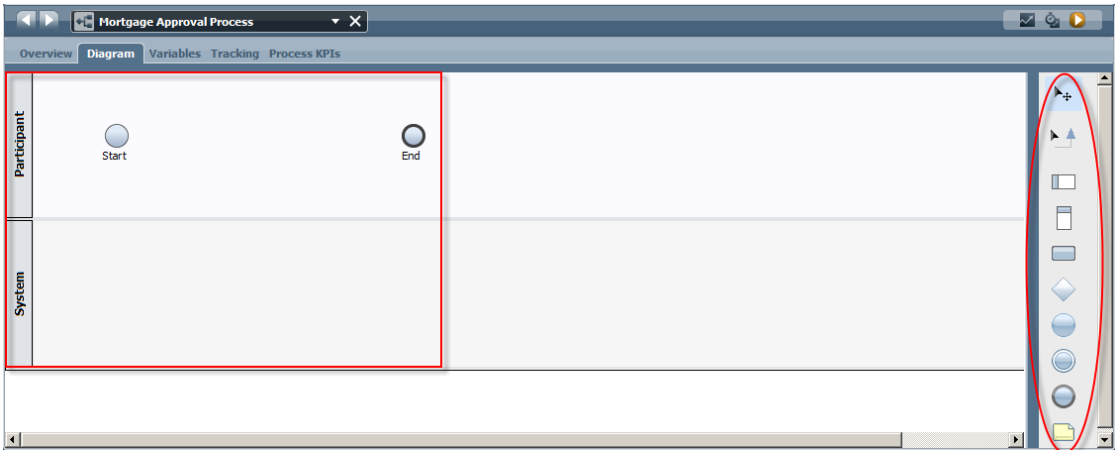
__b. Choose to **Create New Business Process Definition**.



__c. Enter **Mortgage Approval Process** as the name for the business process definition and click **Finish**.

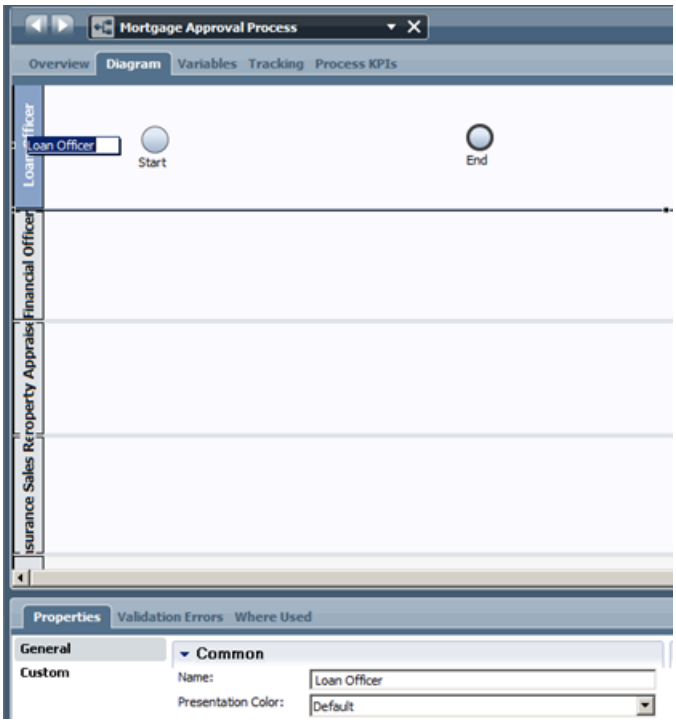


__d. You will be presented with a default business process diagram with two lanes, and a Start and End element as shown in the highlighted box. You will be adding elements to the business process diagram by clicking and dropping elements from the palette shown in the highlighted oval.



__2. Creating process swimlanes

__a. Change the first swimlane from **Participant** to **Loan Officer**. Do this by clicking the grey rectangle on the left side of the swimlane, then either clicking again to change in place or make the change in the properties window below the diagram (see two highlighted ovals). Click **Enter** when you are done with this change.

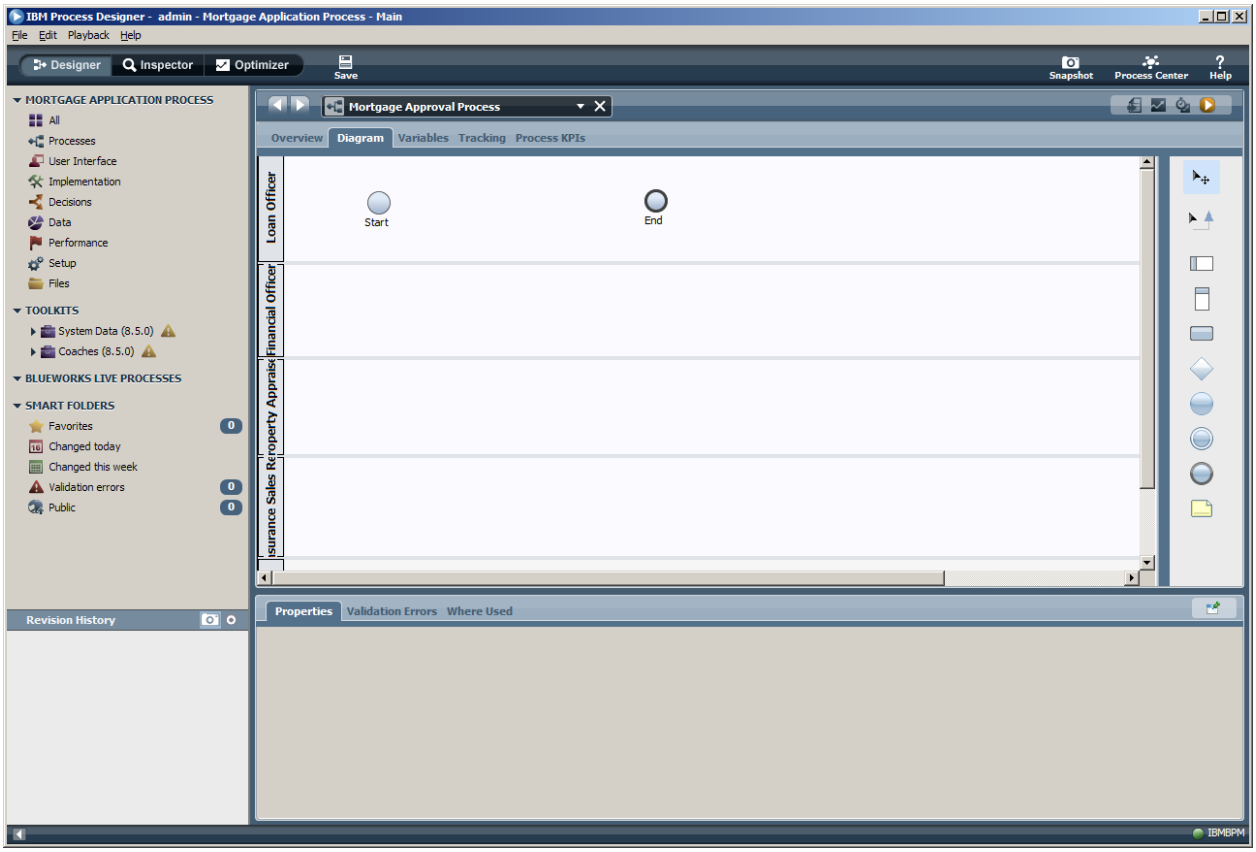


__b. Click the swimlane icon in the palette indicated by the oval and drag it into the diagram similar to the illustration. Change the name of the swimlane to **Loan Officer**.

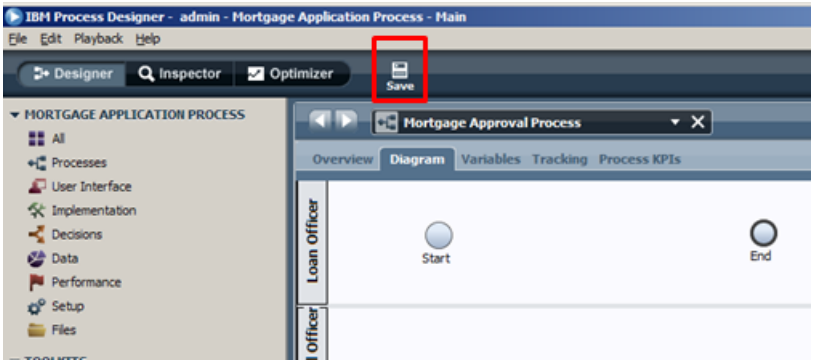
__c. Click and drag three more swimlanes and name them **Financial Officer**, Pro**perty Appraiser** and **Insurance Sales Rep.**

__d. You will now have five swimlanes on the diagram. For readability, rearrange the swimlanes so that they are in this order - Applicant, Loan Officer, Financial Officer, Property Appraiser, Insurance Sales Rep and System. You can rearrange the swimlanes by dragging the grey rectangle “handle” on the left and dropping them down on top of other lanes to move them below the lane you dropped on similar the illustration.

__e. The resulting process diagram will look like this. You can double click the diagram tab to make it full screen.



__f. Press the **Save** button to save our work so far.



__3. Add Activities to Process Diagram.

Now we will add the major process steps, or activities, to the diagram.

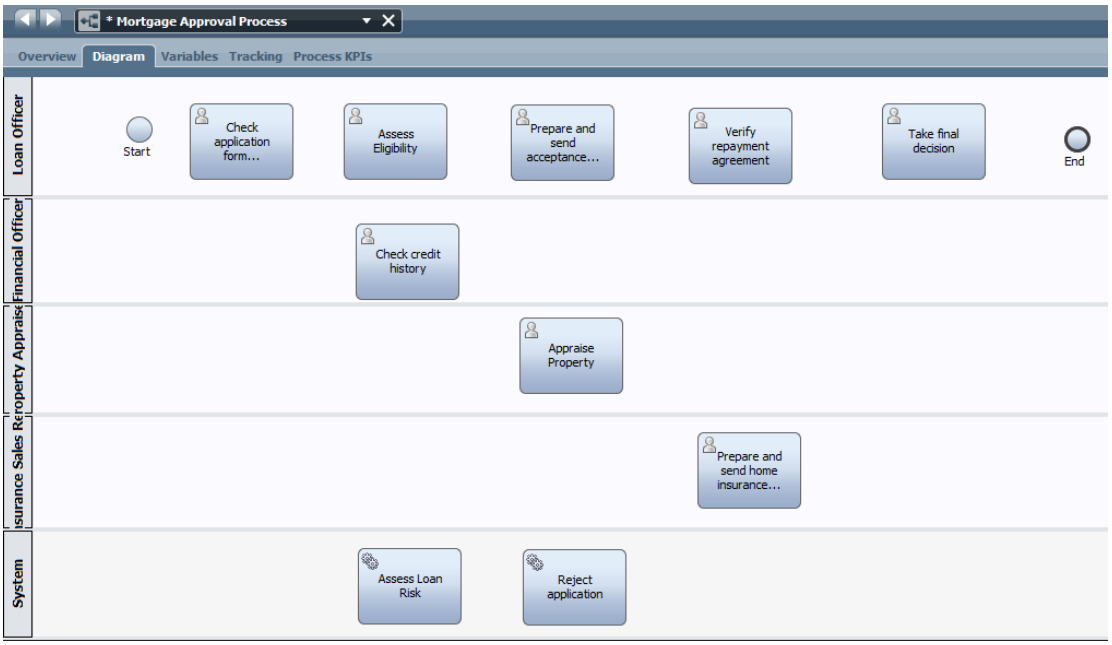
__a. From the palette on the right, click the **Activity** icon and drop it between the **Start** and **End** icons in the **Applicant** swimlane as shown:

__b. Give the Activity a name of **Check Application form** and press **Enter**.

__c. Create the following activities in each of the following swimlanes by dropping an Activity from the palette and giving the appropriate name shown:

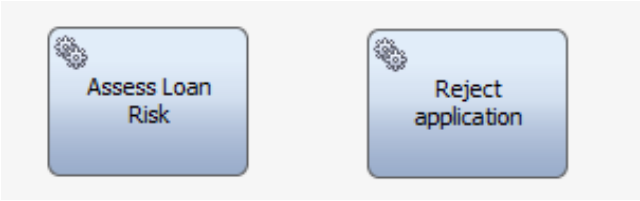
Role	**Steps**
Loan Officer	Check Credit history Assess Eligibility Prepare and send acceptance pack Verify repayment agreement Take final decision
Financial Officer	Check credit history
Property Appraiser	Appraise Property
Insurance Sales rep	Prepare and send home insurance quote

__i. Your process diagram will look similar to the illustration:

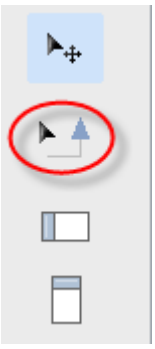


So far we have modeled the human steps in the process. We will now proceed to the system steps.

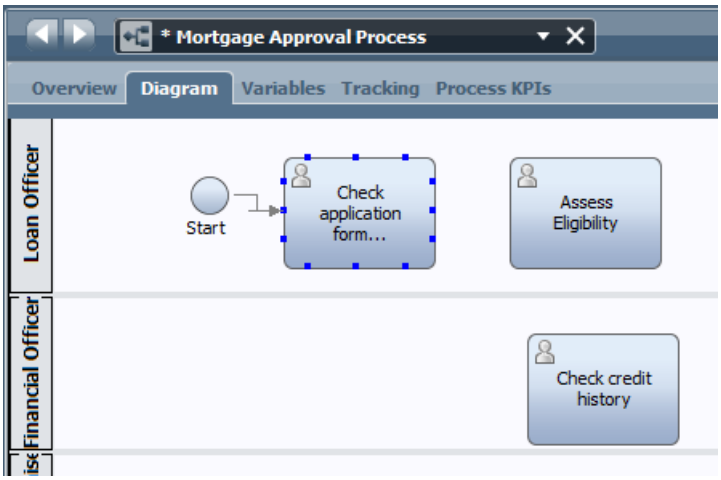
__d. Scroll down to the **System** swimlane. This swimlane is for steps that are performed by IT systems or services. Add two activities to this swimlane – **Assess Loan risk**, and **Reject application**. This part of your diagram will look similar to the illustration in the System swimlane.



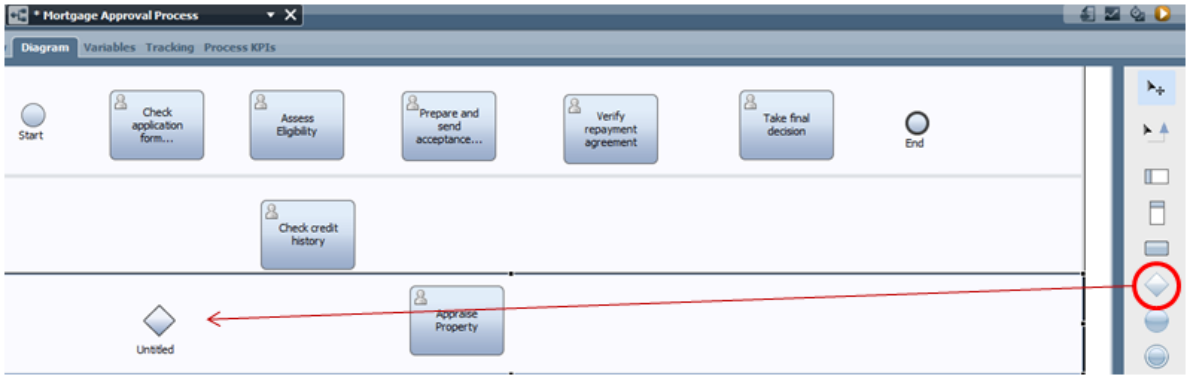
__4. We will now connect the activities in our process together. Locate and click the **connector icon** in the palette on the right.



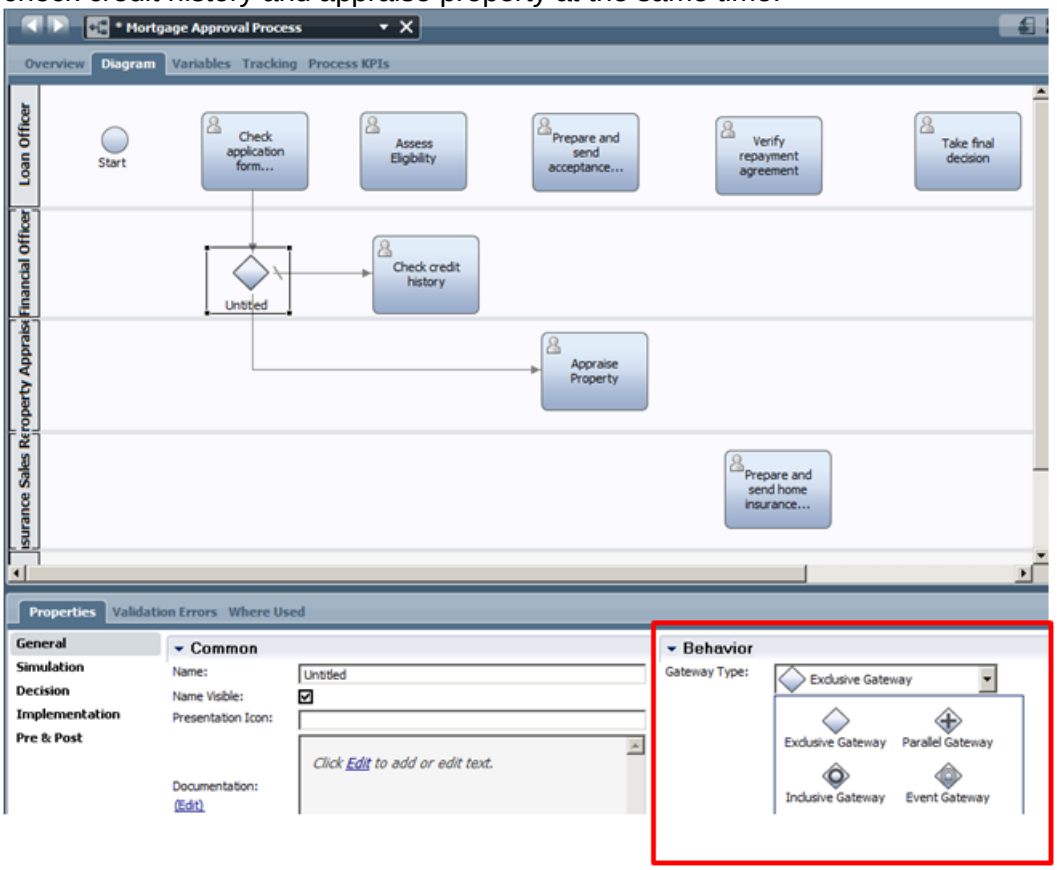
__a. As you hover over each icon in the process diagram, you will see blue connector points appear. First, connect **Start** to **Enter Application Data**, as shown:



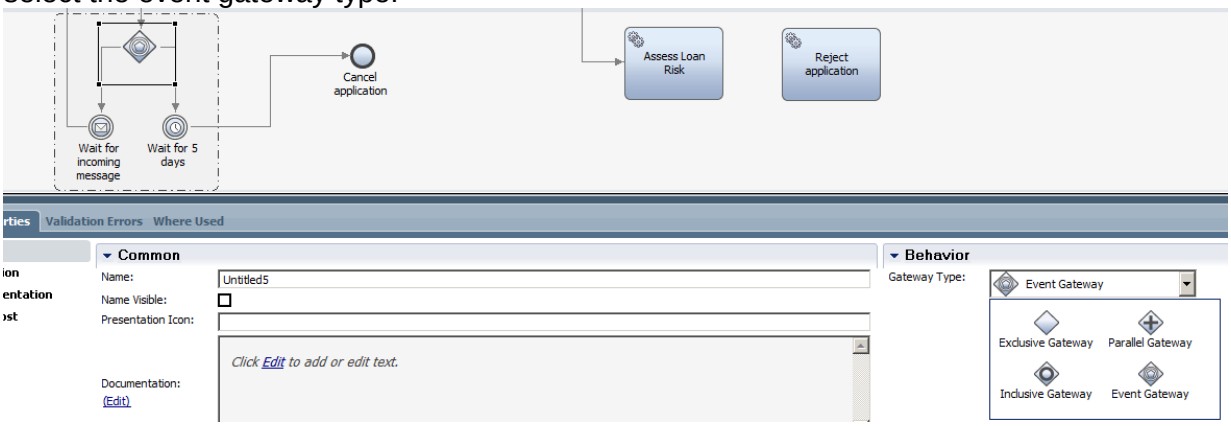
__b. We want to add a decision point to the diagram, drag and drop a diamond from the palette into the canvas:



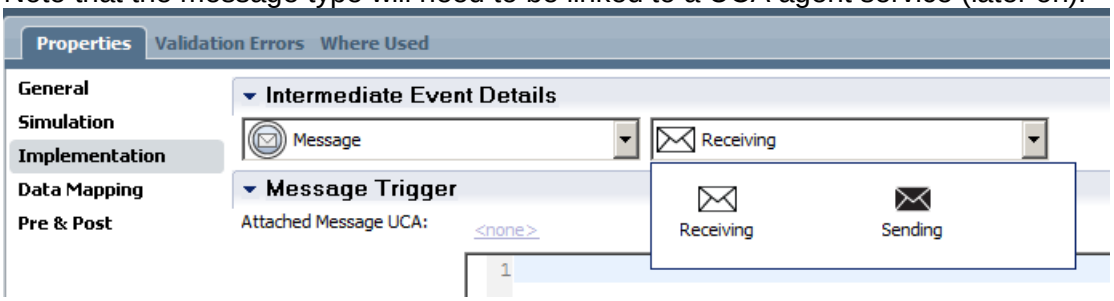
__c. Click the diamond and in the properties tab select the parallel gateway symbol to invoke check credit history and appraise property at the same time:



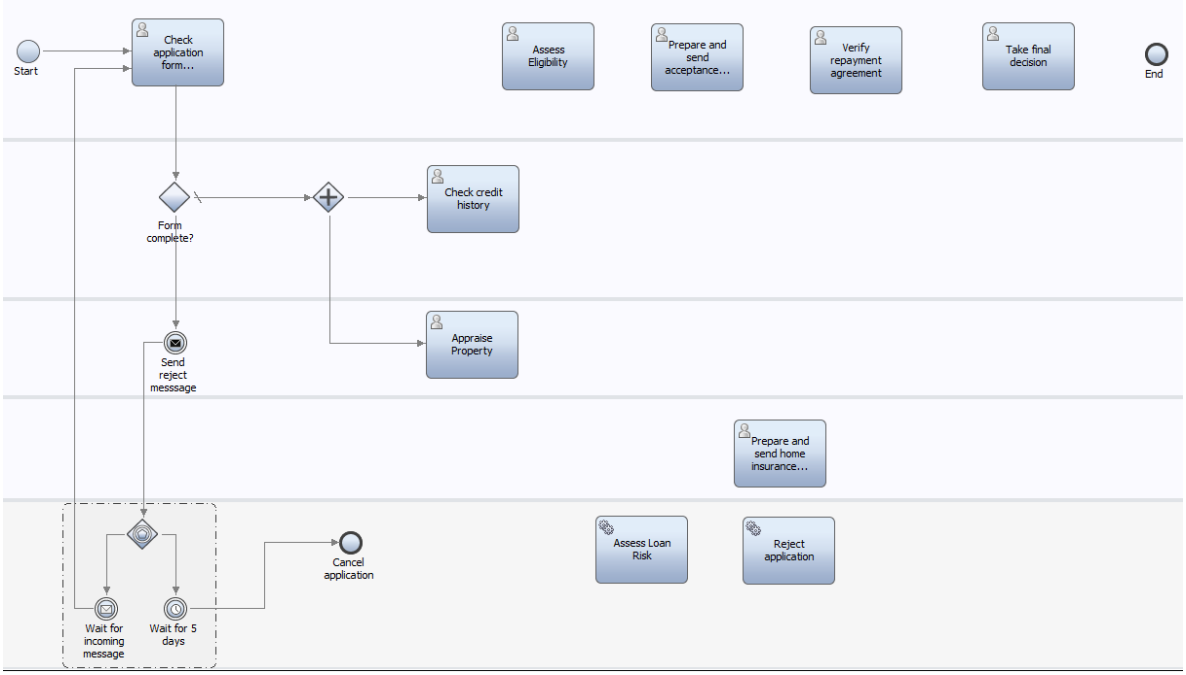
__d. Create messaging steps by dragging an intermediate message event into the canvas and select the event gateway type:



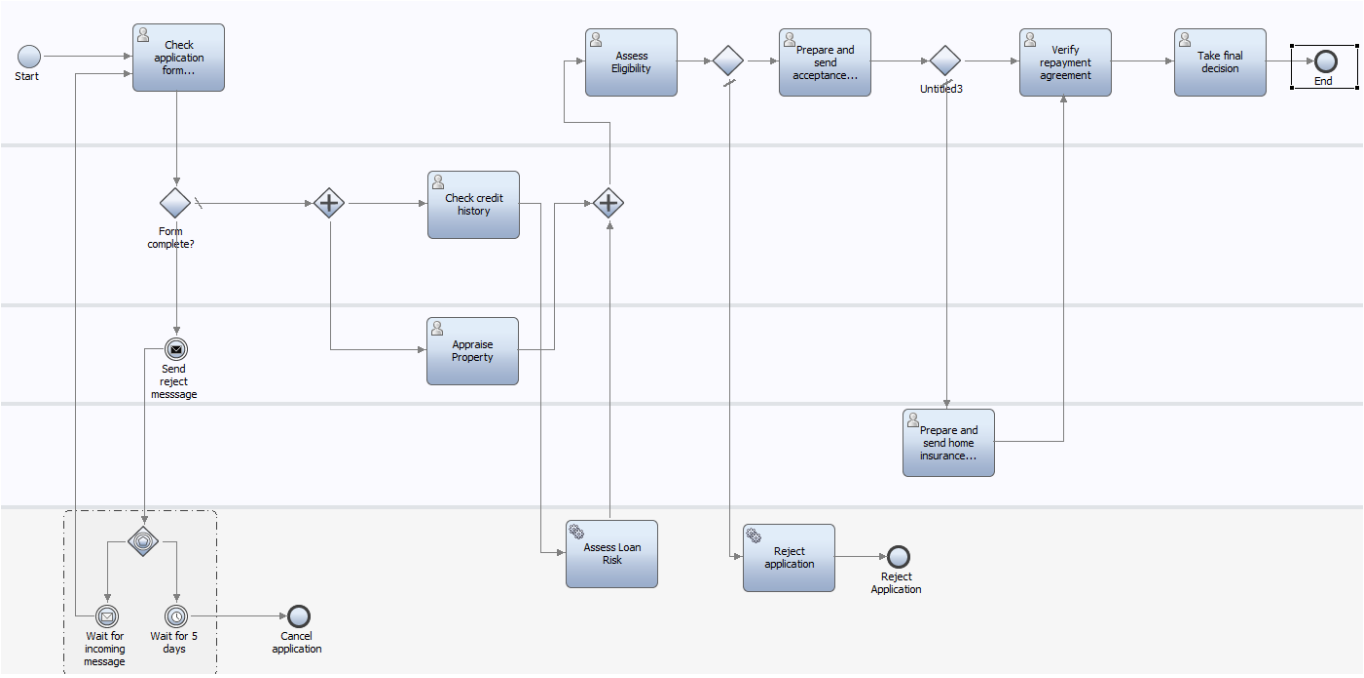
__e. Note that the message type will need to be linked to a UCA agent service (later on):



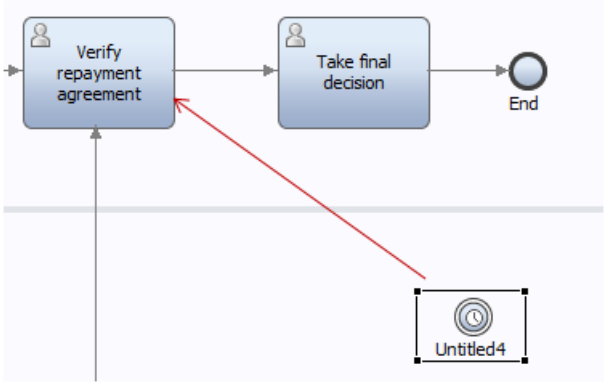
__f. Your process should now look like this:



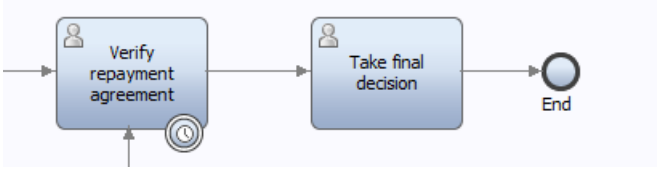
__g. connect the remainder of our process diagram together, connect the activities in the diagram as follows:



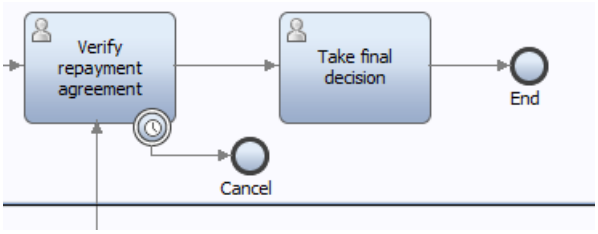
__h. Note that we still need to add a timer event to the verify repayment agreement step, drag an intermediate event to the canvas and change the type to timer:



__i. Now drag the timer event onto the verify repayment agreement until it 'snaps' onto it:

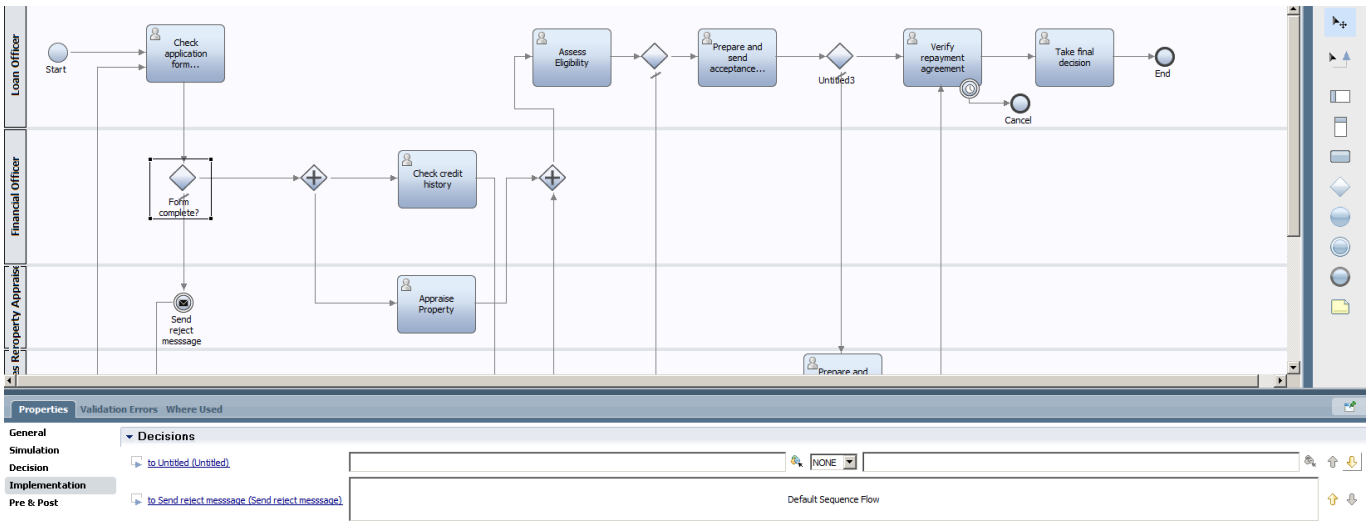


__j. You can then link the timer event to an escalation path if the timer has been triggered:

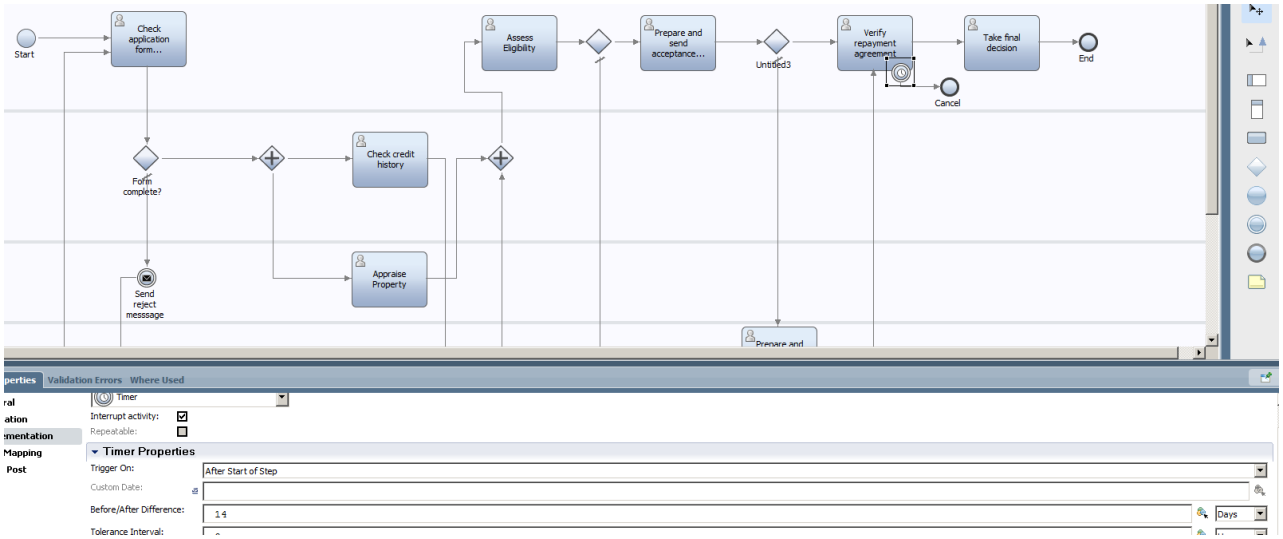


__k. After the last element, press **Esc** to unload the connector tool, and press **Save** to save your work so far.

__5. Change the decision order to have the path to check credit history be selected first by using the arrows on the right.



__6. Change the timer element that is linked to the verify repayment step to 14 days:



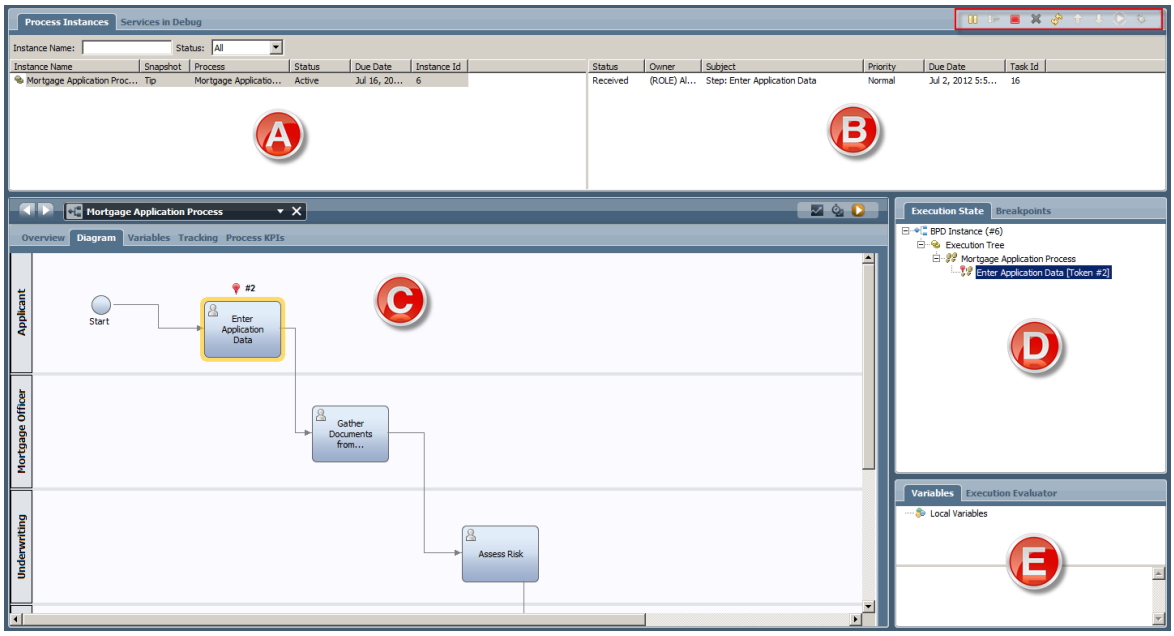
__7. We have now added all the main tasks of our process diagram and connected them together. We are now ready to test the initial flow, because, amazingly, the process is able to execute even at this early stage. We will do our first playback of the process next.

2.3.3 Play back the process – Iteration 1

__1. Locate the **Play** button at the upper right of **Process Designer**. Click the **button**.

__2. When presented with the **Switch View?** dialog, confirm by clicking **Yes**. We are confirming to switch to the Inspector view to "play back" the process steps interactively.

__3. In the **Inspector** view, there are multiple panes to be aware of as we play back the process. Also, notice the control ribbon at upper right which we will use in successive steps.



__a. **Process Instances** – Every time we click **Start**, a process instance is started and will show in this view. Multiple instances can be played through at a time.

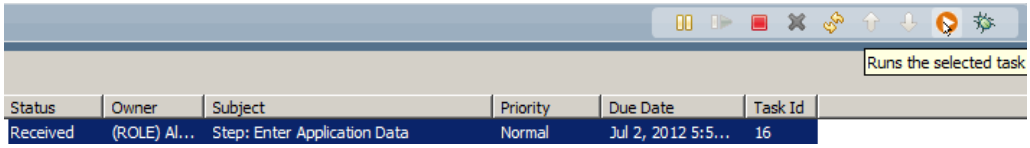
__b. **Tasks** – The current active task(s) are shown in this view, including information about the task such as status, owner, priority, and so on. Each of these tasks can be walked through as we step through the process in our playback.

__c. **Process Diagram** – This shows our business process diagram, with the current active task(s) "haloed" to show the state of the process.

__d. **Execution State/Breakpoints** – This view shows a tree view of all the steps in our process, with the current active step highlighted. The token refers to the task data that is being worked on within the task.

__e. **Variables/Execution Evaluator** – All variables are shown here so that business data can be inspected as it is changed through the process.

__4. Highlight the task in the **Tasks** pane with status of **Received**. Click the **Play** button at upper right again in the control ribbon, indicating we want to Play this task at this point in the process. We will be performing the first task (Enter Application Data) as the Applicant.



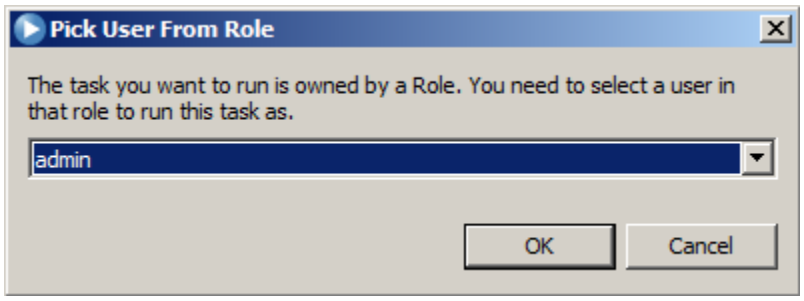
Roles and Tasks in IBM BPM

We have already added roles to each of the tasks by placing each task in a Swimlane. IBM BPM works by assigning a task to a particular role at runtime. In this lab, we will be testing all roles (Applicant, Loan Officer, Underwriting, Mortgage Manager) in the process with a common ID of “admin” for simplicity, since admin can test all roles.

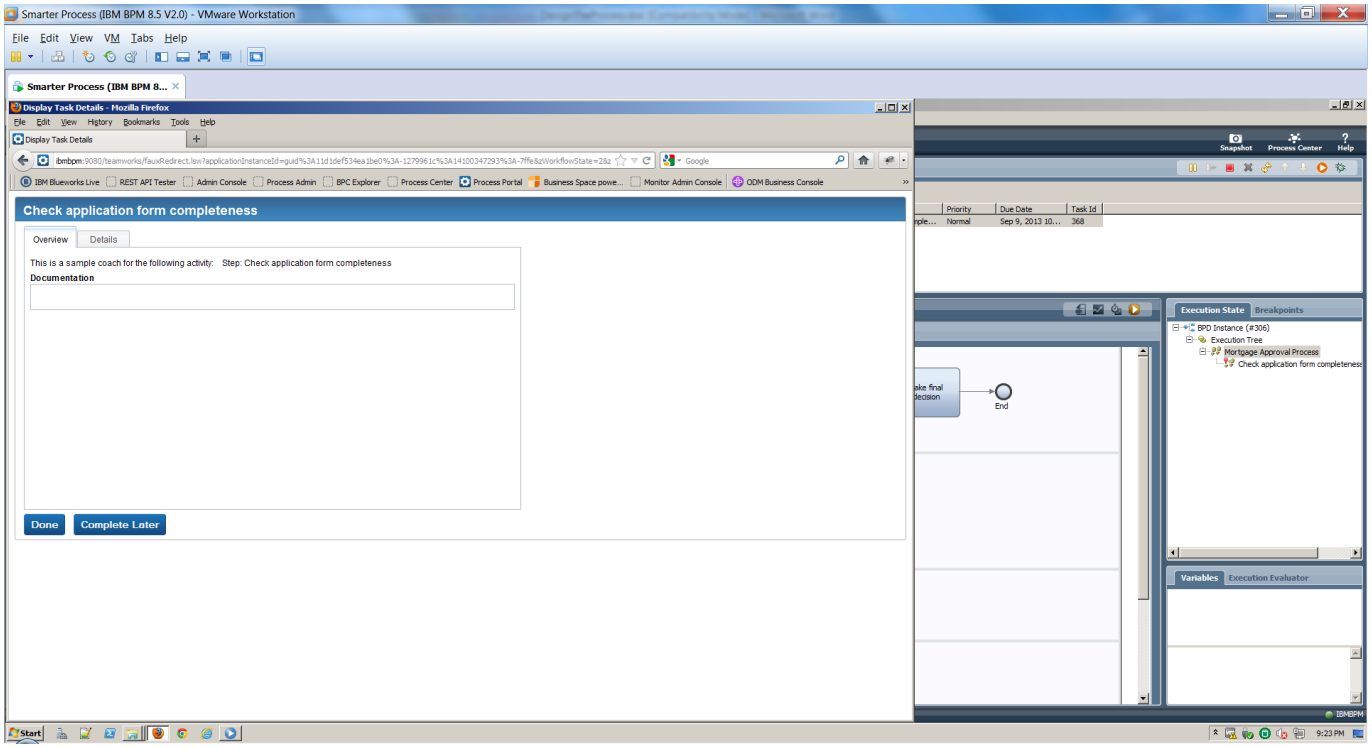
For a real project, you would configure these runtime assignment criteria to, for instance, assign the Loan Officer role to an LDAP directory group in your organization that can perform this role (such as “Home Loan Officers”). Other assignment criteria can be configured such as direct assignment to particular users, last user in lane, and custom criteria which you can customize to your project’s needs (such as “follow the sun” assignment criteria).

You will see in the Process Portal lab how task assignment works in assigning work to individual users within the Process Portal.

__5. In the **Pick User from Role** dialog box, accept the default **admin** role to run the task and click **OK**.



__6. You will see a screen similar to this illustration. This shows a default user screen for the Enter Application Data Step.

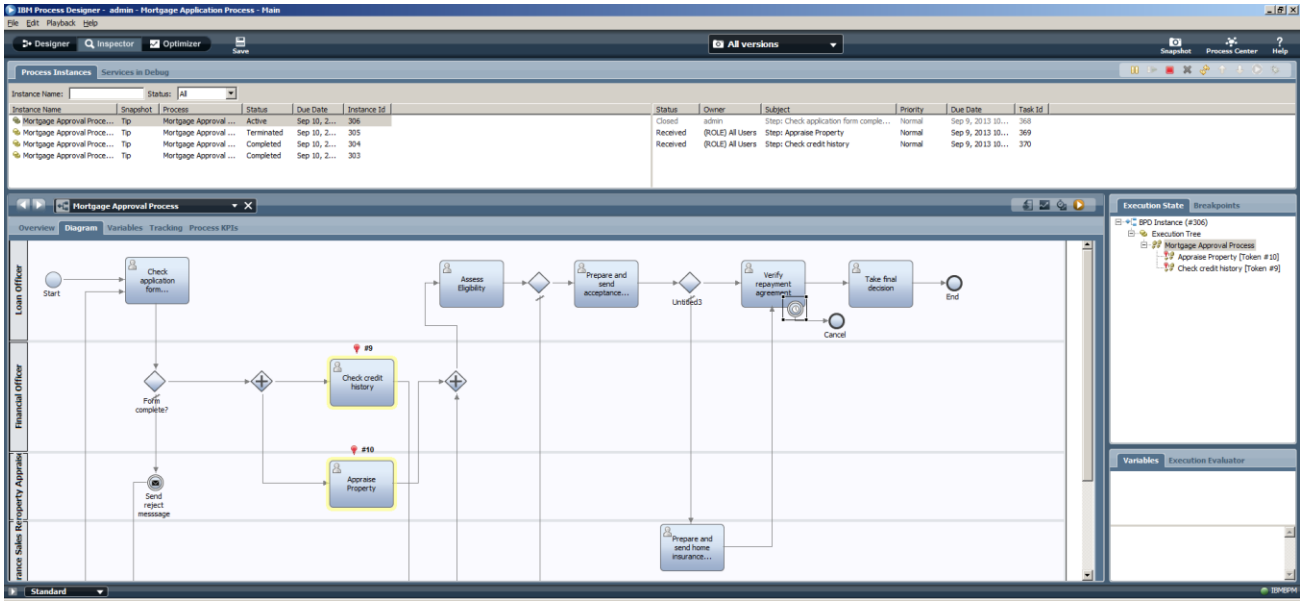


We will make this look a lot nicer in a short while. For now, it's important to understand that even before we have assigned what business data the process will need to work with, we are able to step through the process in its current state, completing activities in their default form. This is useful in being able to validate the current process activities and basic flow, and can be used in an initial playback between the process designer and process owner to ensure that requirements at this early stage have been captured effectively and completely.

__7. Click **Done** to complete the task.

__8. Click the **Refresh** button in the control ribbon to refresh the state of the process. 

__9. You will notice that in the Tasks pane, the first task now shows closed and the second task is now ready to run. In addition, the process diagram shows the current task highlighted in the process, and a new token is being worked on in the execution state pane.



__10. As before, highlight the current task and click **Play**, and click **OK** to **Pick User From Role** as **admin**. As before, you will be presented with the default screen for the next step in the process. Continue completing each task, refreshing the process, and clicking **Play** on each task as you continue through to the end of the process. The process has now completed.

System Tasks

You may have noticed that when you refreshed the process at a certain point, the activities that were in the System lane were not available to be played as a user screen. This is because these are designed to be executed in the background without human interaction. These could be calls to databases, web services, packaged applications, or other back-end systems.

We will do more with system tasks later, but it is important to understand that processes typically are composed of a combination of human and system tasks, and that the Inspector will step through the entire process, including both types of tasks.

__11. Since we have tested through the first iteration of our process, this would be a good time to take a snapshot. This allows us to save our work at this stage in a “version” of the process application to this point.

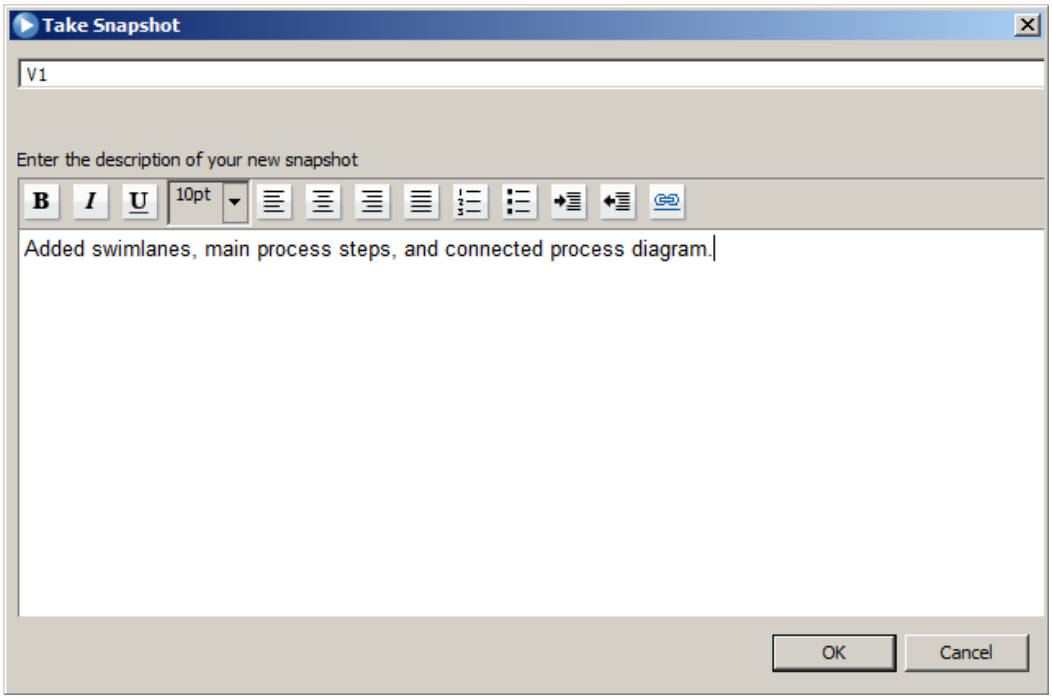
Snapshots in IBM BPM

IBM BPM provides the ability to take “snapshots” throughout the BPM design process. This gives you the ability to create a version of your work quickly and easily within Process Designer, storing the snapshots of your application in Process Center. Snapshots give you the ability to try out changes and then roll back to a prior version later if desired, and also are an effective mechanism for rolling versioned processes into production.

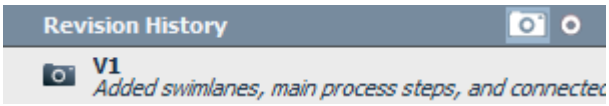
__a. In the upper-left corner, click **Designer** to ensure you are in the Designer view.

__b. In the upper-right corner, click the **Snapshot** button.

__c. Enter the following information for snapshot name and description and click **OK**:



__d. The snapshot will now appear in **Revision History** on the bottom left and can be rolled back to later if desired.



__12. In summary, we have designed the initial process by:

__a. adding swimlanes for roles

__b. adding activities (some human steps, some system steps)

__c. connecting the activities into an overall process flow

__d. tested the basic process flow in the Inspector (also known as a process playback)

__e. taken a snapshot of the first version of the process

The snapshot of the process is available for import here:

Snapshot: Initial process design: Mortgage_Application_Process - Part_1_Process_Design.twx

2.4 Adding “Coach” screens to process – Iteration 2

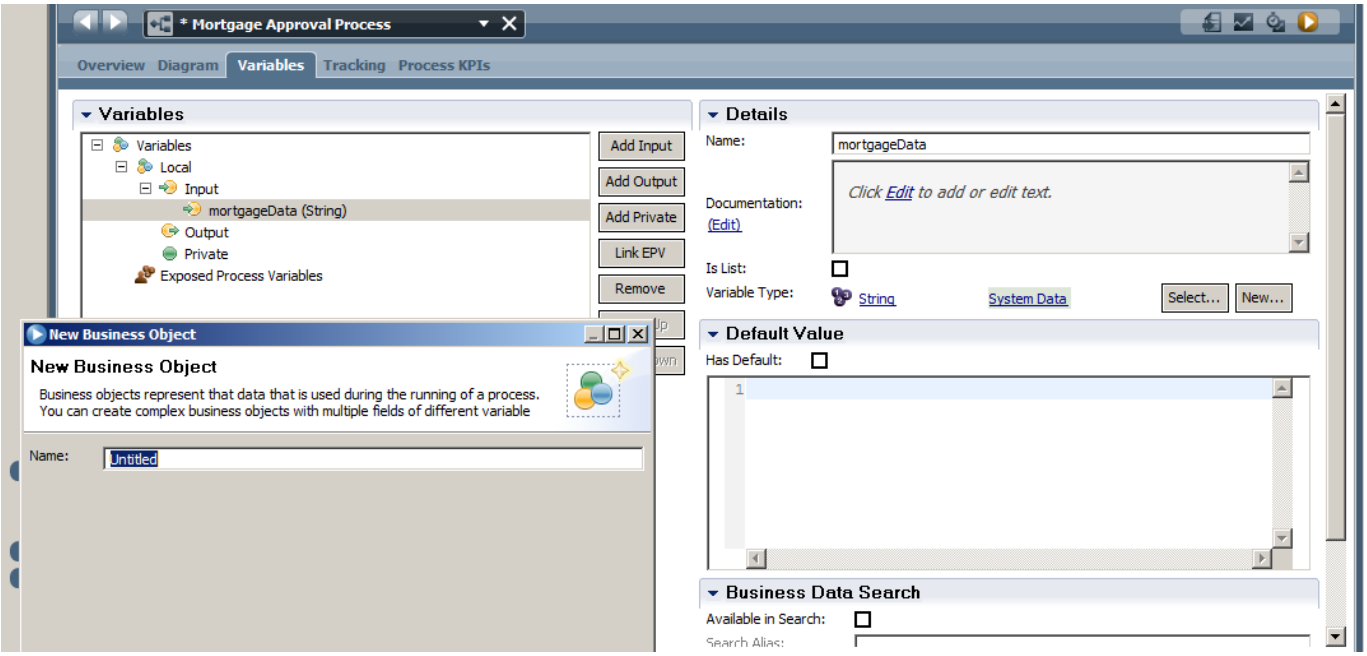
In this section, we will enhance the process by adding business data to keep track of the loan application as it proceeds through the process. We will create highly engaging user screens to present to the process participants as they carry out the human activities within the process. We will then do a second playback of the process and take a second snapshot of our work.

2.4.1 Add business data to the process

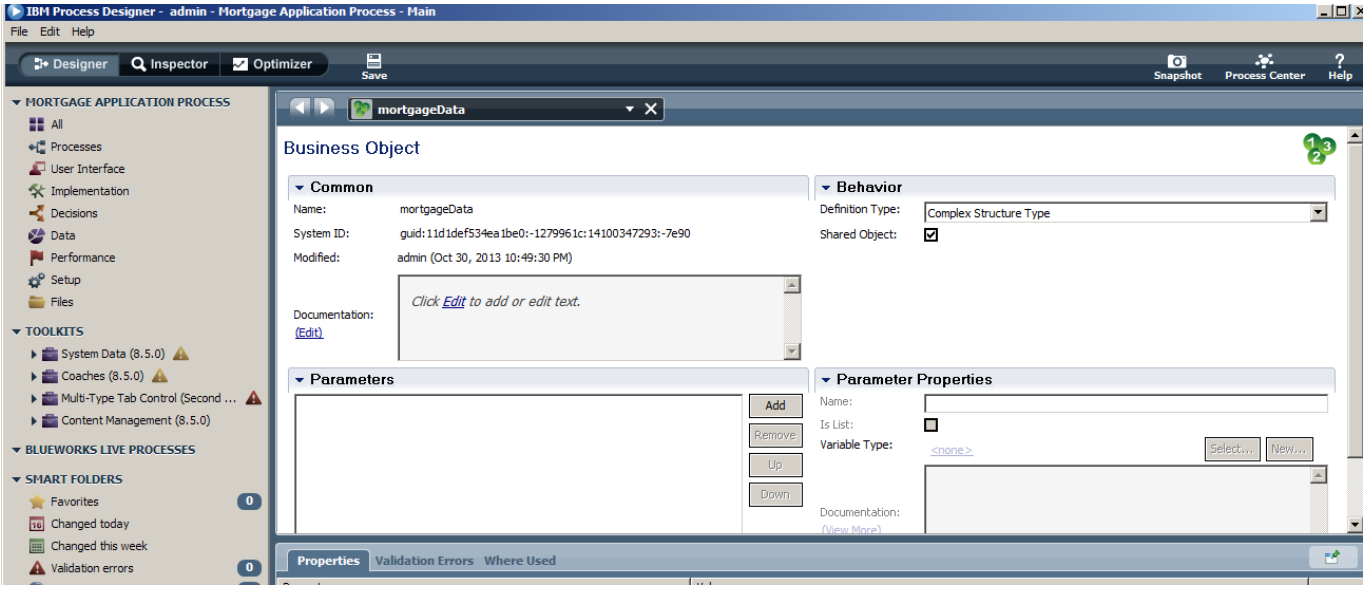
__1. The Loan Assessment Process should be displayed in the main pane. Click the **Variables** tab in the process editor. We are going to add our business data here. Click **Add Input** to add a variable to hold the loan application data for the process.

__2. In **Details**, type **mortgageData** as the name of the private variable.

__3. For **Variable Type**, click the **New** button and enter mortgageData as the new object name.



__4. The variable editor screen now pops up. This screen is used to define the structure of the data element that is passed through the process, the object can be of complex type with several sub objects. Make sure you check the box "Shared Object" so that data in parallel paths is immediately synchronized.

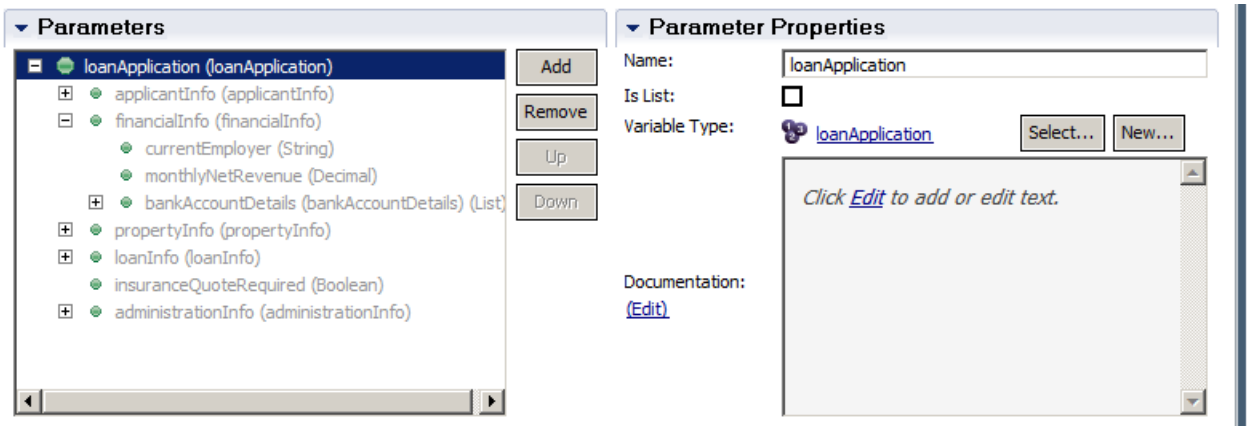


__5. Click "Add" to add a new entry to the **mortgageData** variable, enter **loanApplication** as its name and select New to define a new object for it, name this new object **loanApplication**.

__6. Inside the loanApplication object enter the following data fields:

Object Name	**Variables and type**
applicantInfo *	Name (String) Surname (String) Email (String) Home Phone (String) Mobile Phone (String) CurrentAddress (String) PreviousAddress (String) (non mandatory)
financialInfo *	CurrentEmployer (String) MonthlyNetRevenue (Double) BankAccountDetails (bankAccountDetails)
bankAccountDetails (List) *	bankName (String) accountType (String) accountNumber (String) accountBalance (Double)
propertyInfo *	propertyType (String) address (String) purchasingPrice (String)
loanInfo *	loanType (String) loandProvider (String) amount (Double) duration (Double) startDate (Date) endDate (Date) interestRate (Double) interestType (String)
insuranceQuoteRequired *	Boolean
administrationInfo	applicationIdentifier (String) submissionDate (Date) revisionDate (Date) status (String) comments (String) eligibility (Boolean) loanOfficerIdentifier (String)

__7. * indicates a mandatory field. We'll implement the logic for that later.



The screen should look as follows with the mortgageData the top object and the loanApplication its first complext object. Let's now create the rest of the objects.

__8. The **credit history report data** fields are as follows:

Object Name	**Variables and type**
creditHistoryReport	financialOfficerIdentifier (String) loanApplicationReference (String) creditInformation (creditInformation)
creditInformation	loanApplicationHistory (loanInfo) (List) overdueCreditAccounts (loanInfo) (List) currentCreditCardProvider (loanInfo) (List) publicRecordInfo (String) bankruptcyInfo (String) creditAssessment (String)

__9. The **risk assessment data** fields are as follows:

Object Name	**Variables and type**
riskAssessment	creditHistoryReportReference (String) riskWeight (Decimal)

__10. The **property appraisal data** fields are as follows:

Object Name	**Variables and type**

propertyAppraisal	loanApplicationReference (String) identifier (String) surroundingProperties (surroundingProperties) (List) estimatedValue (Decimal) comments (String)
surroundingProperties	Name (String) Value (Decimal)

__11. The **repayment data** fields are as follows:

Object Name	**Variables and type**
repaymentAgreement	loanApplicationReference (String) monthlyRepaymentAmount (Decimal) numberOfRepayments (Decimal)

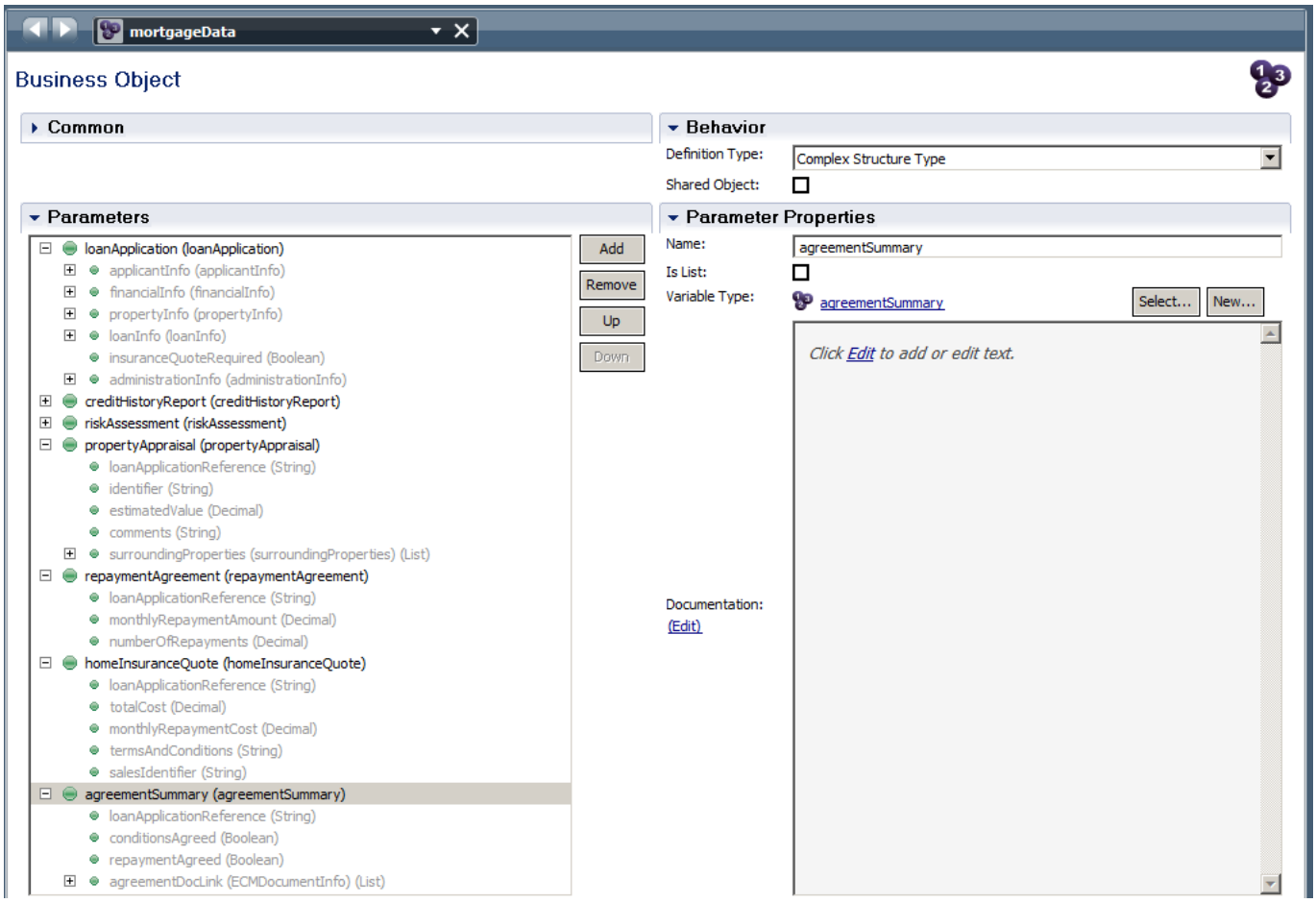
__12. The **home insurance data** fields are as follows:

Object Name	**Variables and type**
homeInsuranceQuote	loanApplicationReference (String) totalCost (Decimal) monthlyRepaymentCost (Decimal) termsAndConditions (String) salesRepIdentifier (String)

__13. The **agreement summary data** fields are as follows. if the ECMDocumentInfo type is not visible, activate the Content Management toolkit, click on the '+' symbol next to TOOLKITS to add it, right click on the toolkit to upgrade its version to 8.5.0 if necessary.

Object Name	**Variables and type**
agreementSummary	loanApplicationReference (String) conditionsAgreed (Boolean) repaymentAgreed (Boolean) agreementDocLink (ECMDocumentInfo) (List)

__14. The final business object should look like this:



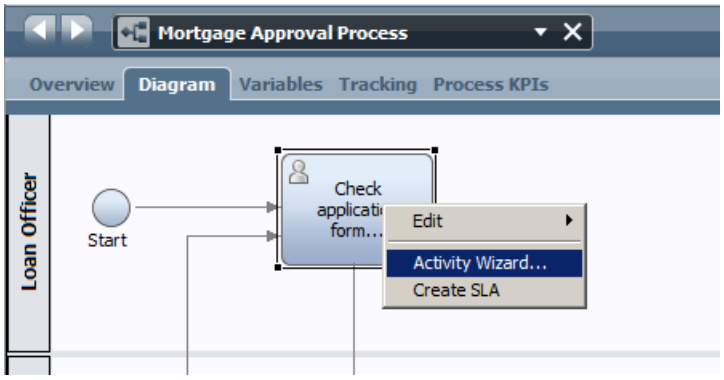
As you can see all data is part of a single encapsulating object called mortgageData, this structure reduces the need for reference strings that are part of the current object such as loanApplicationIdentifier. Furthermore we are able to re-use the riskAsessment object multiple times as part of mortgageData.

__15. Press **Save** to save your changes so far.

2.4.2 Create user screens to Enter Application Data

__16. This chapter will explain how to create a Human Service for Enter Application Data

__17. Click the **Diagram** tab in the process editor. Right-click the **Check application form** activity and click the **Activity Wizard**.

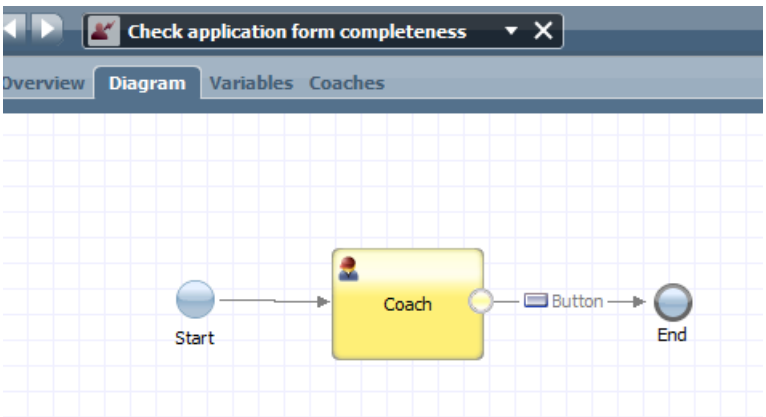


__a. In the **Activity Wizard**, accept the default names for **Activity Name** and **Service**. Click **Next**.

__b. Accept the defaults for passing the **mortgageData** Business Process Variable as **Input** and **Output**. Click **Finish**.

__18. Modify Generated Coach

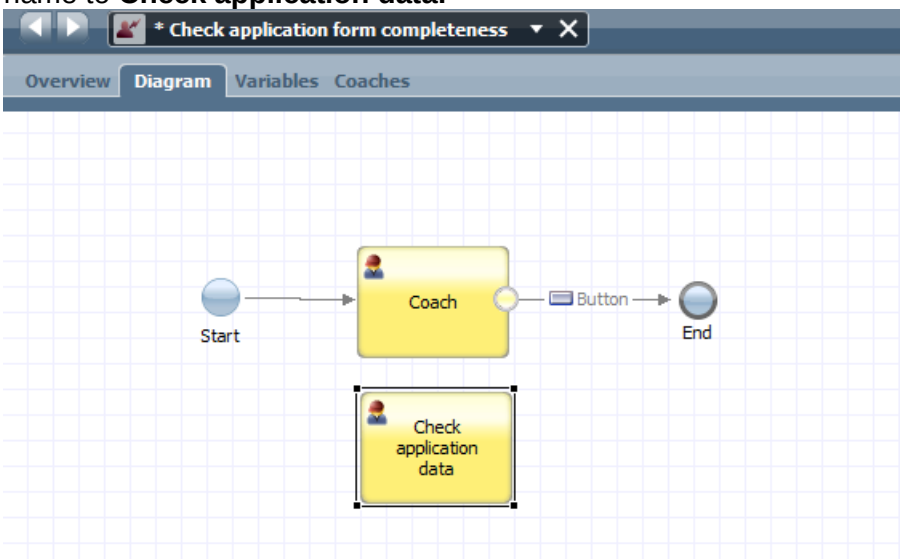
__a. Go back to the **Check application form** activity in the process diagram. Double-click **Check application form** activity. You will now see the diagram for the human service with a single **Coach** element in the middle and a **Start** and **End**.



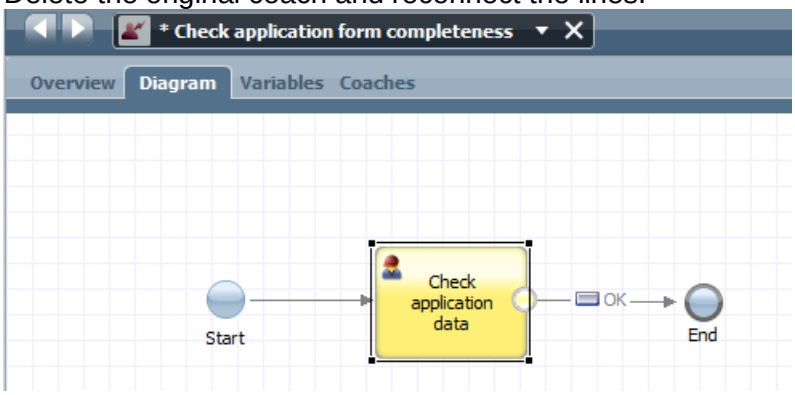
Coaches

Coach is the term in IBM BPM for a user screen aka form, which is designed to “coach” process participants through the completion of their task. IBM BPM provides a built-in drag-and-drop screen editor, making it easy to create rich user interfaces for our process applications within the same Process Designer tool.

__b. Drag and drop a new coach from the palette on the right onto the canvas, change its name to **Check application data.**



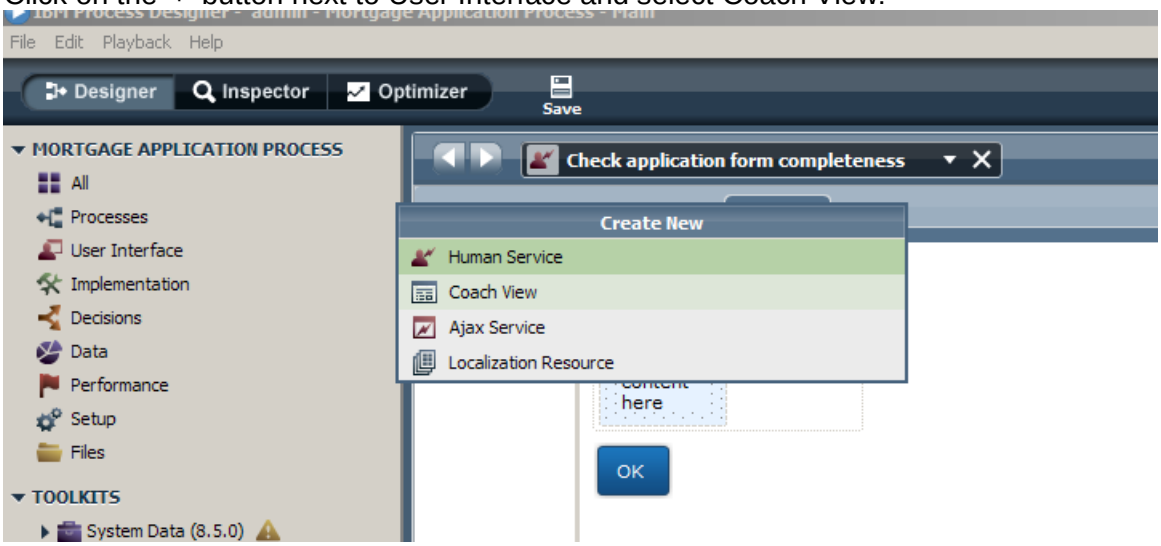
__c. Delete the original coach and reconnect the lines.



__d. Now double click on the coach. You will see an empty form with a single OK button.

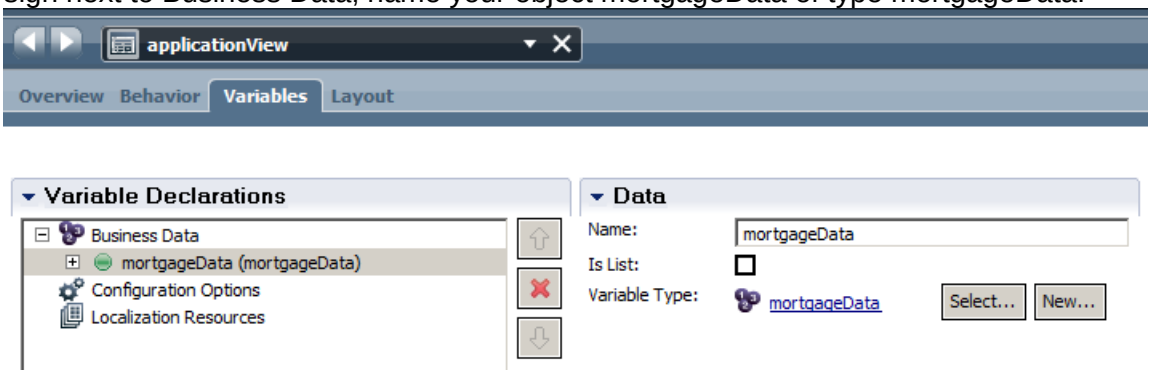
__e. We now have an empty form. The variable data that we have in our mortgageData object is quite large so we want to split it up. In order to do that we use a feature called 'coach views' which are parts of a coach that can be shared across multiple coaches.

__f. Click on the '+' button next to User Interface and select Coach View:

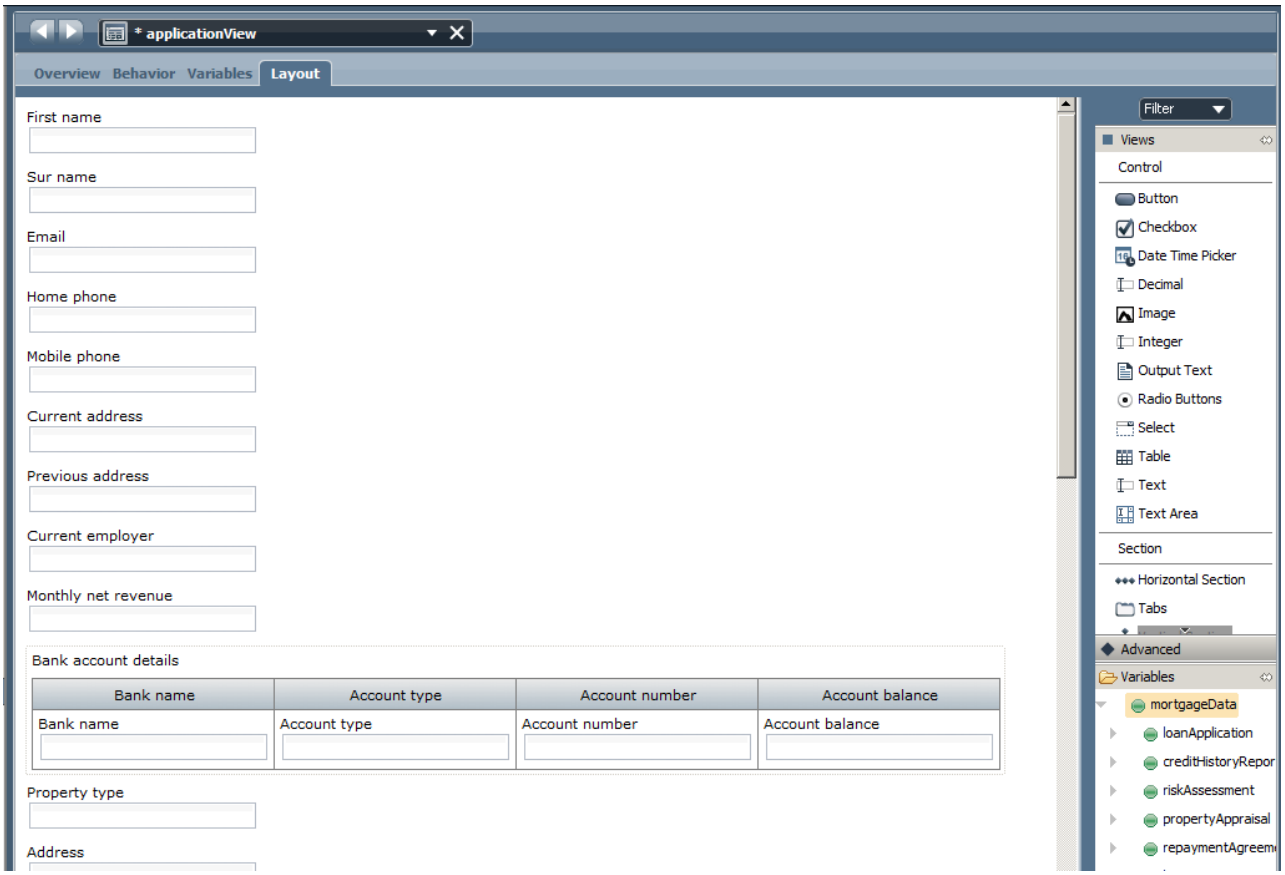


__g. Name the new coach View: applicationView

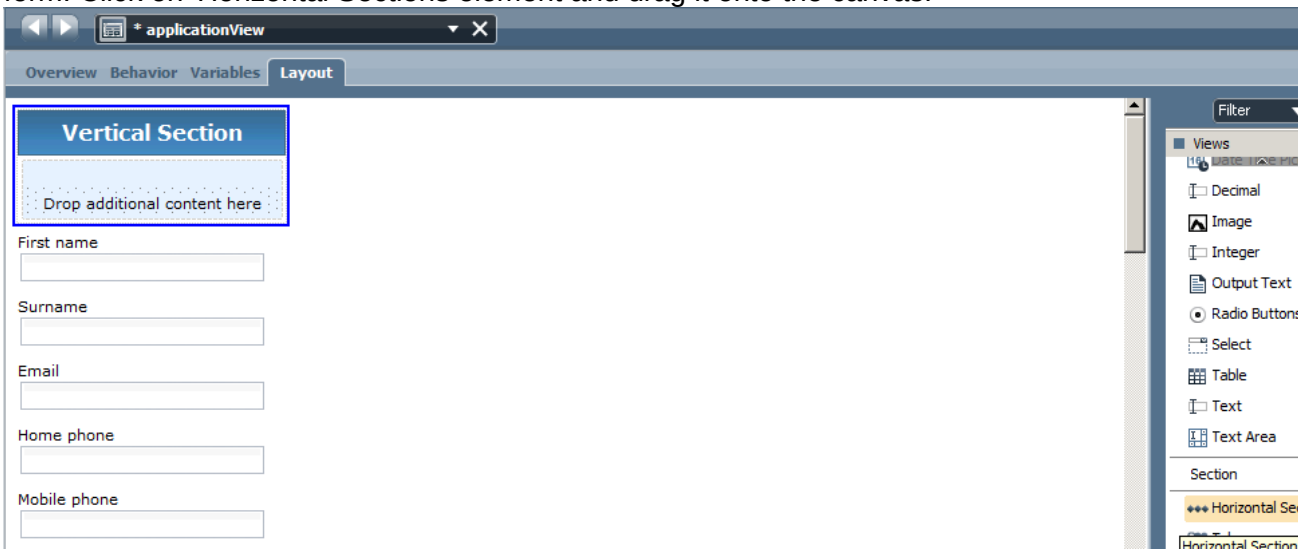
__h. You will now be presented with a black canvas, click on the Variables tab and click the '+' sign next to Business Data, name your object mortgageData of type mortgageData:



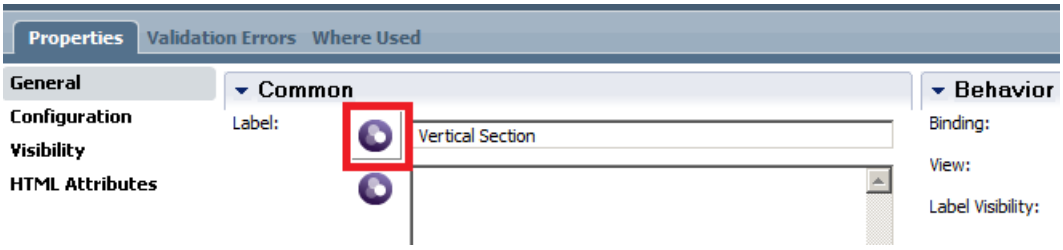
__i. From the palette on the right, click the **Variables** drawer and drag a **loanApplication** onto the canvas as shown, all the variables in the object will now be shown on the screen.



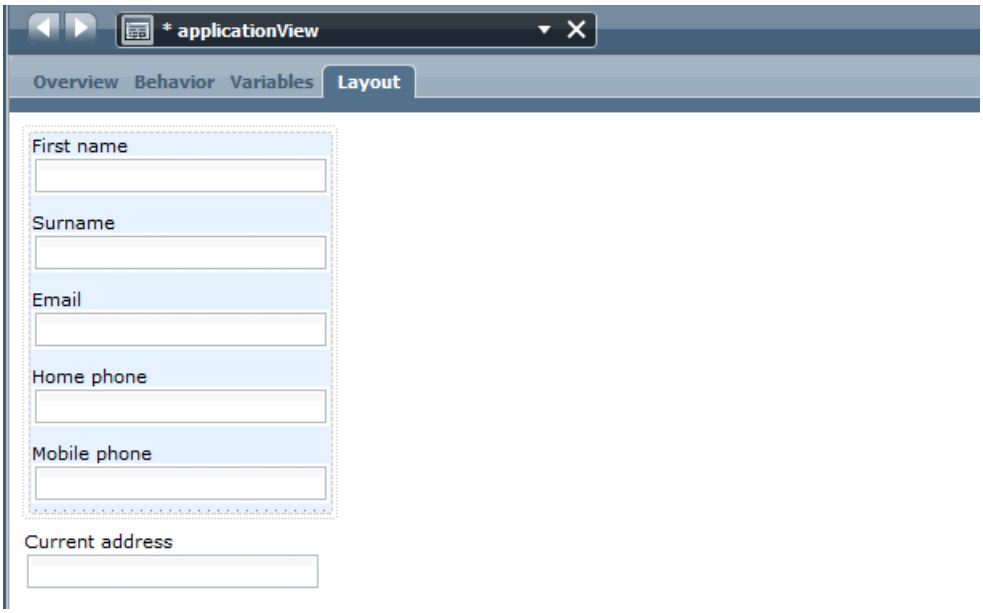
__j. We need to re-arrange the data fields to make it easier to identify the sections of the form. Click on 'Horizontal Sections element and drag it onto the canvas:



__k. In the properties part of the Vertical Section remove the title by clicking on the button in front of the label.

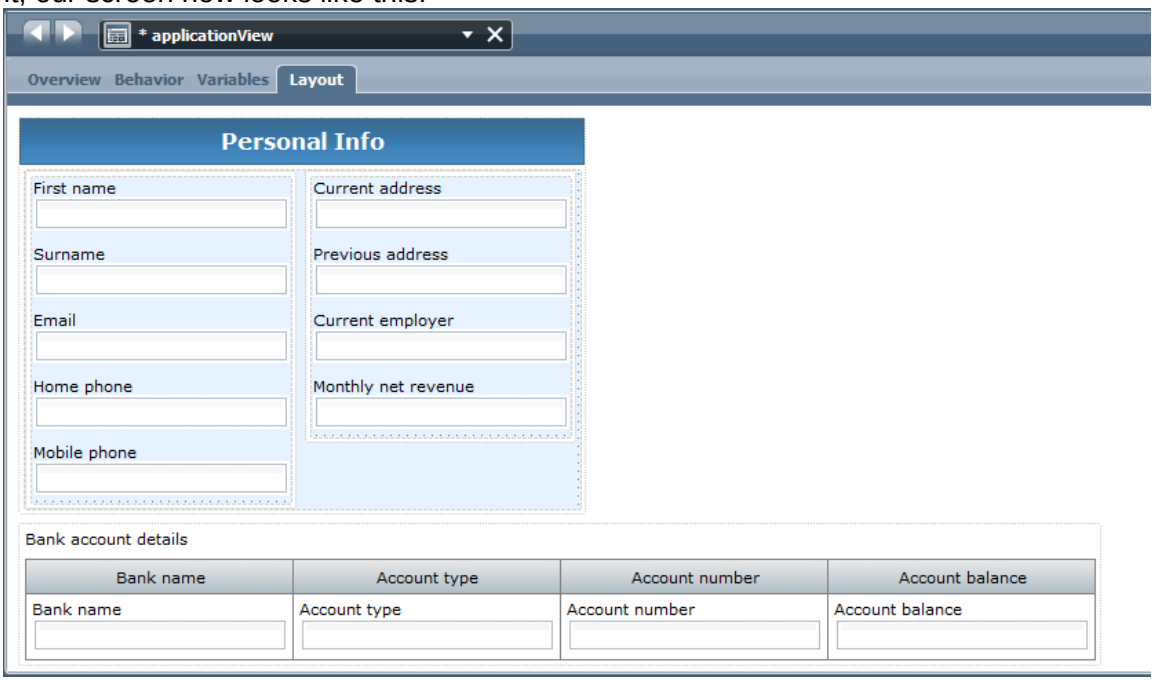


__l. Drag the first 5 fields into the vertical section:

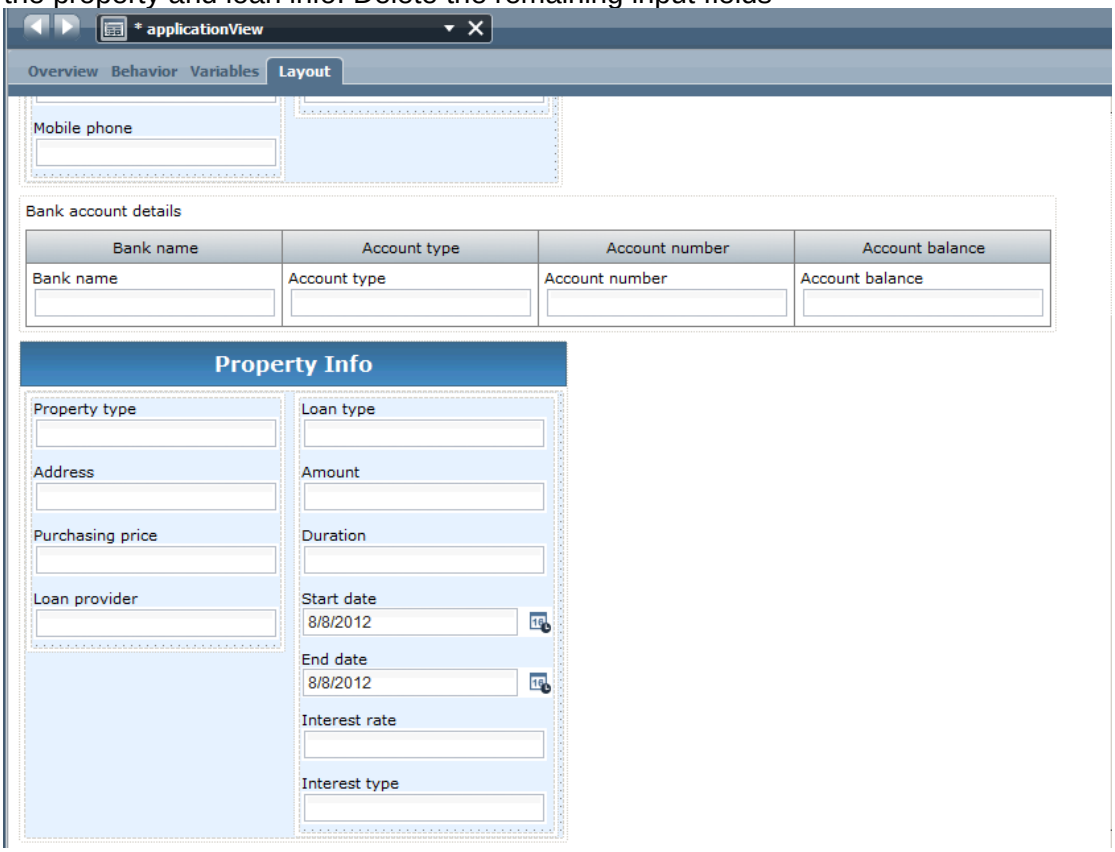


__m. Create another vertical section, remove the title and drag the next 4 fields into it.

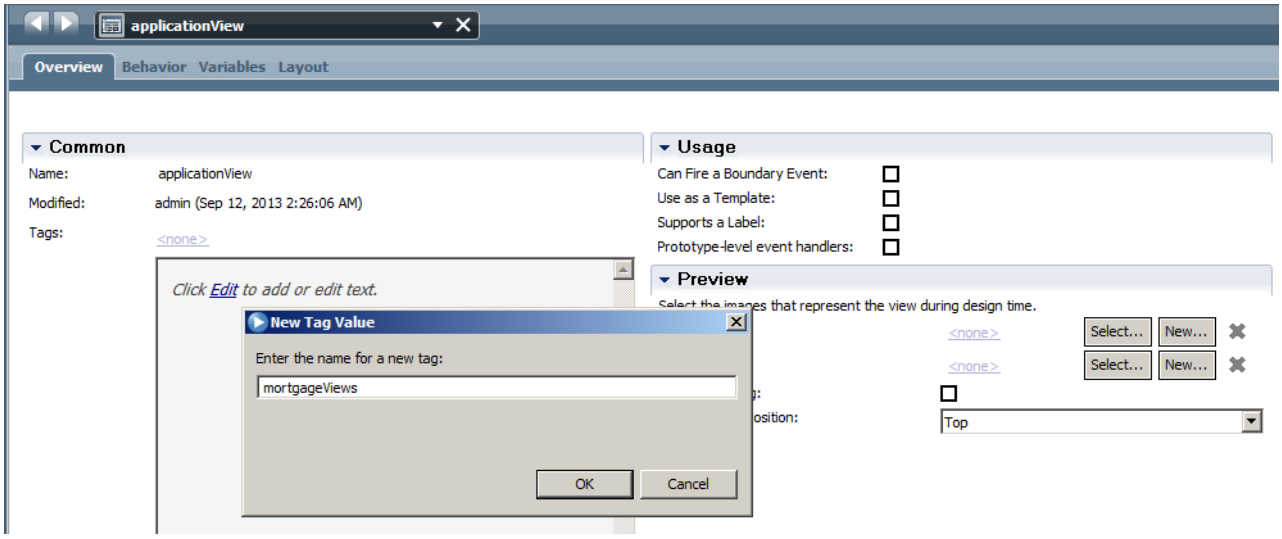
__n. Create a horizontal section, name it **Personal Info** and drag the two vertical sections into it, our screen now looks like this:



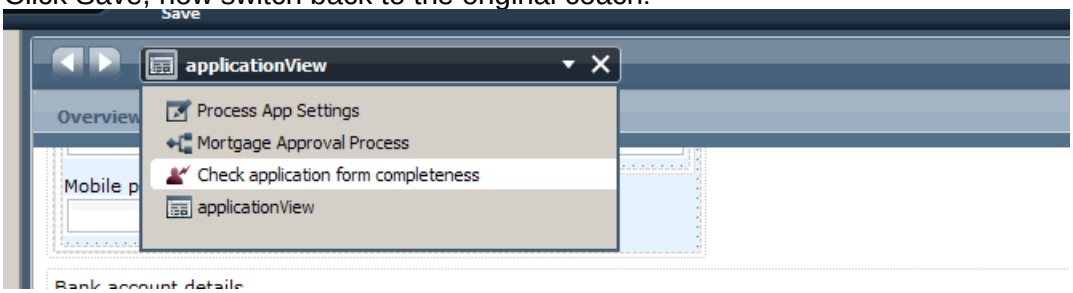
__o. Keep the bank account details in place, scroll down and create an identical structure for the property and loan info. Delete the remaining input fields



__p. Switch to the Overview tab, and create a new tag for this coach view, name it mortgageViews. This will allow us to group all views related to a certain topic

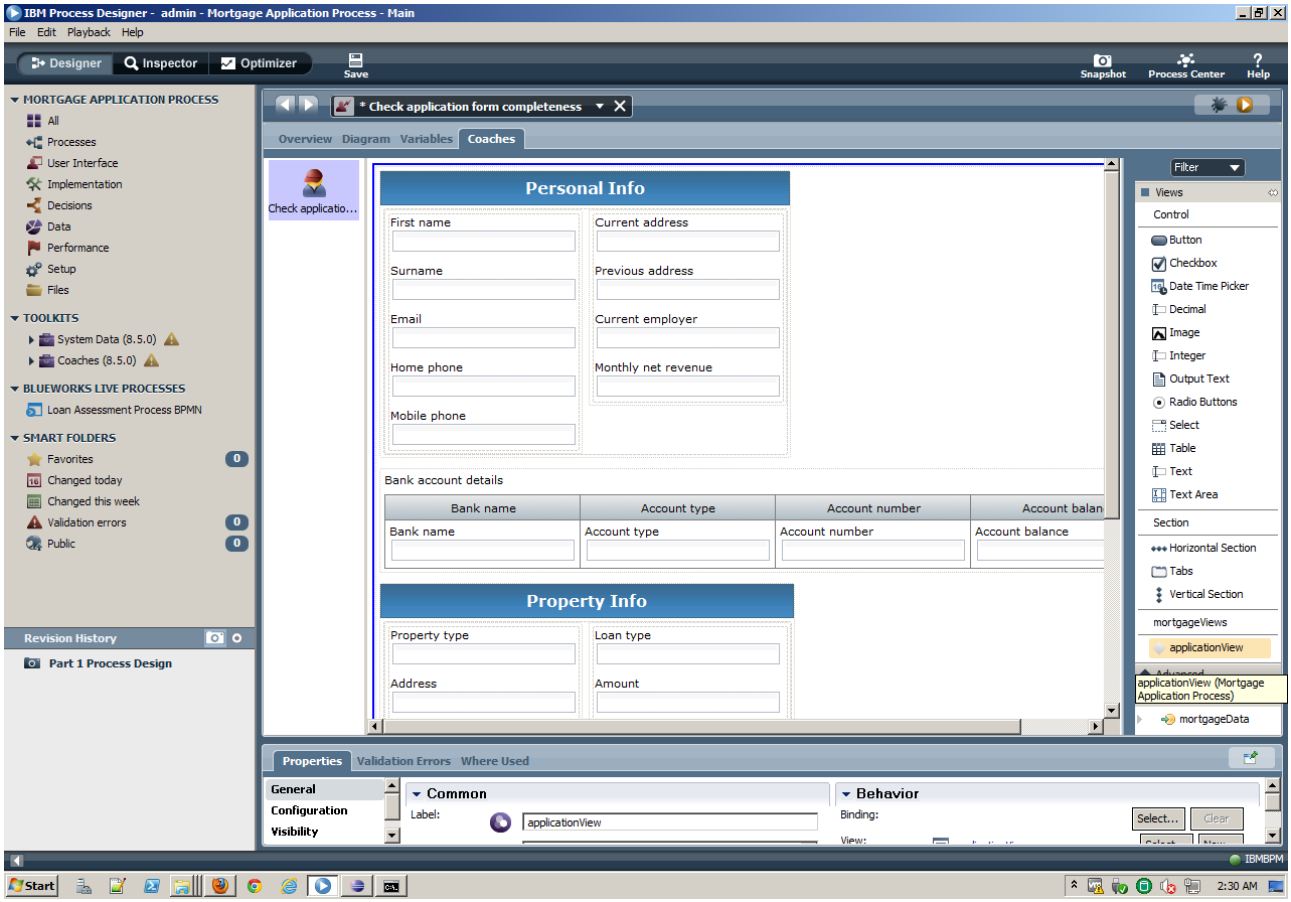


__q. Click Save, now switch back to the original coach:

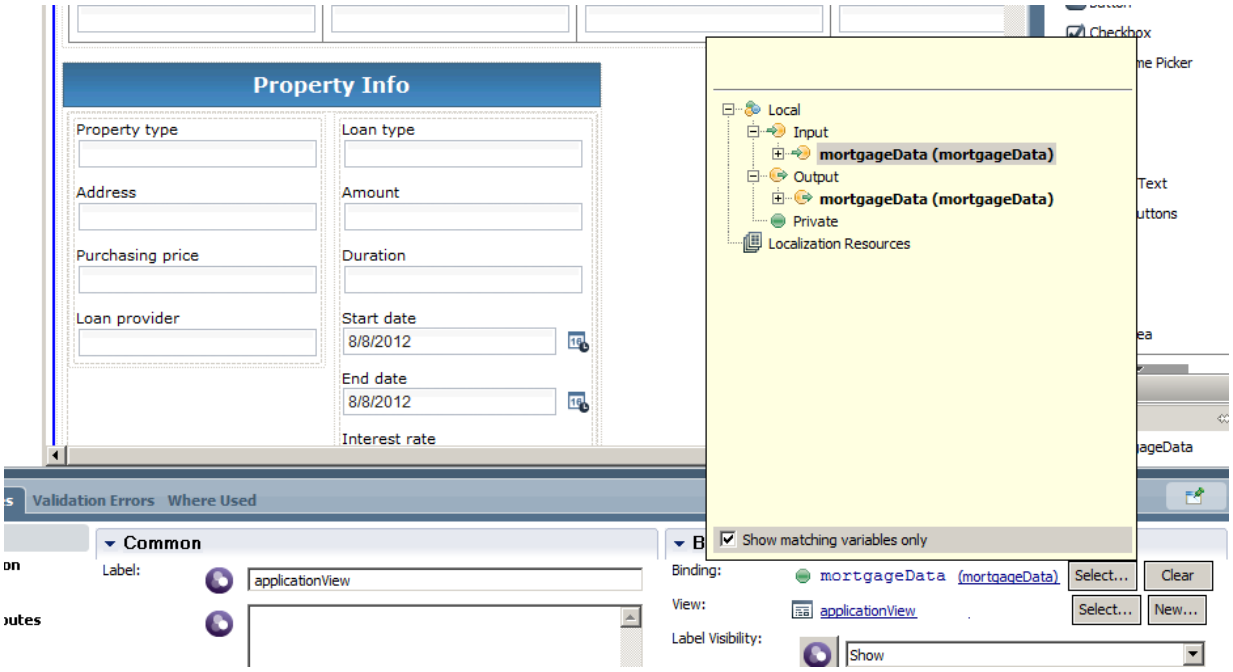


__r. Make sure that the mortgageViews checkbox is checked:

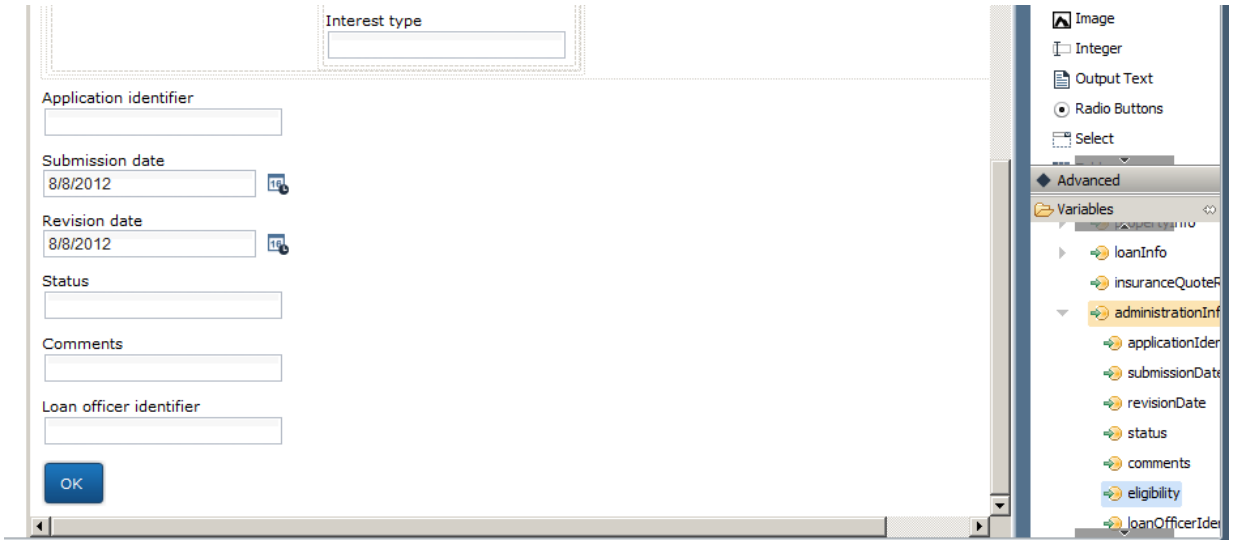
__s. Click **applicationView** and drag it into canvas.


IBM Process Designer - admin - Mortgage Application Process - Main
File Edit Playback Help
Designer
Inspector
Optimizer
Save
Snapshot
Process Center
Help
MORTGAGE APPLICATION PROCESS
All
Processes
User Interface
Implementation
Decisions
Data
Performance
Setup
Files
TOOLKITS
System Data (8.5.0)
Coaches (8.5.0)
BLUEWORKS LIVE PROCESSES
Loan Assessment Process BPMN
SMART FOLDERS
Favorites
Changed today
Changed this week
Validation errors
Public
Revision History
Part 1 Process Design
* Check application form completeness
Overview Diagram Variables Coaches
Check applicatio...
Personal Info
First name
Current address
Surname
Previous address
Email
Current employer
Home phone
Monthly net revenue
Mobile phone
Bank account details
Bank name
Account type
Account number
Account balan
Account balance
Property Info
Property type
Loan type
Address
Amount
Filter
Views
Control
Button
Checkbox
Date Time Picker
Decimal
Image
Integer
Output Text
Radio Buttons
Select
Table
Text
Text Area
Section
Horizontal Section
Tabs
Vertical Section
mortgageViews
applicationView
applicationView (Mortgage Application Process)
mortgageData
Properties Validation Errors Where Used
General
Configuration
Visibility
Common
Label:
applicationView
Behavior
Binding:
Select...
Clear
IBMBPM
Start
2:30 AM

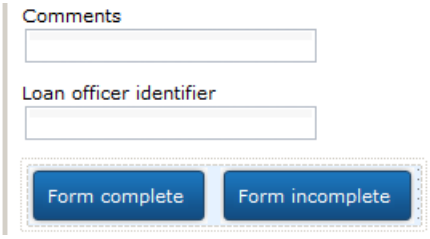
__t. Next you will bind the applicationView to the application process variable. Click the **applicantionView** component you just dropped. On the **Properties** view on **General** under **Behavior**, locate **Binding** and click **Select**. Then double-click the **mortgageData Output** variable.



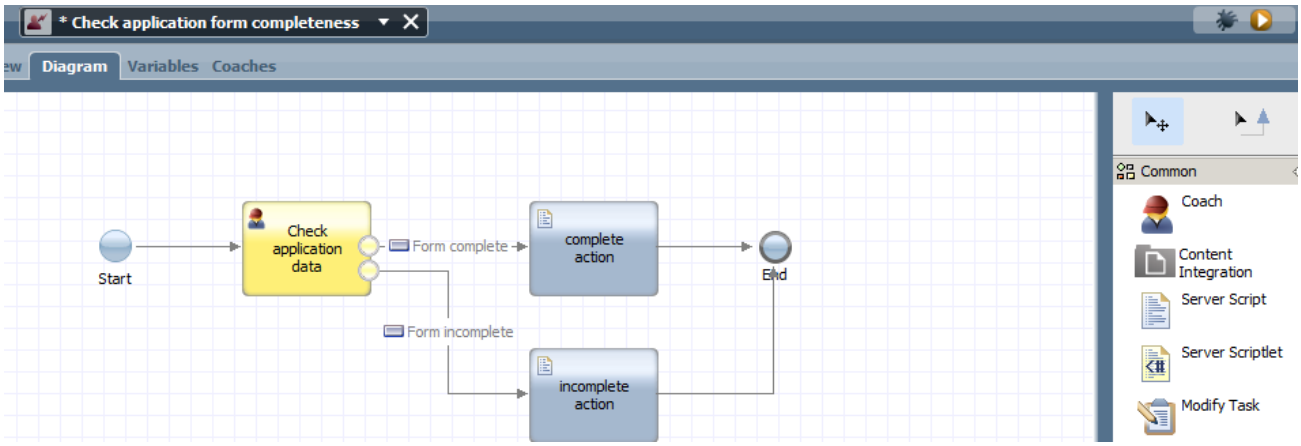
__u. Drag the administrationInfo to the bottom of the coach. Delete the eligibility Boolean as we'll set it using buttons instead.



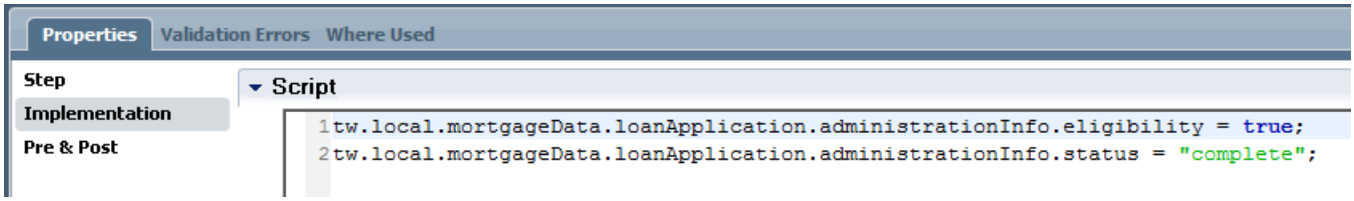
__v. Create a horizontal section without title to hold the two action buttons: "Form complete" and "Form incomplete"



__w. Switch to the diagram tab of the coach and drag two server script activities onto the canvas, wire them to the Check application data and the End, delete the current OK wire.

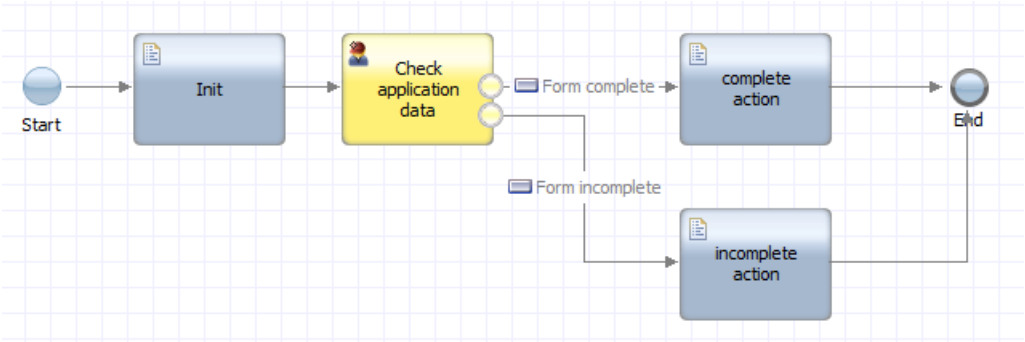


__x. Click on complete action, select the Implementation section of the properties tab and enter the following script.
tw.local.mortgageData.loanApplication.administrationInfo.eligibility = true;
tw.local.mortgageData.loanApplication.administrationInfo.status = "complete";

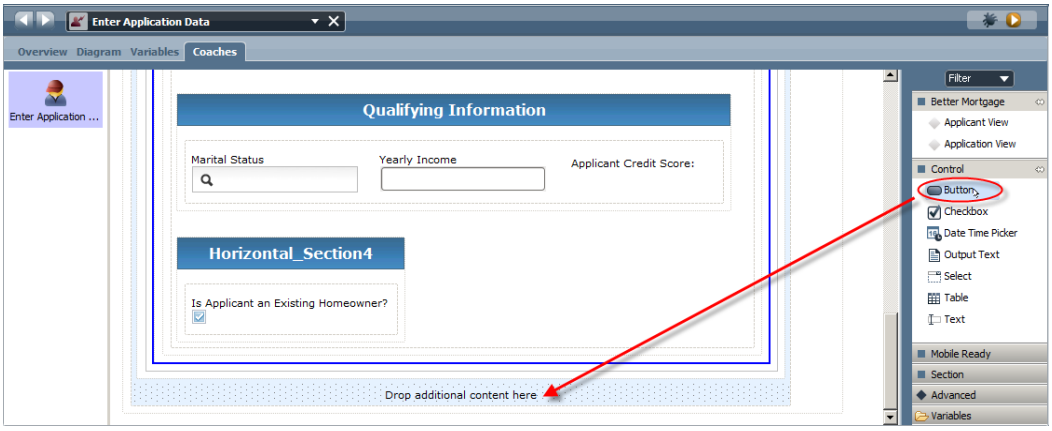


__y. Click on incomplete action and enter the following data:
tw.local.mortgageData.loanApplication.administrationInfo.eligibility = false;
tw.local.mortgageData.loanApplication.administrationInfo.status = "incomplete";

__z. Drag another server script element onto the canvas and wire it to the 'start' and 'Check application data' activity. Call it 'Init'.



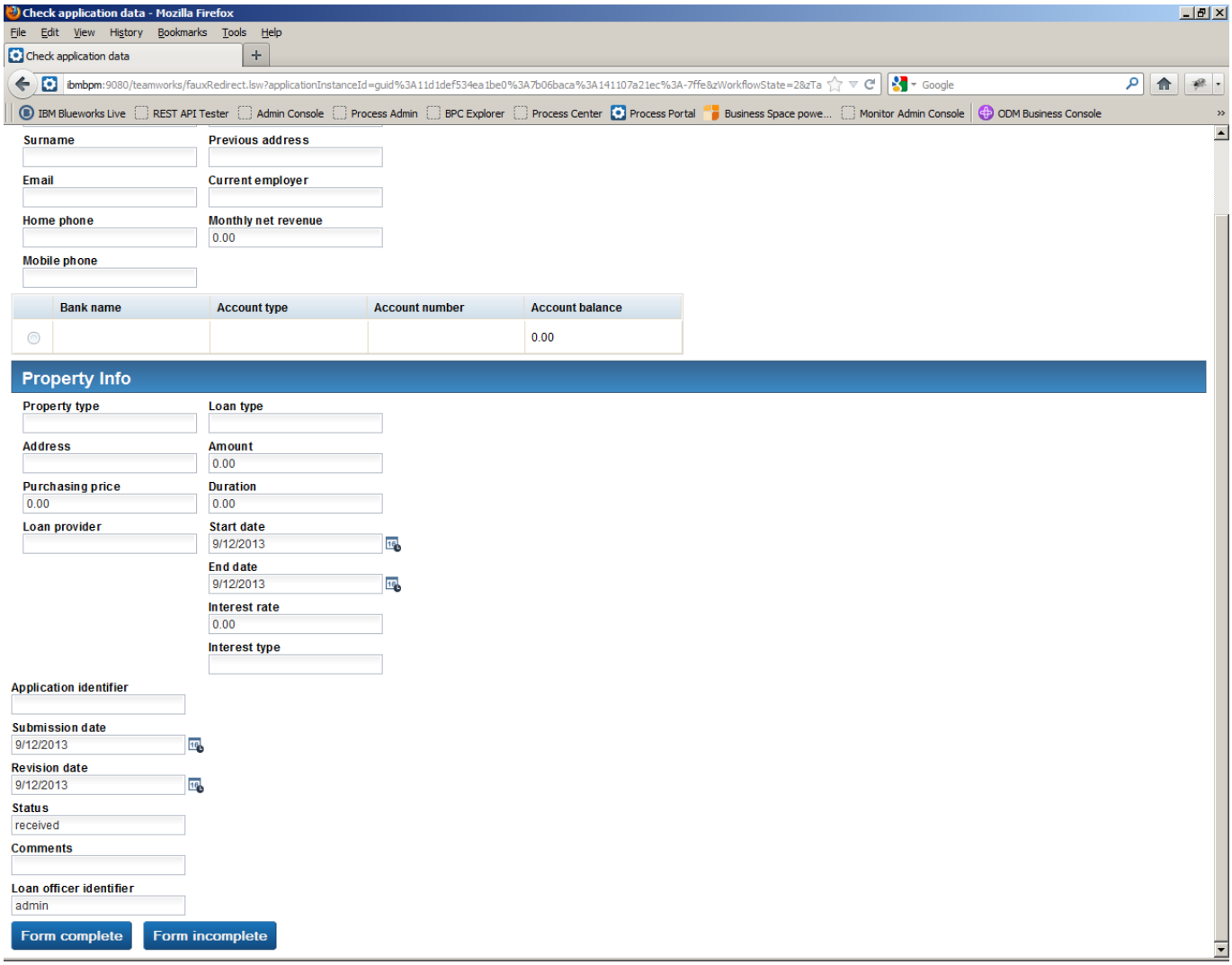
__aa. Enter the following script.
tw.local.mortgageData.loanApplication.administrationInfo.applicationIdentifier = tw.system.currentProcessInstanceID;
tw.local.mortgageData.loanApplication.administrationInfo.status = "received";
tw.local.mortgageData.loanApplication.administrationInfo.revisionDate = new TWDate();
tw.local.mortgageData.loanApplication.administrationInfo.loanOfficerIdentifier = tw.system.user_loginName;



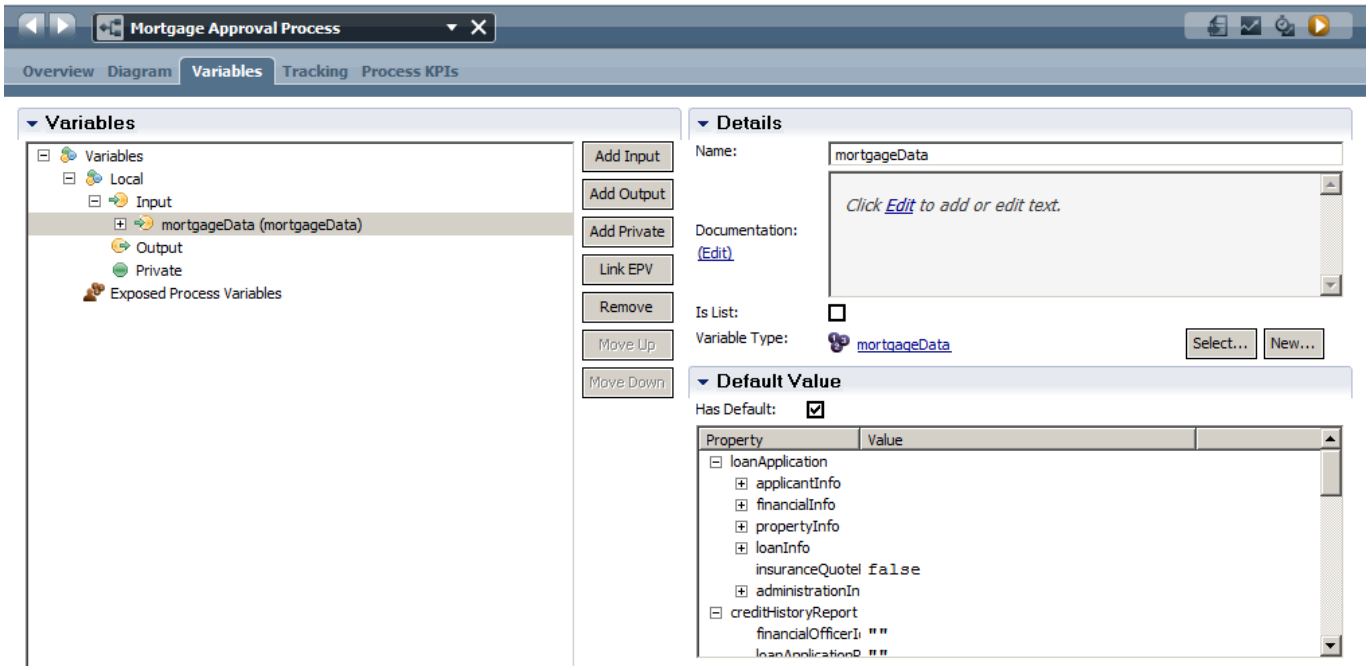
__19. Test the Coach

__a. Now we are ready to test the coach we have just built. Just as before, when we wanted to “play back” a process, we can do the same for a coach (as well as other types of services within IBM BPM). Press **Play** on the upper right to play back this Coach in the browser.

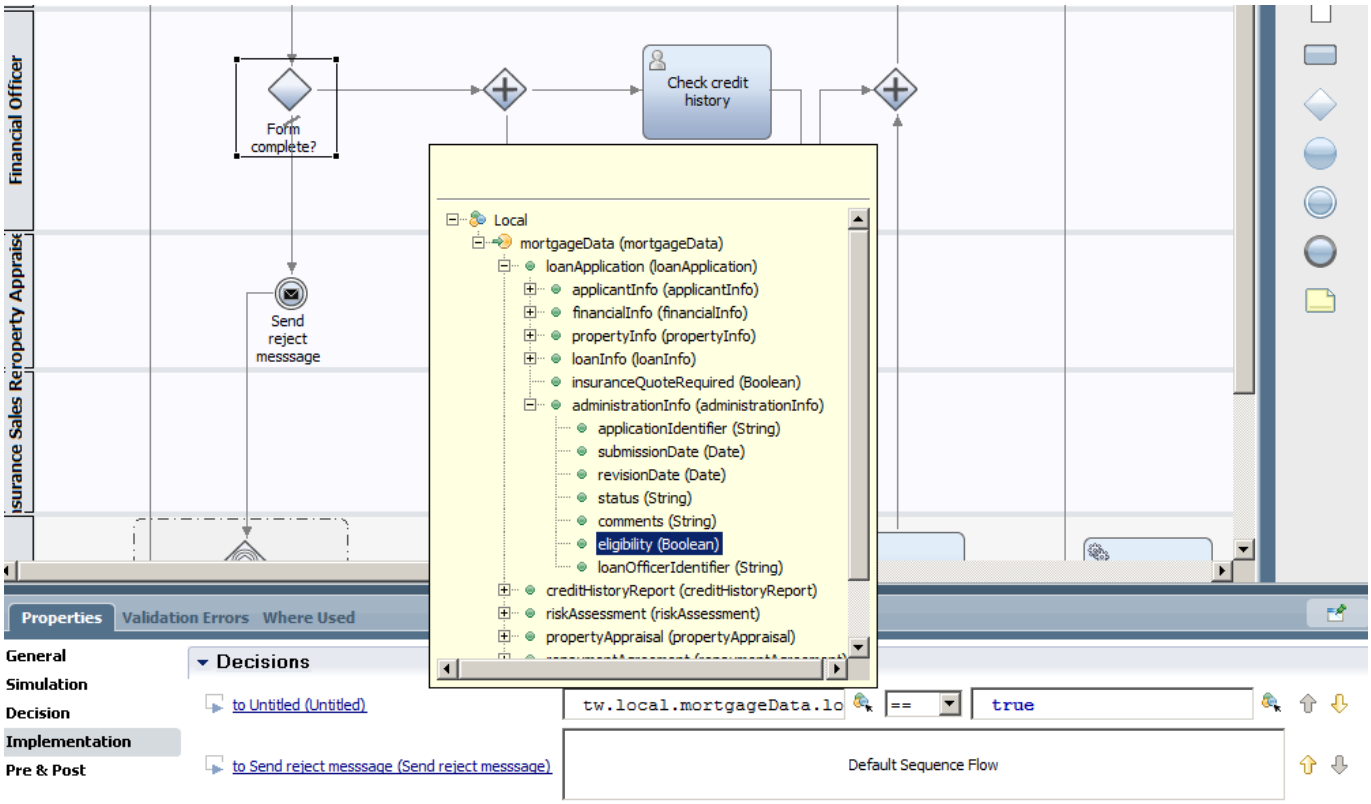
__b. You should see the browser launch and the coach preview into the browser. Notice some of the system data. Enter in some data into both tabs, and click **Submit**. This will close the browser


Check application data - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Check application data
ibmbpm:9080/teamworks/fauxRedirect.lsw?applicationInstanceId=guid%3A11d1def534ea1be0%3A7b06baca%3A141107a21ec%3A-7ffe&zWorkflowState=2&zTa
Google
IBM Blueworks Live
REST API Tester
Admin Console
Process Admin
BPC Explorer
Process Center
Process Portal
Business Space powe...
Monitor Admin Console
ODM Business Console
Surname
Previous address
Email
Current employer
Home phone
Monthly net revenue
0.00
Mobile phone
Bank name
Account type
Account number
Account balance
0.00
Property Info
Property type
Loan type
Address
Amount
0.00
Purchasing price
0.00
Duration
0.00
Loan provider
Start date
9/12/2013
End date
9/12/2013
Interest rate
0.00
Interest type
Application identifier
Submission date
9/12/2013
Revision date
9/12/2013
Status
received
Comments
Loan officer identifier
admin
Form complete
Form incomplete

__20. **Add initialization data to the process:**
Switch to the process view and click the variables tab. Tick the 'Has Default' checkbox to ensure that all variables are properly initialized. The same principle applies when you create a coach, the default values in coach are overwritten when the process is played back instead of a standalone coach.



__a. Select the 'Form complete' decision point and view the Implementation tab in its properties view. Set the eligibility Boolean to true as follows:

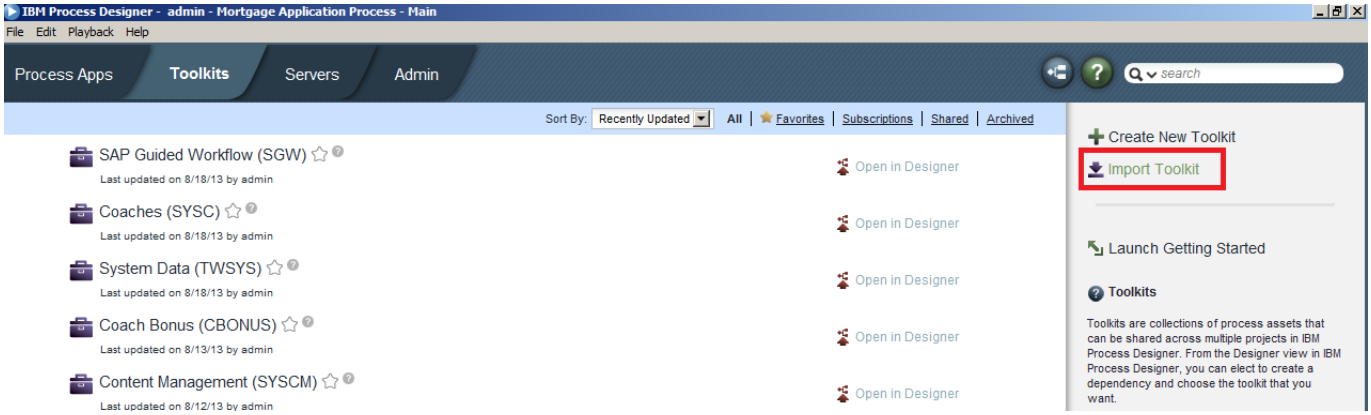


__b. Test the decision logic by playing back the process.

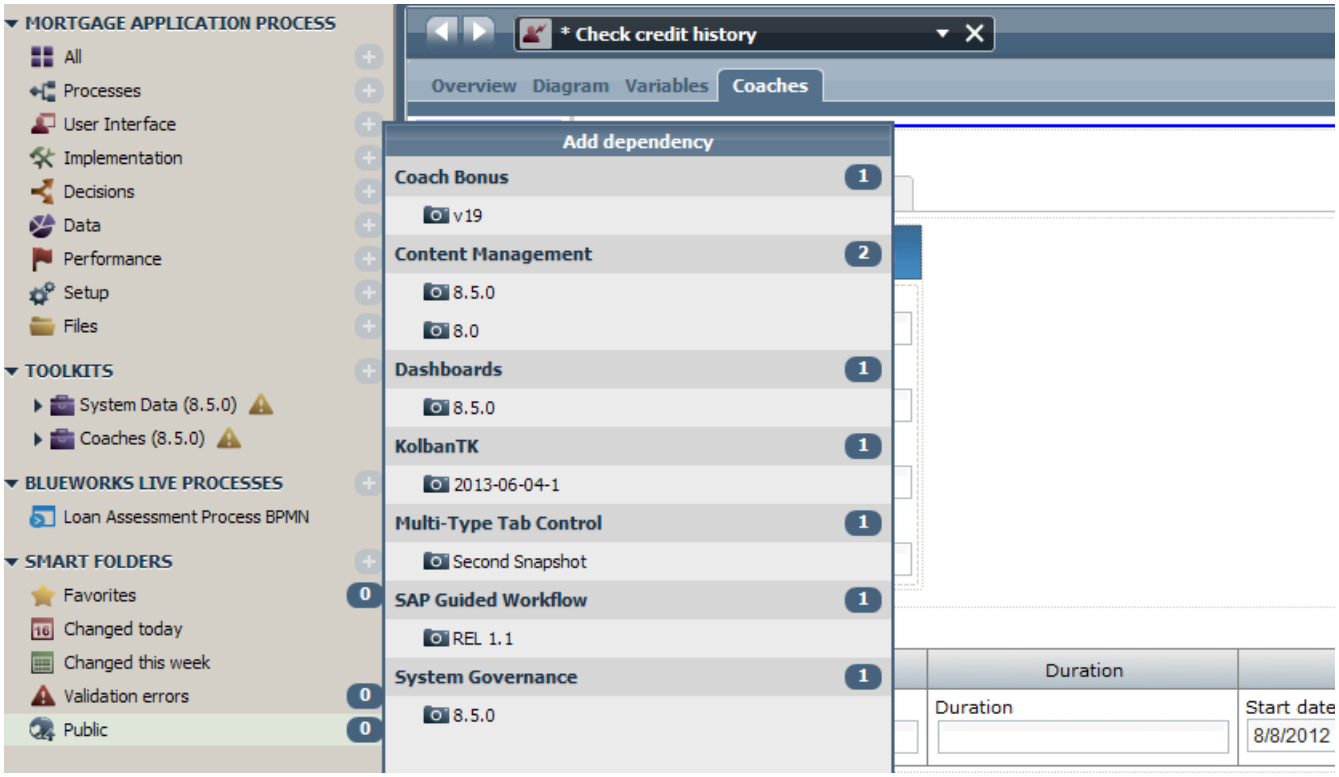
__21. **Importing Toolkits into the project:**
For this coach we want to use an element that does not exist in the standard palette. IBM BPM uses the principle of toolkits which are re-usable libraries that contain processes, coach, coach views, rules, data and palette controls. Basically anything that is available in a normal project. Toolkits are extremely powerful as they allow the managed re-use and sharing of components.

__a. Import the following toolkit into the Process Center by switching to the Process Center view's Toolkit tab.

Snapshot: Multi-tab control: Multi-Type_Tab_Control - Second_Snapshot.twx



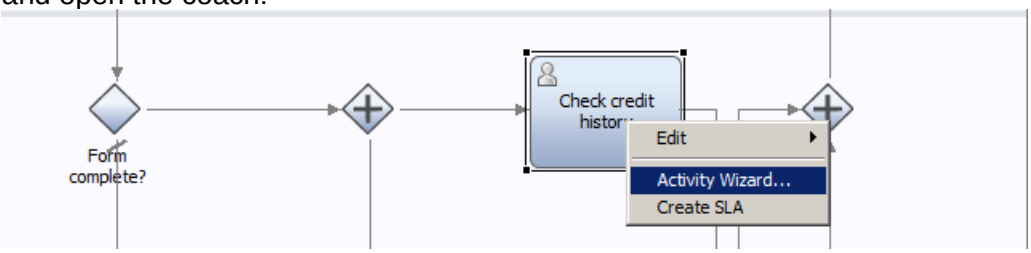
__b. After the import has been completed, open the Mortgage Application Process again and click on the '+' on the Toolkits menu option:



__c. Select 'Second Snapshot' of the Multi-Type Tab control.

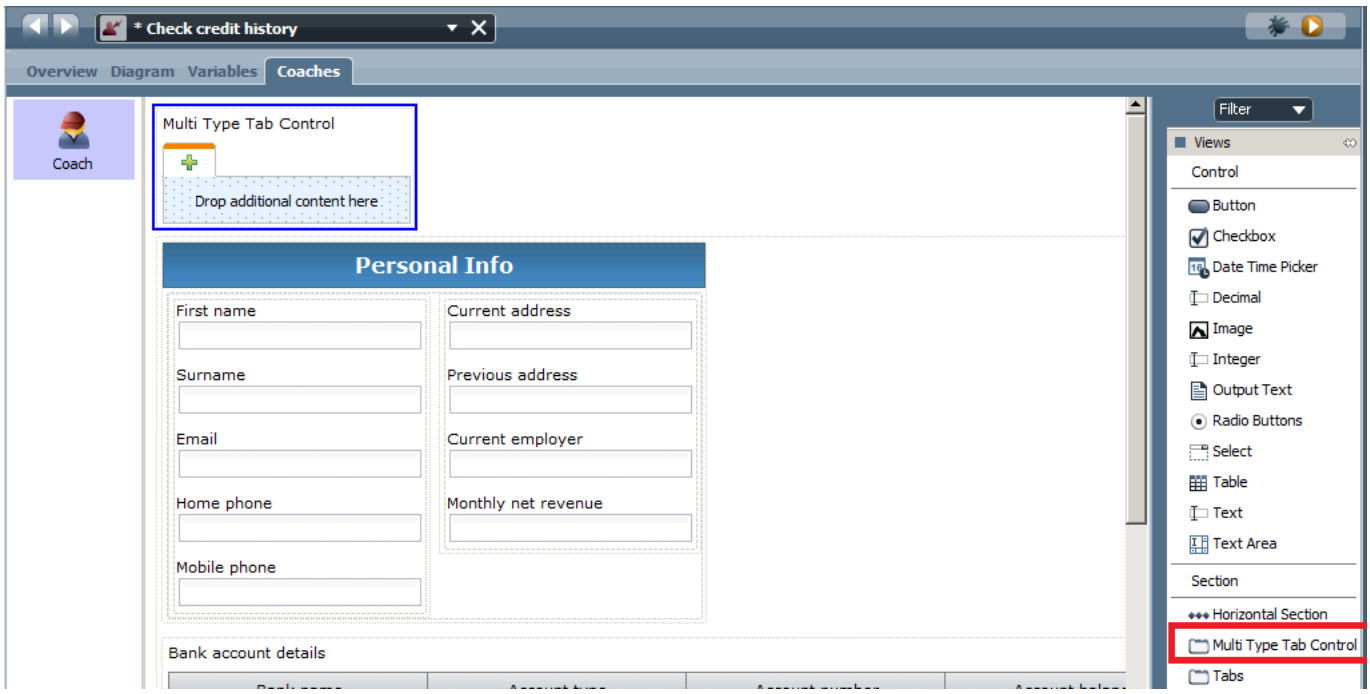
__d. Ignore the version dependency warning.

__e. Change the view to the mortgage application process and create a new coach by right-clicking Check credit history and selecting 'Activity Wizard', choose the default options and open the coach:

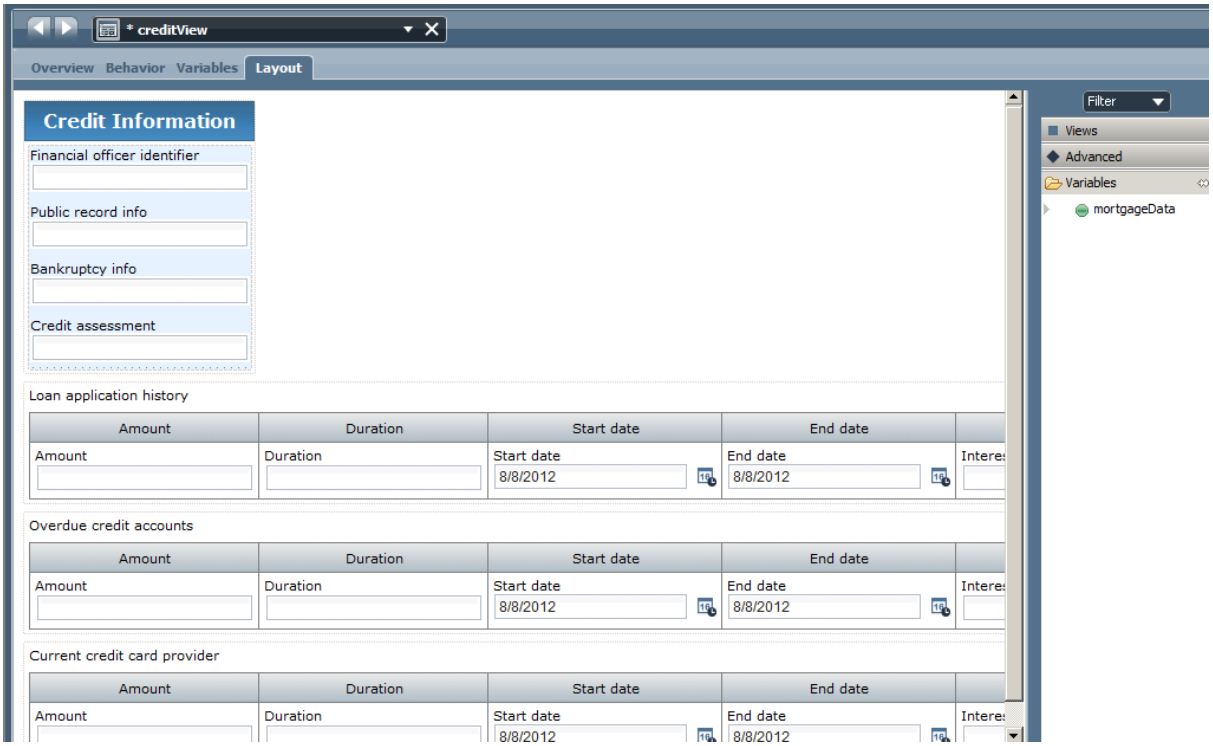


__f. As the in/out parameters of this coach are of mortgageData, it auto populates the coach with the applicationView coach view that is of type mortgageData. The application data is used as reference data for this person so we want to show it in a tab on the screen.

__22. Drag a 'Multi TypeTab control' section to above the coach view, then drag the applicationView view into the first tab, bind the applicationView coach view to mortgageData:

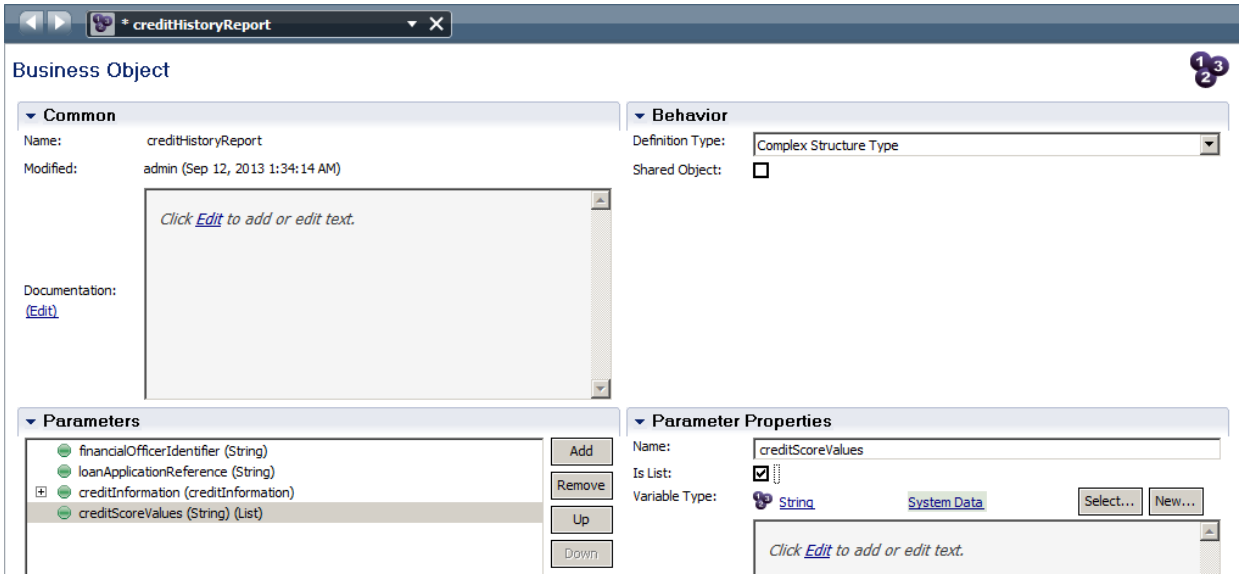


__a. Now we'll create a new coach view to contain the Credit check info. Using the same structure as in the applicationView we'll build up the creditView. Ensure to set the tag to 'mortgageViews' and bind the 'mortgageData' variable. There's no need to show the loanApplicationReference variable as all data is in a single object.

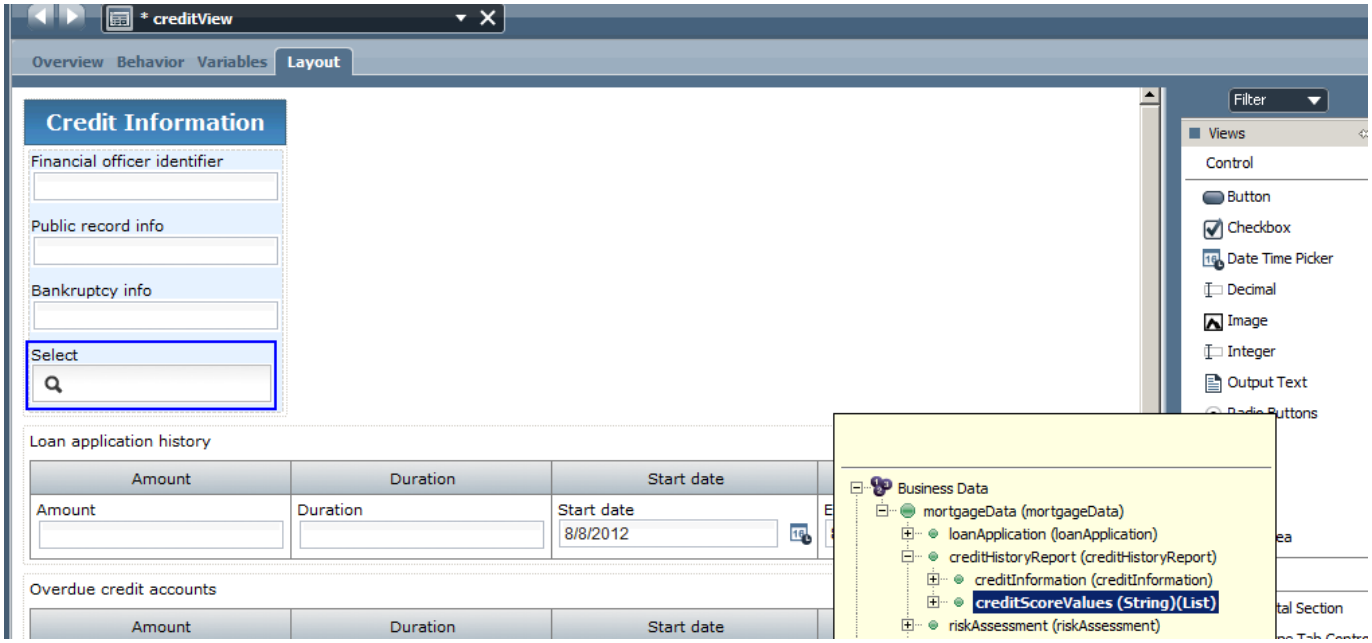


__b. The Credit assessment text input box should in fact be a drop down menu with allowable choices. We can create a prepopulated drop down menu as follows:

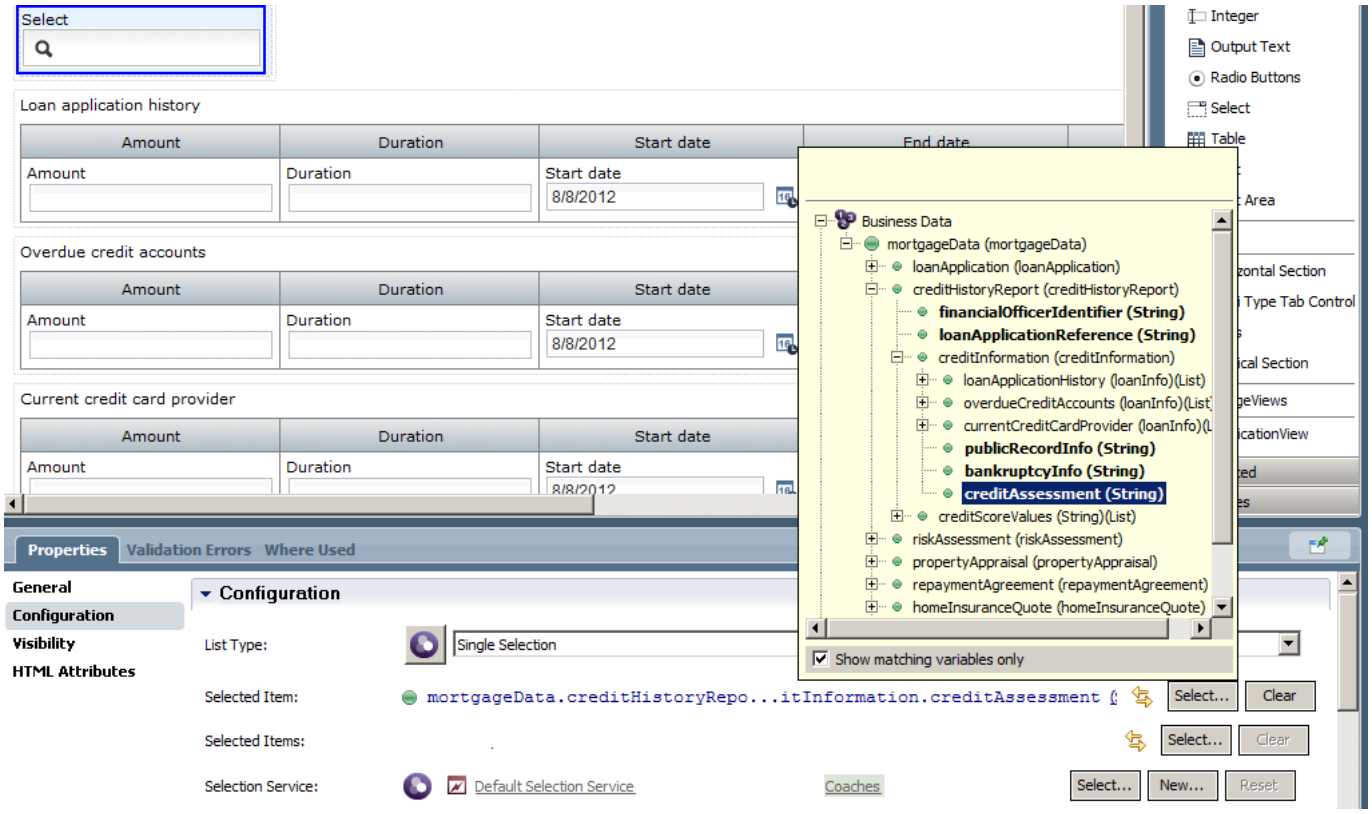
__c. Add a new variable (String, List) to the creditHistoryReport object called creditScoreValues:



__d. Switch to the creditView coach view and delete Credit assessment, drag a Select object into the canvas. Bind it to the creditScoreValues List object that you've just created.

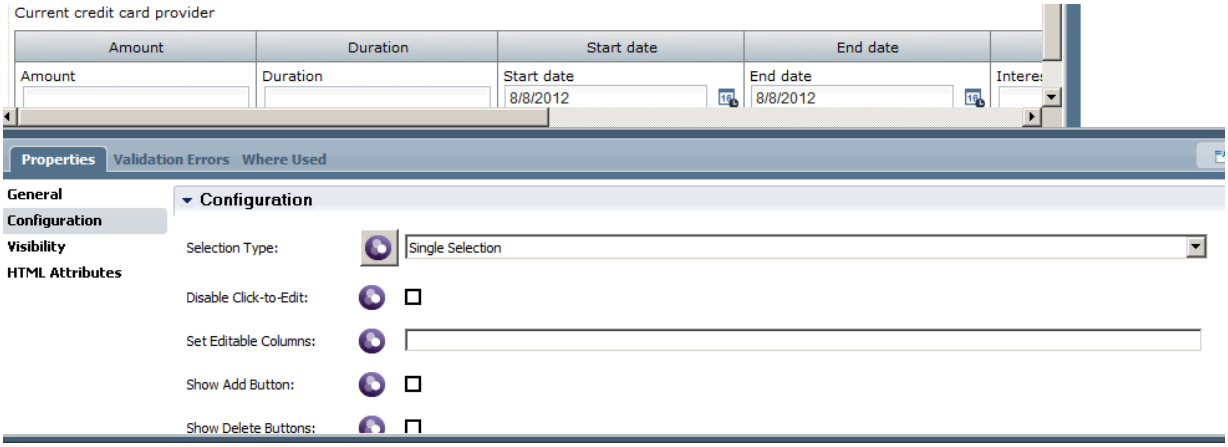


__e. In the configurations tab of the Select object, bind the 'Selected Item' entry to the creditAssessment String:

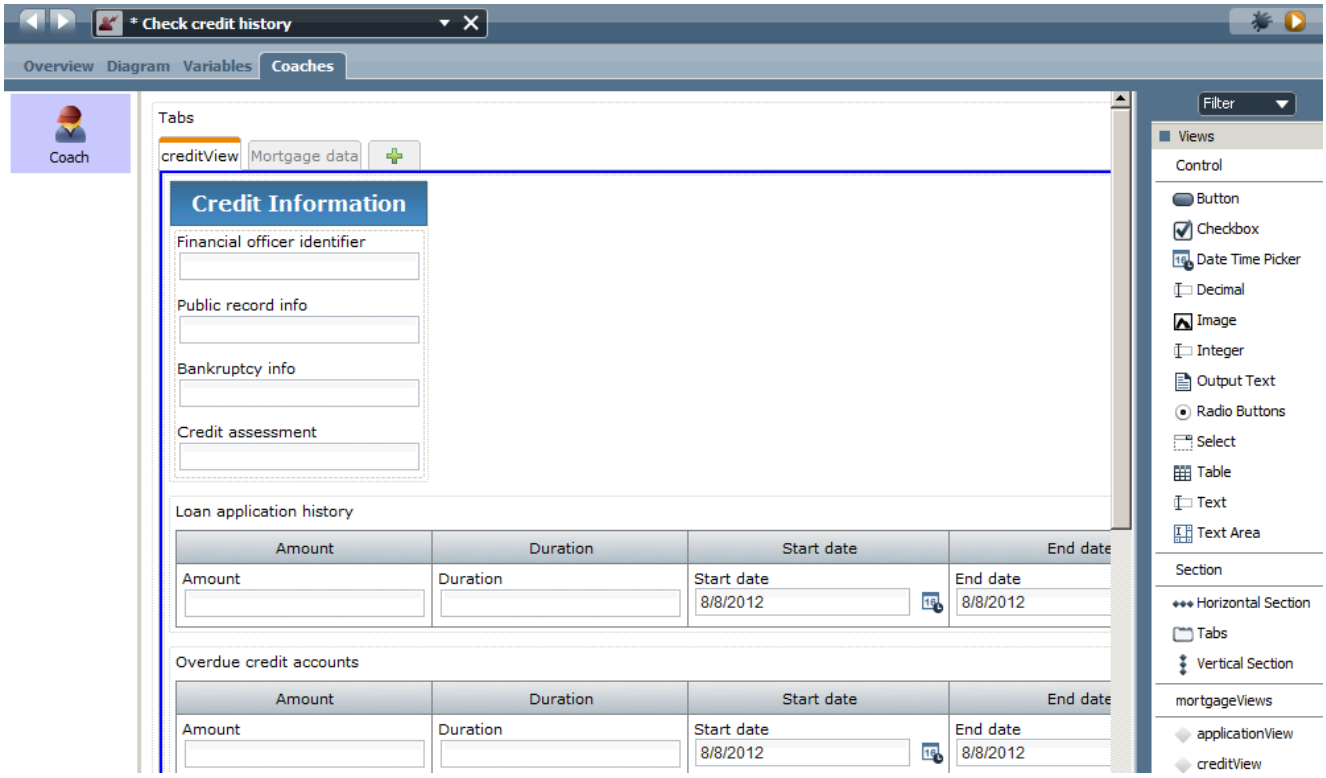


__f. Change the name of the 'Select' element to Credit Assessment.

__g. In the coach view you can see that the loan application history is depicted as a set of tables that allows us to enter multiple rows of data for each entry. We need to enable actions on the table that allow us to control the allowable actions for the table. Click on the 'Configuration' tab of a table property:

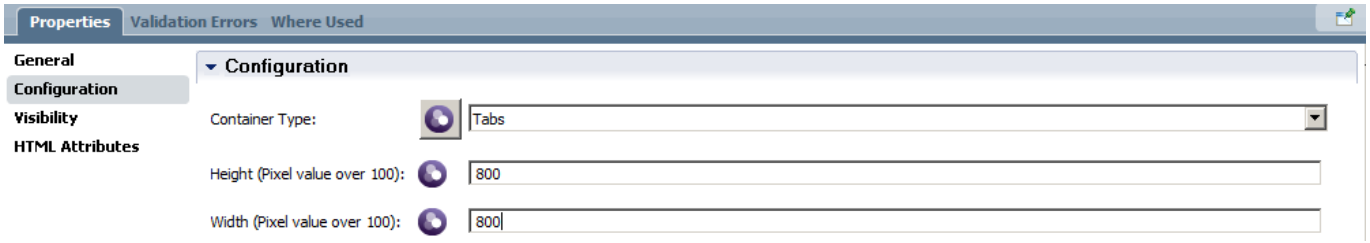


__h. Check the 'Add' and 'Delete' buttons. Save and switch back to the Check Credit history tab. Drag the new creditView into a new tab as the first option and bind the view to mortgageData.

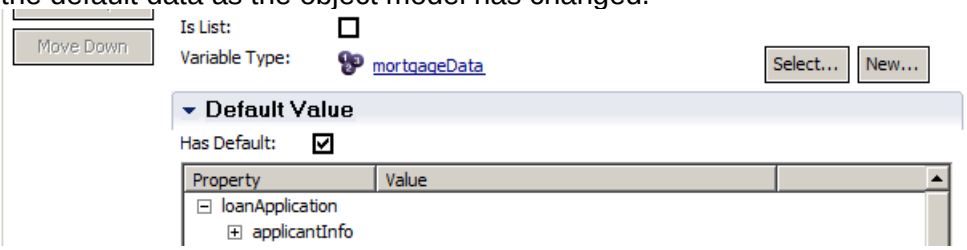


__i. change the containerType to 'Tabs' in the properties of the Tabs element and change the size to 800 by 800:

__j. Change the label of the submit button from 'Button' to 'Submit Credit Assessment'.

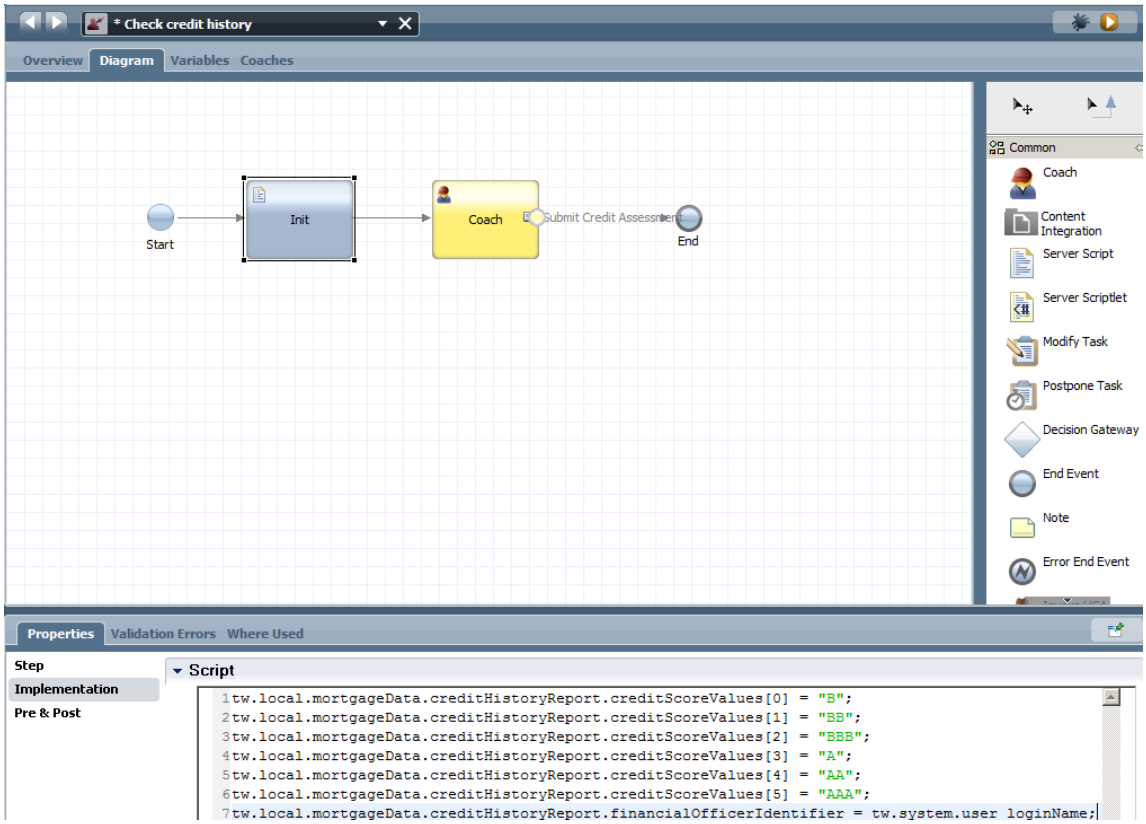


__k. Switch to the variables tab and uncheck/recheck the 'Default Value' checkbox to refresh the default data as the object model has changed.

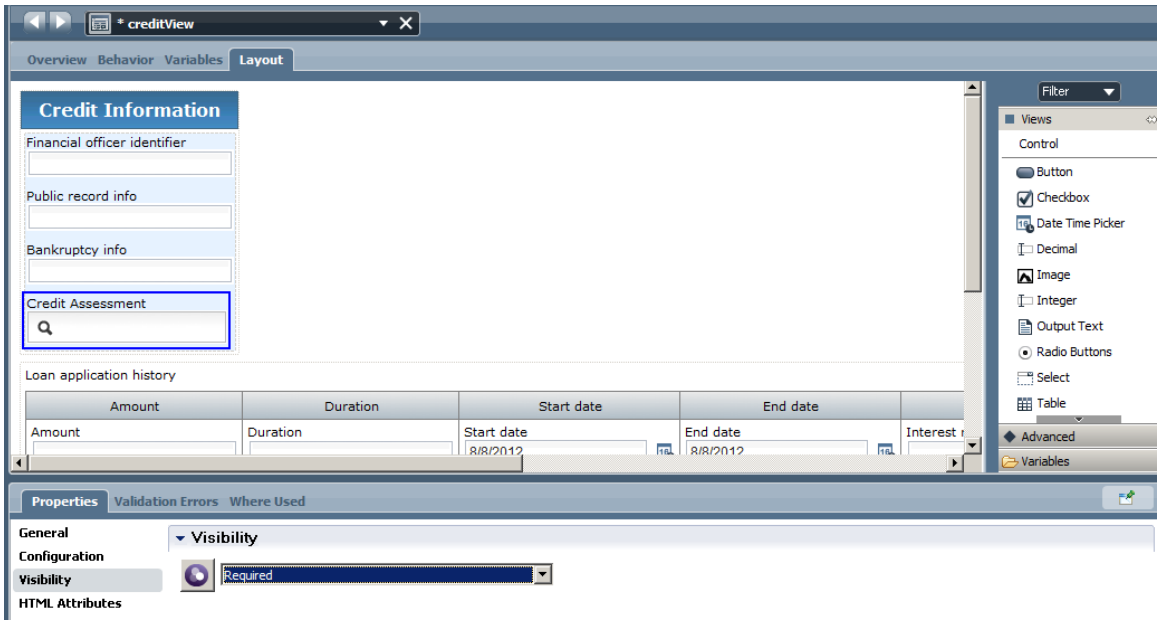


__l. Switch to the Diagram view and add a new server script activity before the coach, name it 'Init'. Add the following code to the Implementation tab:

```
tw.local.mortgageData.creditHistoryReport.creditScoreValues[0] = "B";
tw.local.mortgageData.creditHistoryReport.creditScoreValues[1] = "BB";
tw.local.mortgageData.creditHistoryReport.creditScoreValues[2] = "BBB";
tw.local.mortgageData.creditHistoryReport.creditScoreValues[3] = "A";
tw.local.mortgageData.creditHistoryReport.creditScoreValues[4] = "AA";
tw.local.mortgageData.creditHistoryReport.creditScoreValues[5] = "AAA";
tw.local.mortgageData.creditHistoryReport.financialOfficerIdentifier =
tw.system.user_loginName;
```



__m. The dropdown menu now has pre-populated values for the credit scores. Make this field mandatory by changing the visibility settings to "Required"



Press **Save** to save your work and play back the coach. Experiment with the tabs and table controls.

2.4.3 Creating the appraise Property coach

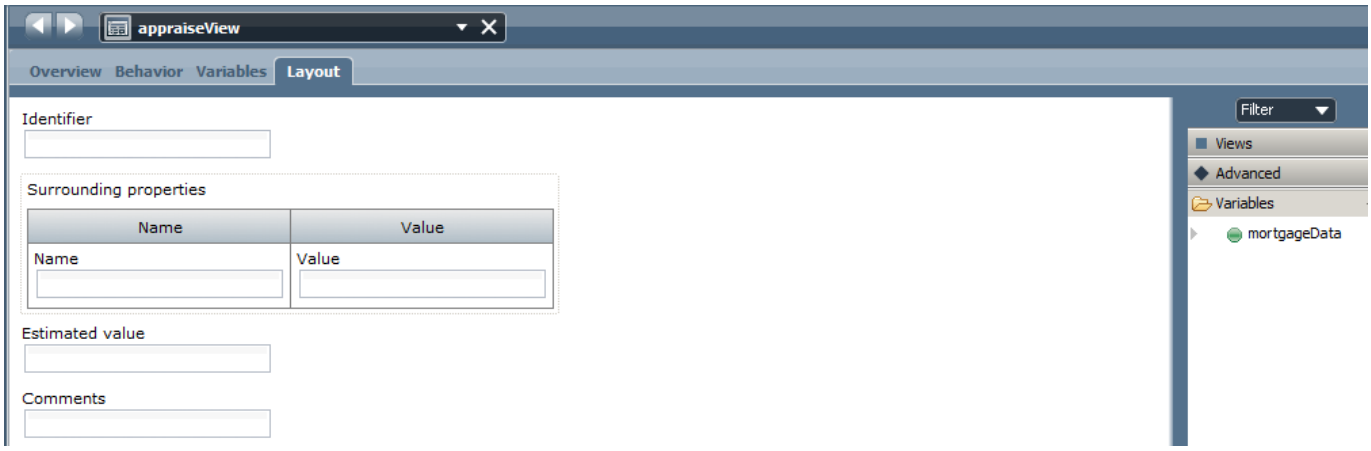
Human Services

Notice that so far every human service we have created has been simple – a Start, a single coach, and an End. It is possible, and very useful, to have multiple steps in a human service. This helps us create multi-screen flows as well as screens that can integrate with calls to external systems.

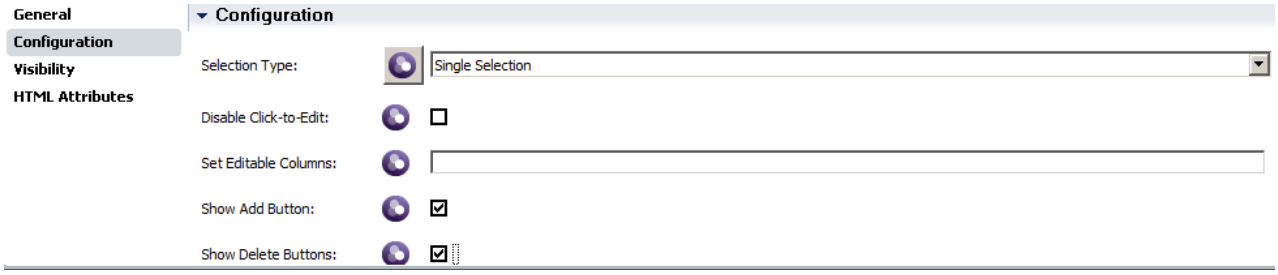
__23. The appraise property task will be created in similar fashion to the Check credit history task. Create a new coach View called appraiseView

__a. Tag it as part of the group mortgageViews

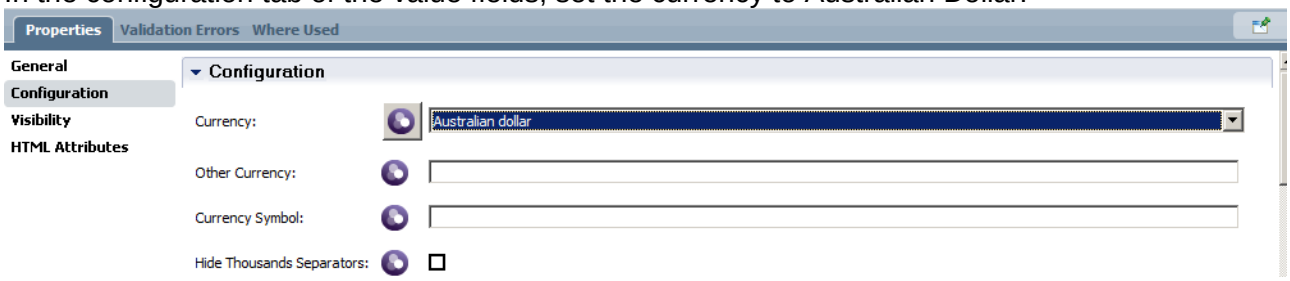
__b. set the business data to bind to mortgageData, then drag the whole propertyAppraisal variable object onto the canvas, delete the loan application reference field.



__24. Add 'Add' and 'Delete' functionality to the surrounding properties table

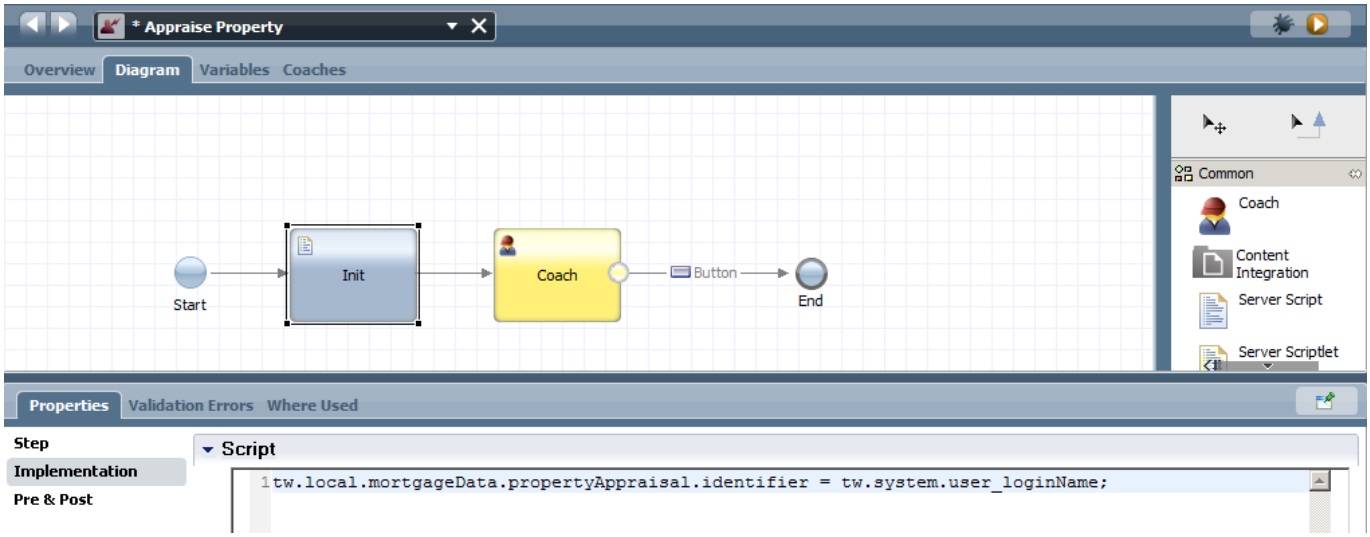


__25. In the configuration tab of the value fields, set the currency to Australian Dollar:

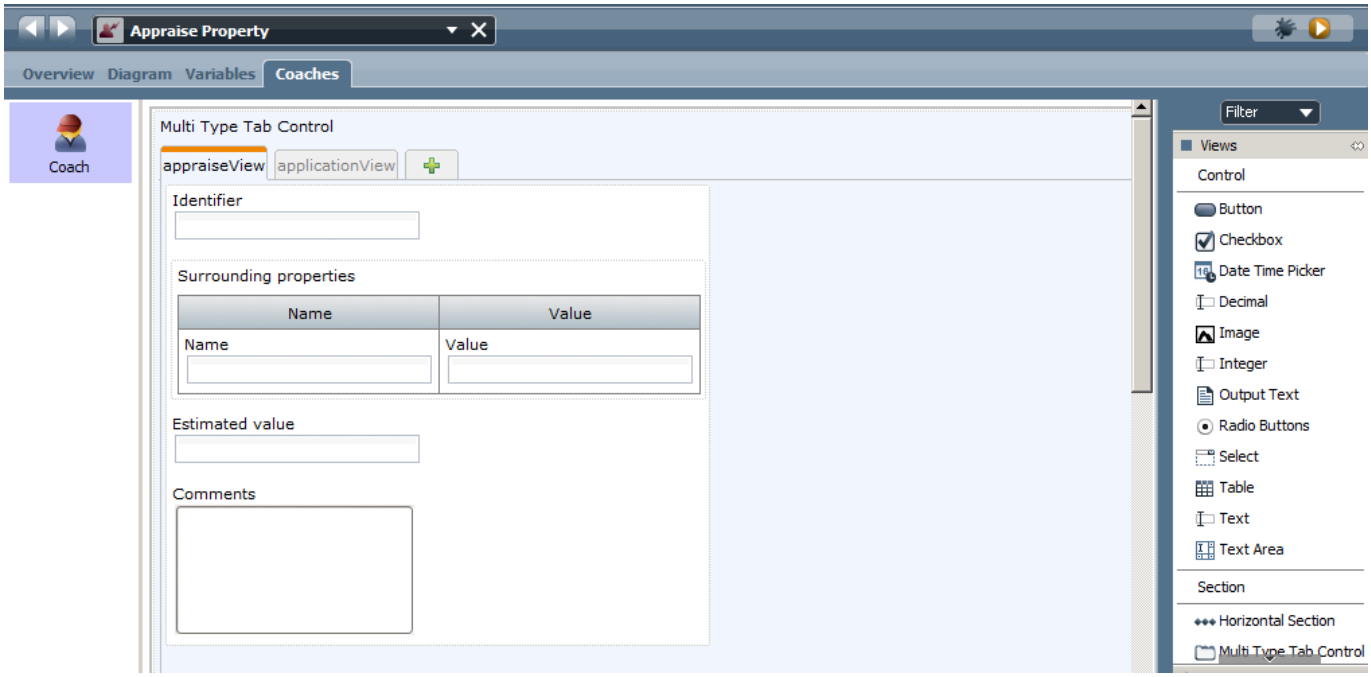


__26. Change the view type of Comments to Text area instead of Text.

__27. Create a new coach called Appraise Property based on its task. Add a new Init server script to the coach diagram as follows:



__28. Create a new Multi Type Tab control onto the canvas and drag the appraisalView and applicationView onto it. Change its type from accordion to tab view. Change the submit button label to 'Submit Appraisal Result'. Save and test.

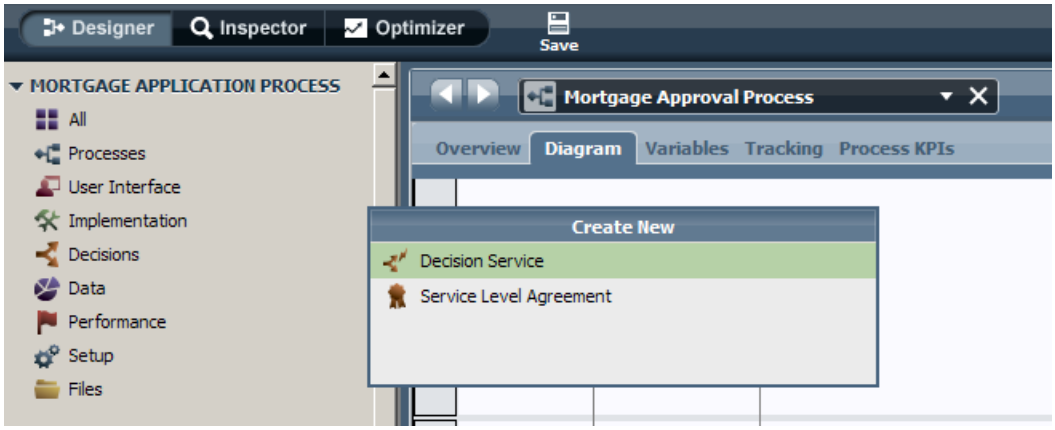

Appraise Property
Overview
Diagram
Variables
Coaches
Coach
Multi Type Tab Control
appraiseView
applicationView
Identifier
Surrounding properties
Name
Value
Name
Value
Estimated value
Comments
Filter
Views
Control
Button
Checkbox
Date Time Picker
Decimal
Image
Integer
Output Text
Radio Buttons
Select
Table
Text
Text Area
Section
Horizontal Section
Multi Type Tab Control

2.5 Adding Decisions to the process

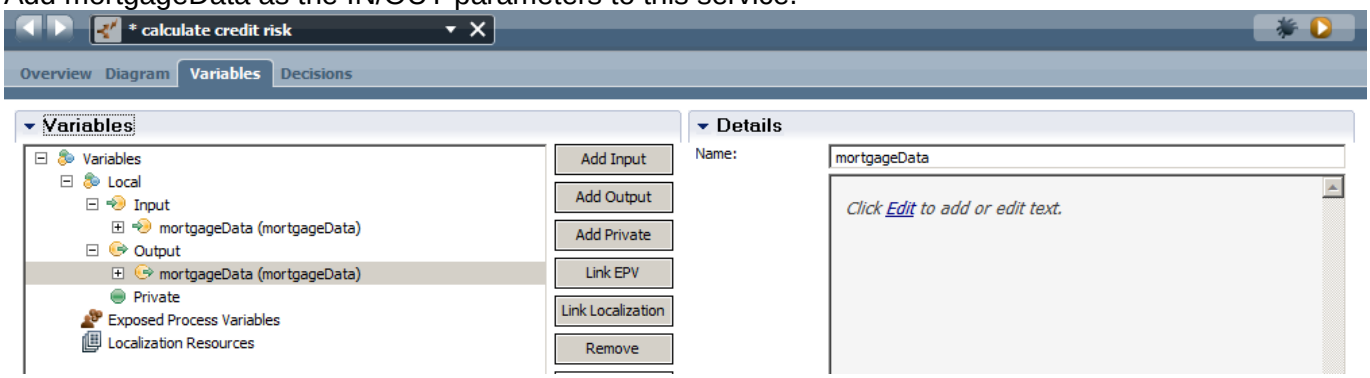
We will add a simulated credit check step as part of the Assess Loan Risk activity so that the Loan Officer will have this information to make their decision.

2.5.1 Add Calculate Credit Risk to Enter Application Data

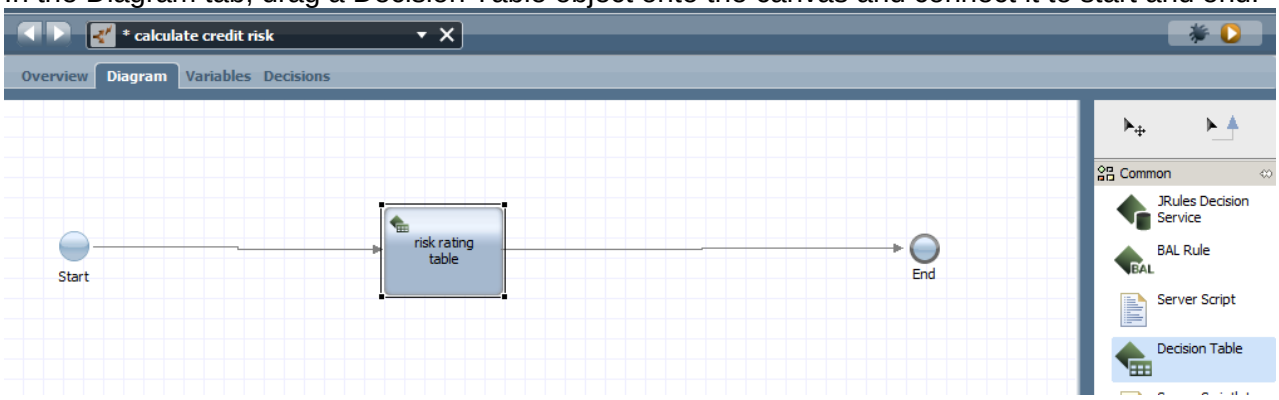
__29. In the menu click the '+' sign next to Decisions and select Decision Service, name the service 'calculate credit risk'



__30. Add mortgageData as the IN/OUT parameters to this service.

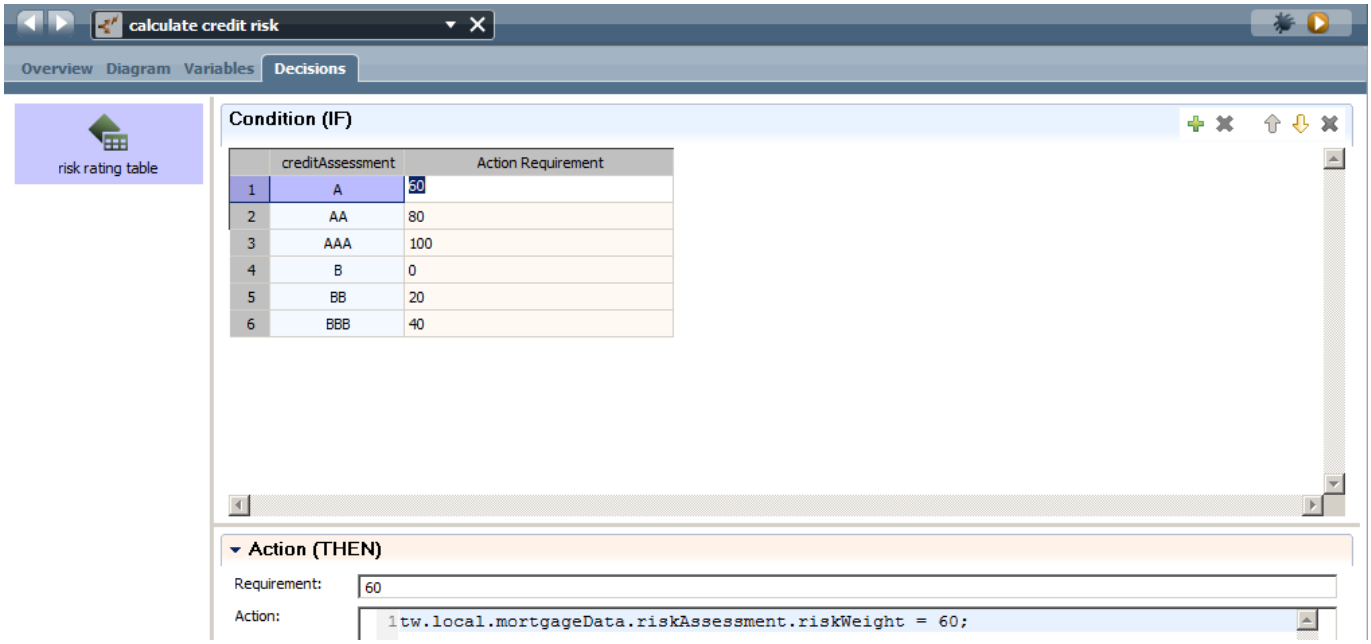


__31. In the Diagram tab, drag a Decision Table object onto the canvas and connect it to start and end.

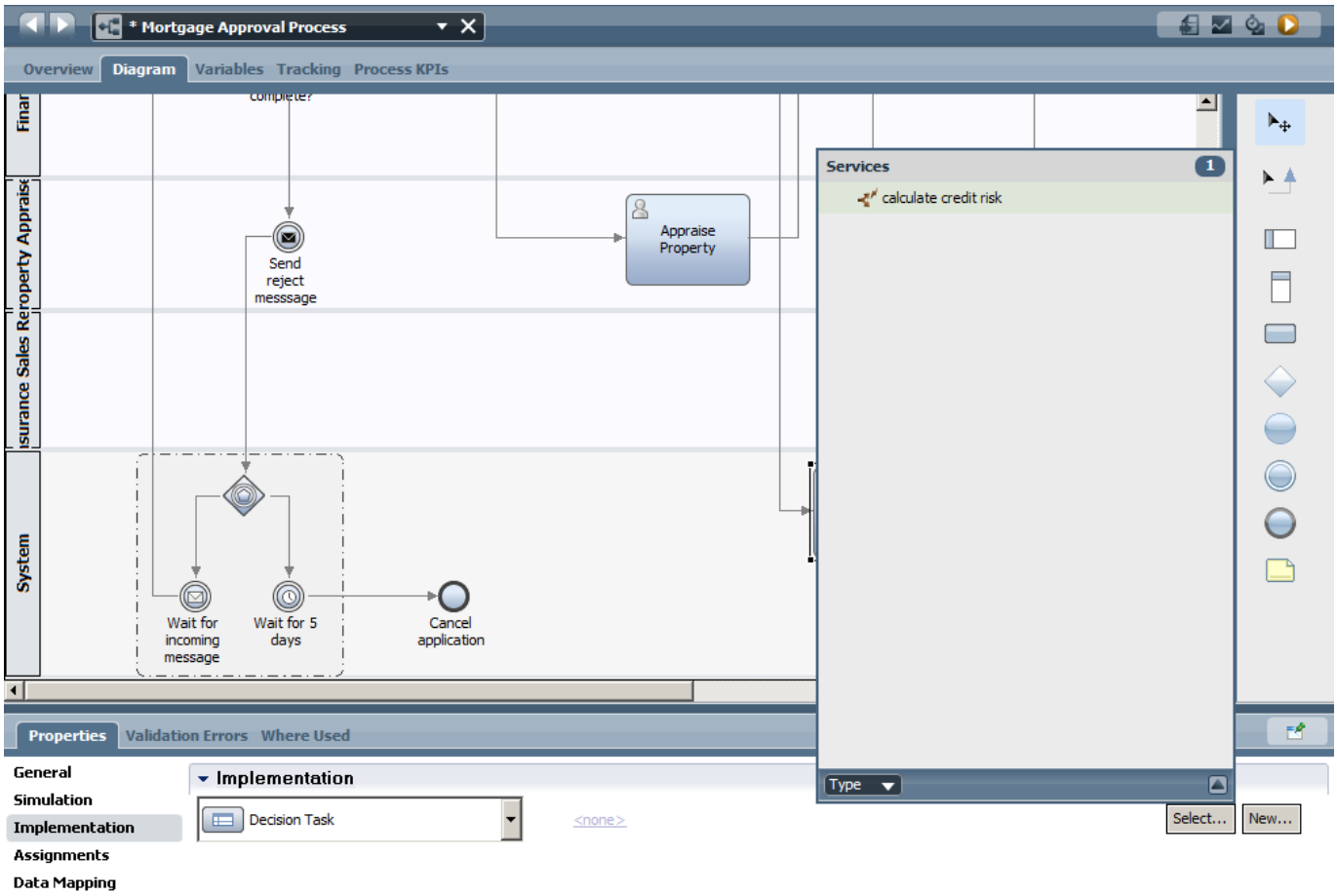


__32. Doubleclick on the risk rating table activity. Click on the green '+' sign on the right to add a new condition column. Select the creditAssessment variable as part of creditInformation as the condition.

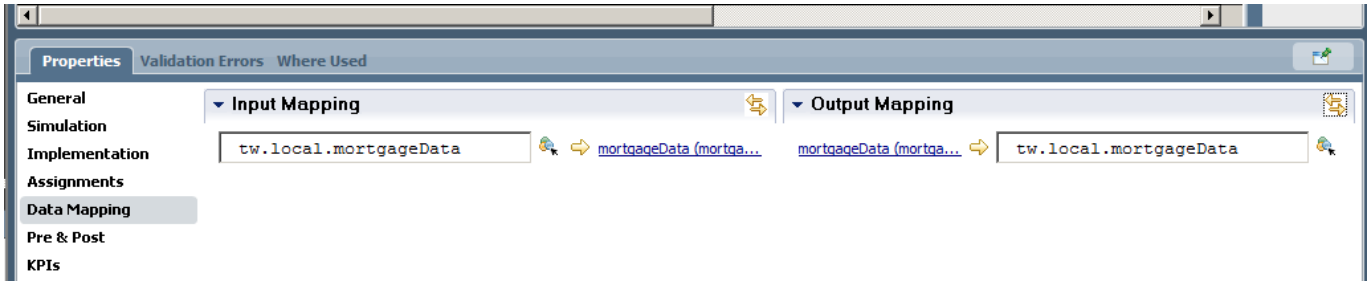
__33. Click on the row number to add a new line. Add the following data to the risk table:



__34. Save and close the service. In the process design screen, click on the Assess Loan Risk task. Change to the implementation tab in the properties view and change the Implementation type to Decision Task and then press on 'Select…' and choose the Calculate credit risk decision service just implemented:



__35. Map the IN/OUT parameters of the service to mortgageData:

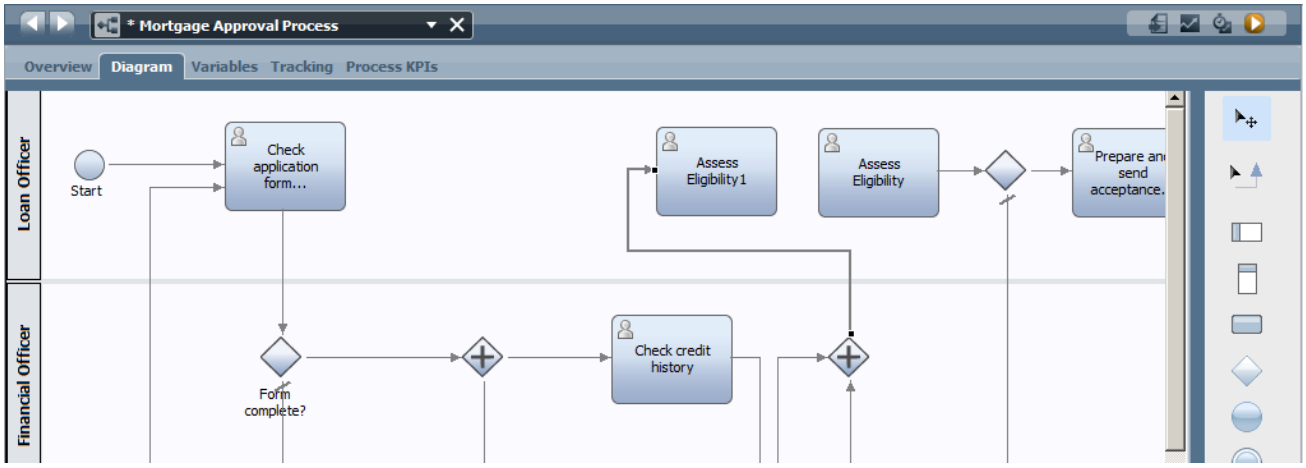


2.5.2 Creating the Assess Eligibility coach

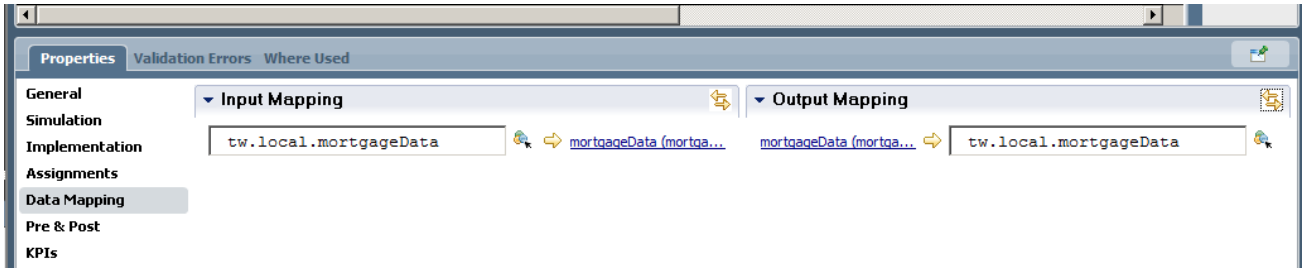
In order to speed up development of the Assess Eligibility coach we'll simply duplicate the appraise property coach.

__36. From the **Process** diagram, right click the Appraise Property coach and select 'Duplicate…'. Drag the new coach onto the canvas. Connect the path from the original assess eligibility task to the new one and delete the old one.

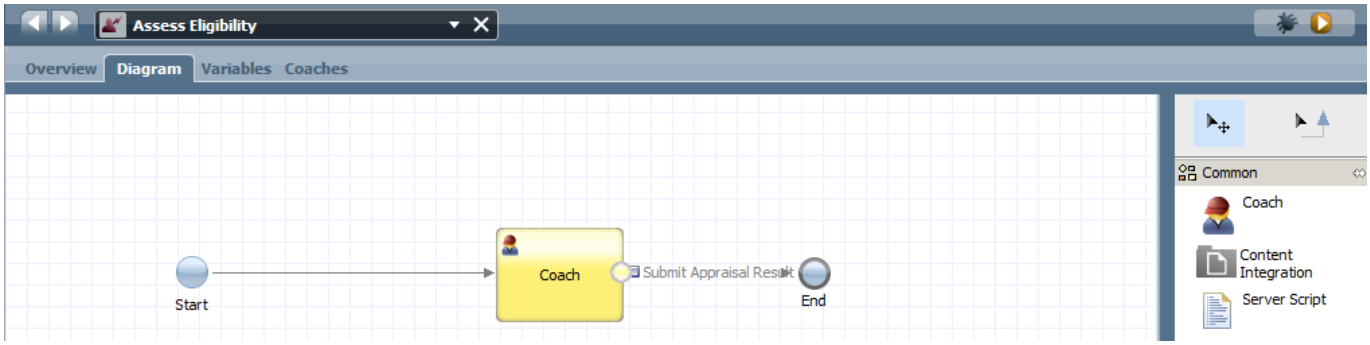
__37.



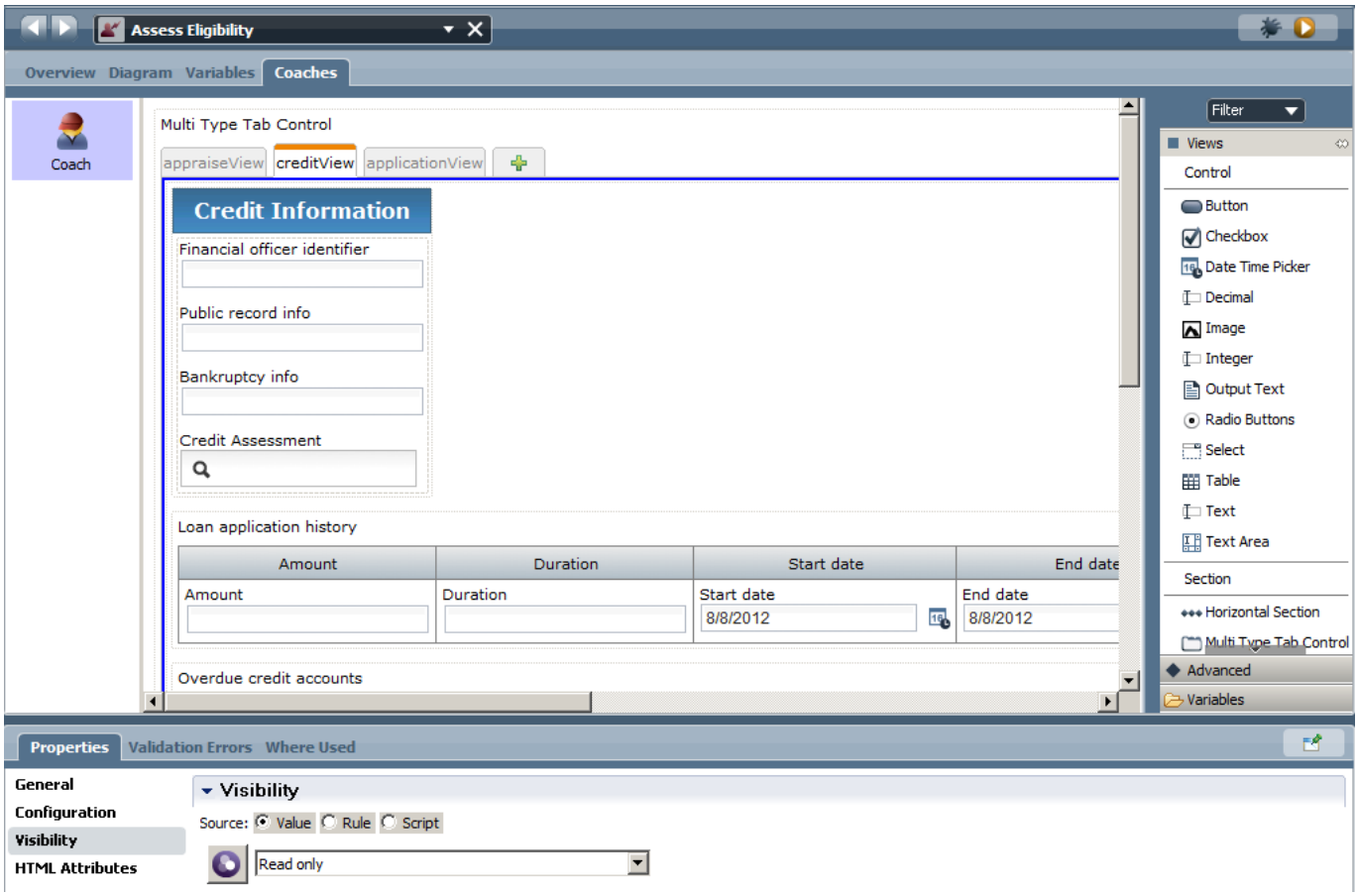
__38. Map the IN/OUT parameters of the coach to mortgageData:



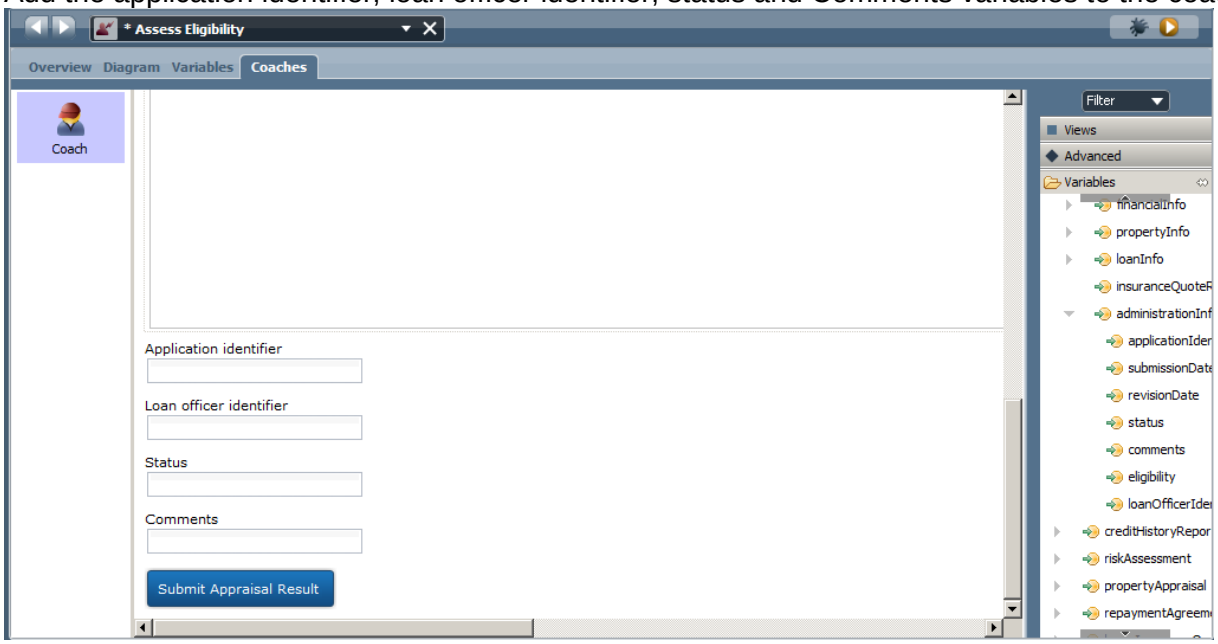
Delete the Init task from the Diagram view and double click on the coach to open it.



__39. Add the creditView to the tab and in the visibility option, select 'read only' to make the entire coachView read only. Do the same for the applicationView and appraiseView tabs. Alternatively you can set the entire Multi Type Tab Control to be read-only.

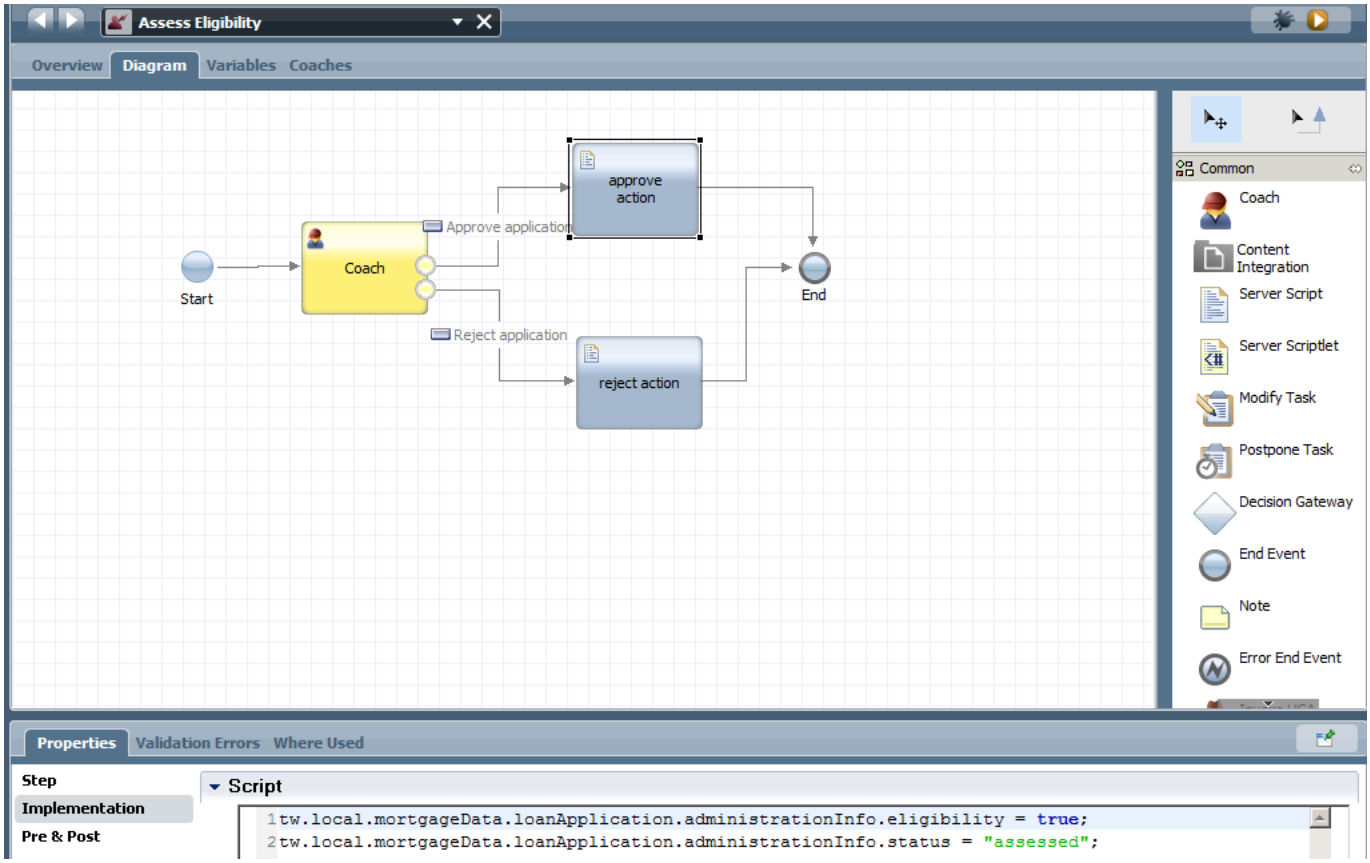


__40. Add the application identifier, loan officer identifier, status and Comments variables to the coach.

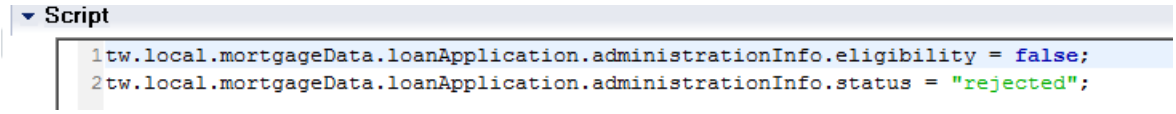


__41. Change the Comments from text to text area type. Add two buttons at the bottom with the following labels:

__42. In the Diagram view, add two server scripts to the canvas, connect them and add the following script to the activities:



and



__43. Playback the process. **Areas of potential errors:**
Uncheck/Recheck the 'Has Default' options of the variable in the process screen and coaches.
Make sure you have mapped each coach view to the mortgageData variable.
A snapshot of the current state is available here:

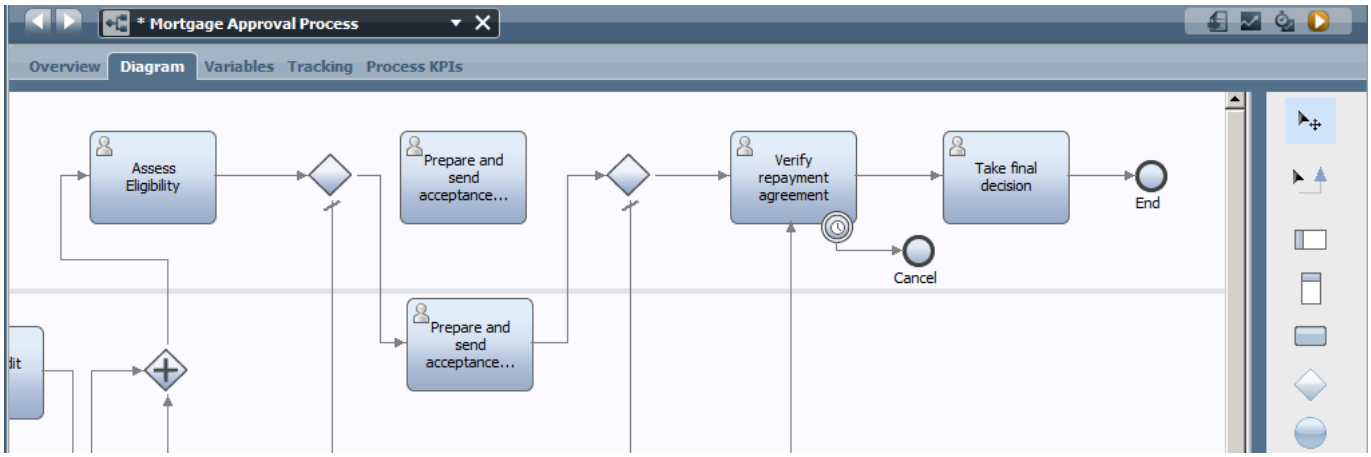
Snapshot: Decision Service and coach views.

Mortgage_Application_Process - Part_2_Decision_service_and_coach_views.twx

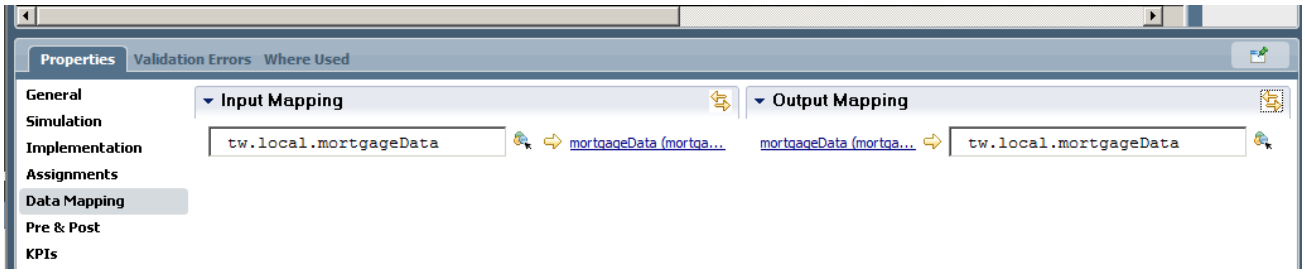
2.6 Creating the Prepare and send acceptance pack coach

In order to speed up development of the Prepare and send acceptance pack coach we'll simply duplicate the Assess Eligiblity coach.

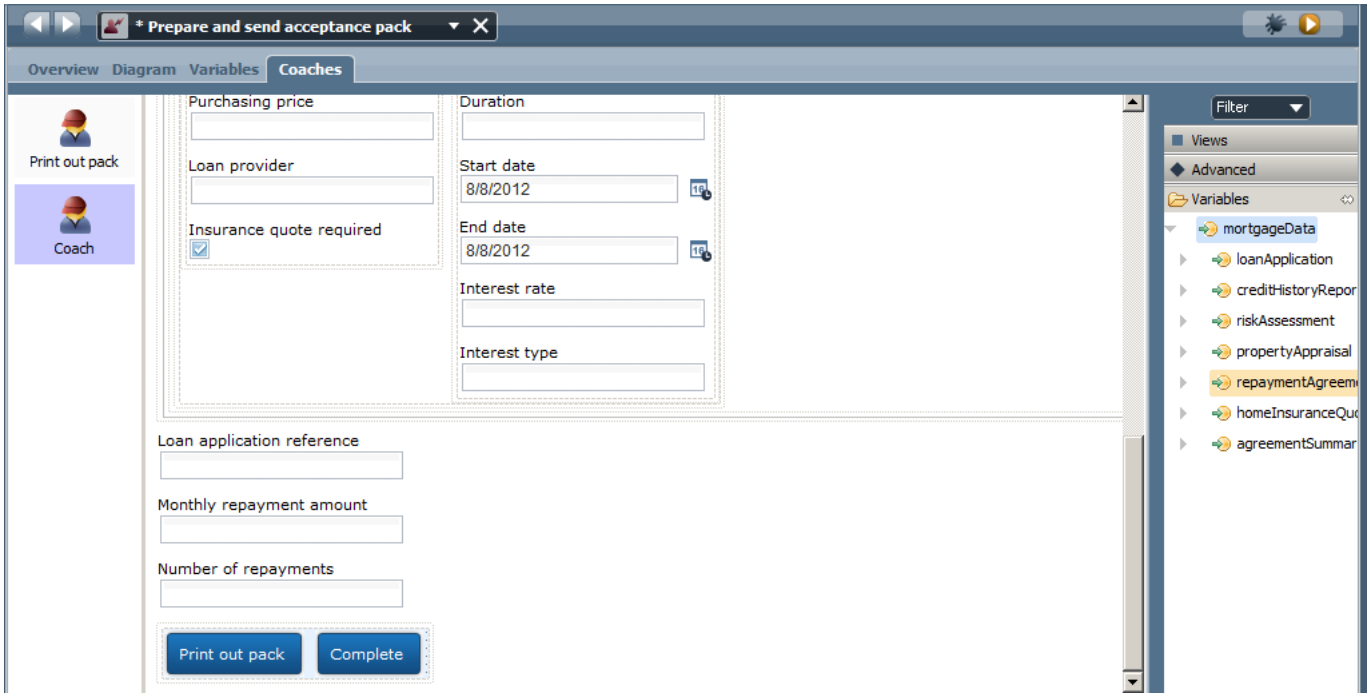
__1. From the **Process** diagram, right click the Assess Eligiblity coach and select 'Duplicate..'. Drag the new coach onto the canvas. Connect the path from the original Prepare and send acceptance pack task to the new one and delete the old one.



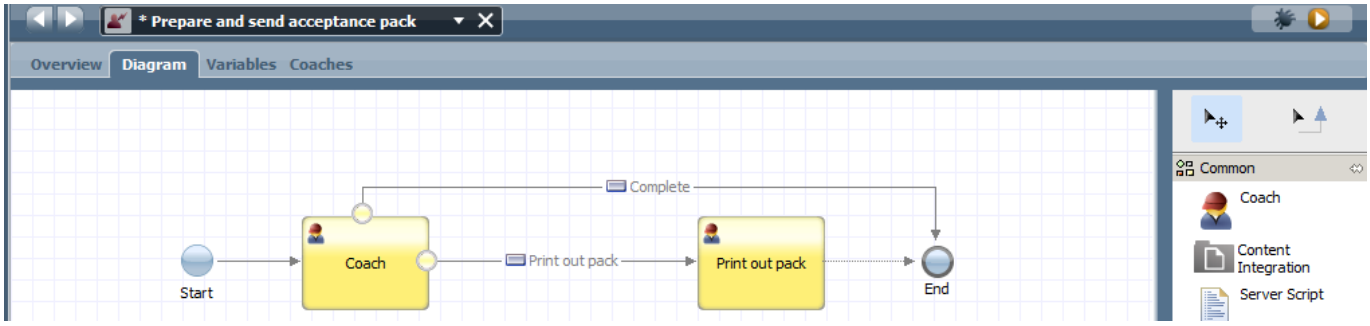
__2. Map the IN/OUT parameters of the coach to mortgageData:



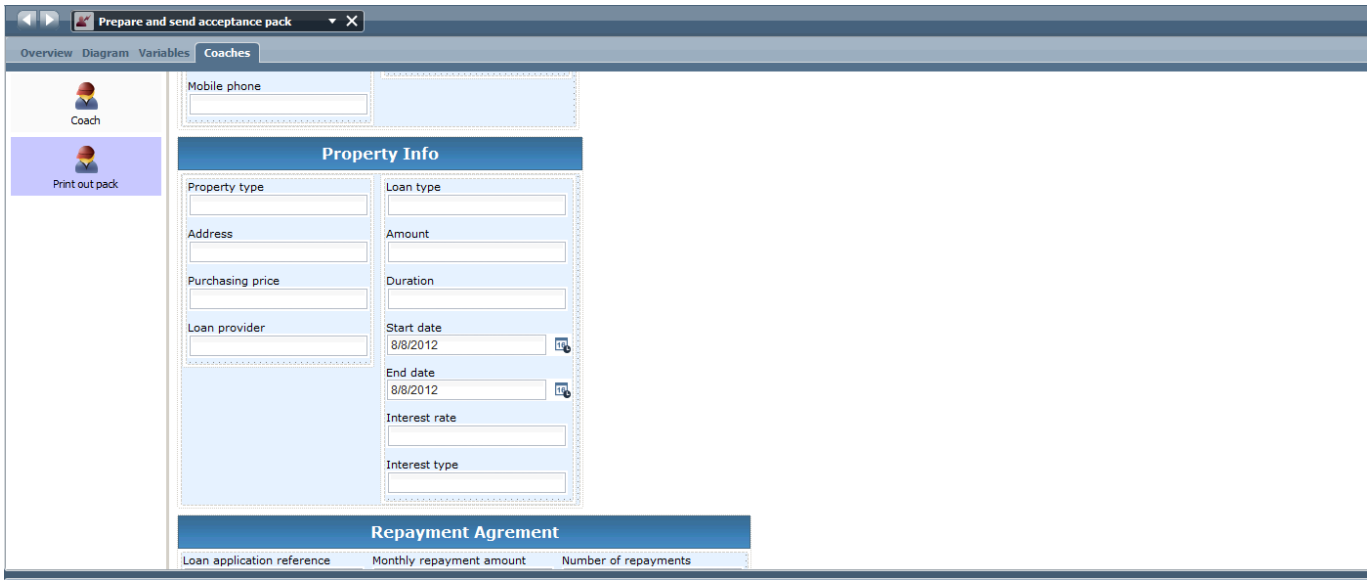
__3. in the original coach delete all extra fields that are under the coach views as we don't need them. Then drag the repaymentAgreement variable to the coach, this will generate 3 text fields containing the repayment information. Rename the 'Reject' application button to 'Complete'.



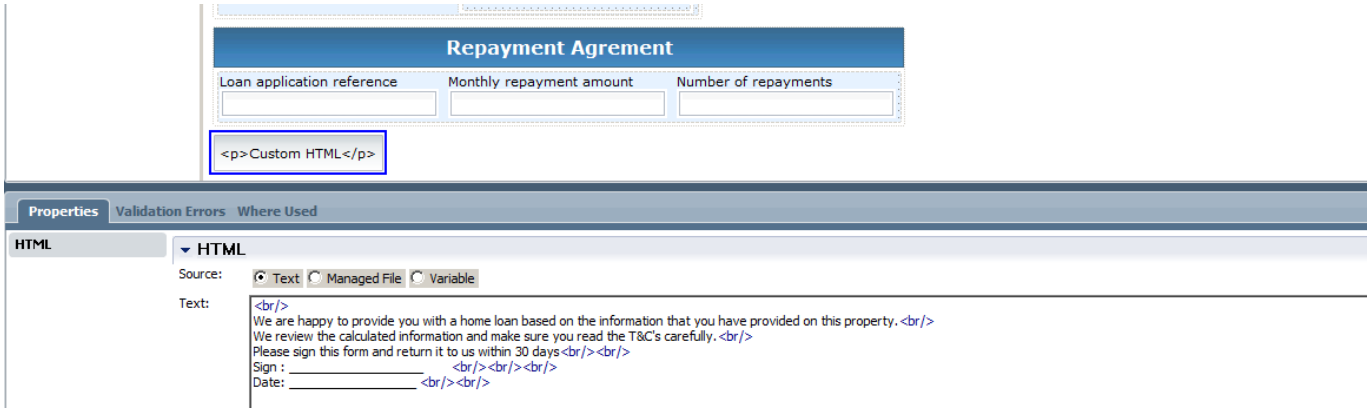
__4. Double click on the task to open the coach diagram. We now want to create a separate coach that contains all the data that we want to print. Drag a coach symbol onto the canvas and connect it to the existing coach. Drag another arrow from the coach to the End for the complete action. In the new coach change the label to 'Print out pack'



__5. Double click the 'Print out pack' coach to open it. We can now add any information that we want to print out to this coach. Open the applicationView coachView and copy the personal and property info table. Then drag and drop the repayment agreement variable onto the coach.

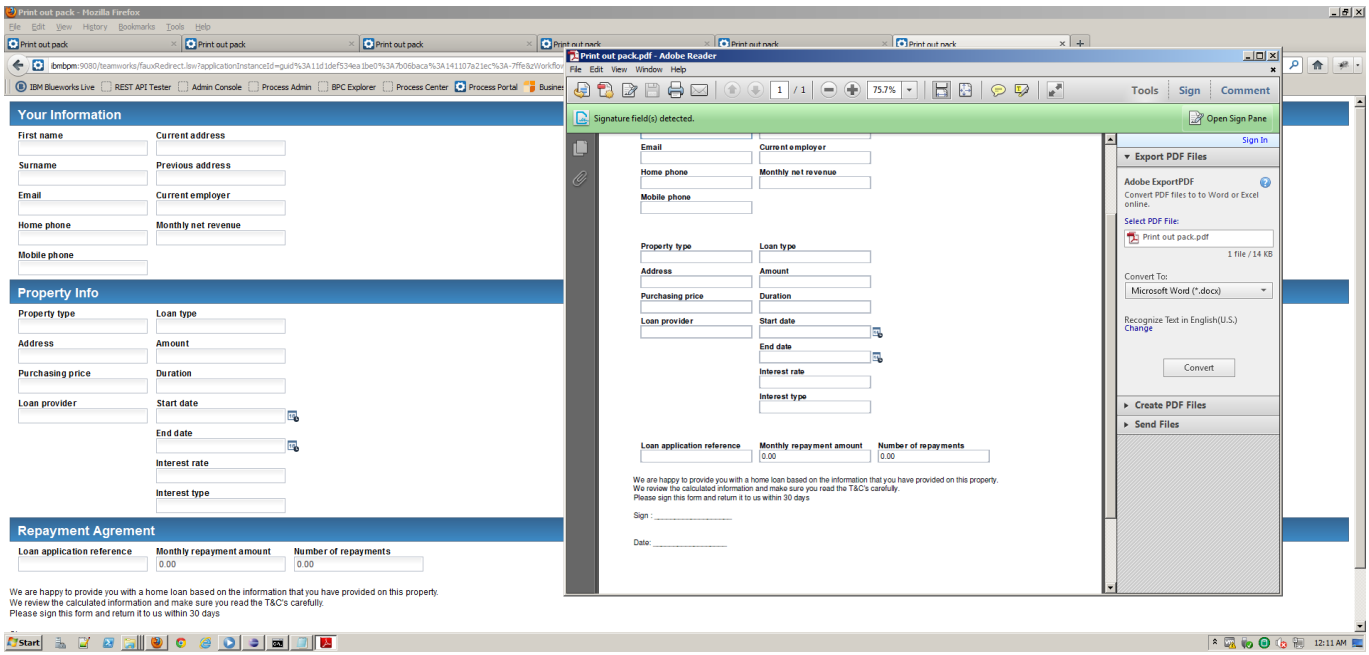


__6. To create custom text onto the coach, drag a custom HTML object onto the coach and enter free text on the screen.



__7. Download and install the free CutePDF writer from http://www.cutepdf.com/Products/CutePDF/writer.asp:

__8. Preview the coach, you can now use the normal print functions of your browser to either print the coach to a printer or to a PDF using CutePDF.



__9. For more advanced PDF export functionality and to programmatically build up the PDF export, import this toolkit.

Snapshot: PDF Generation sample.

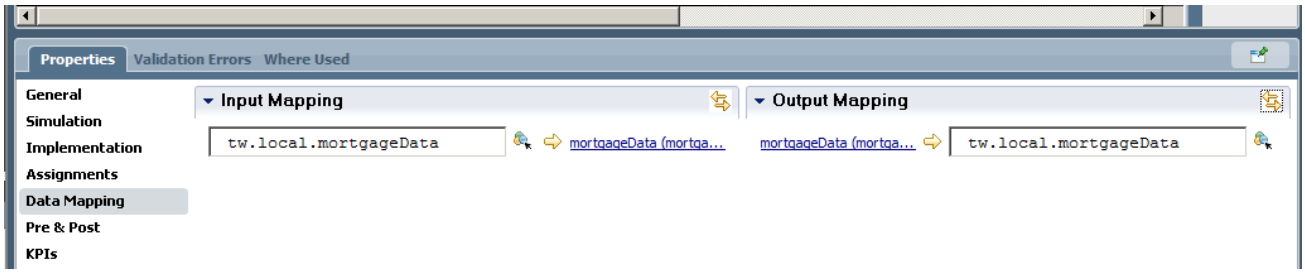
PDF_Generation_Sample - 0.3.twx

2.6.1 Creating the Prepare and send home insurance quote coach

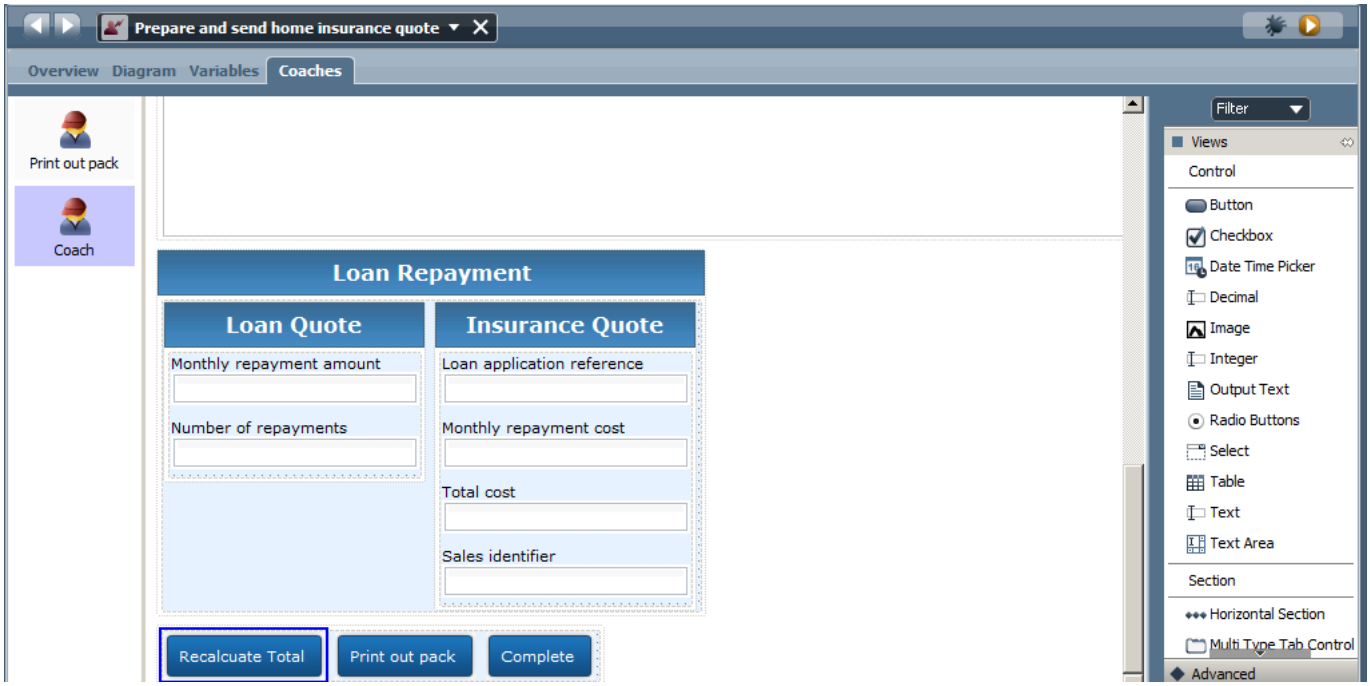
Duplicate the Prepare and send acceptance pack coach.

__10. From the **Process** diagram, right click the Prepare and send acceptance pack coach and select 'Duplicate..'. Drag the new coach onto the canvas. Connect the path from the original coach to the new one and delete the old one.

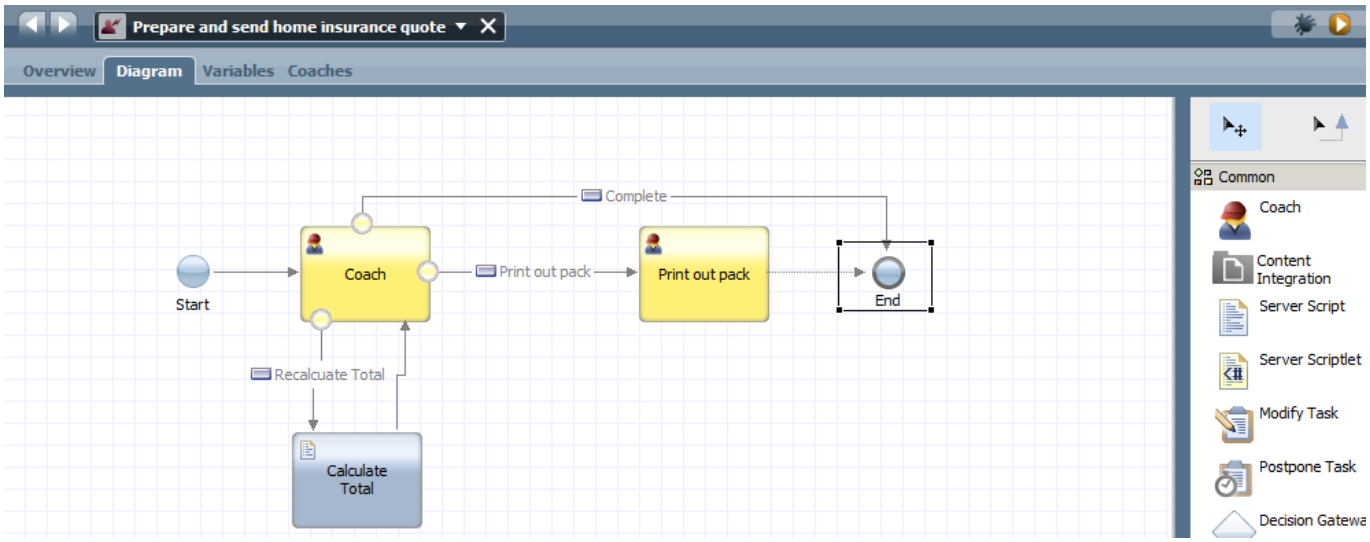
__11. Map the IN/OUT parameters of the coach to mortgageData:



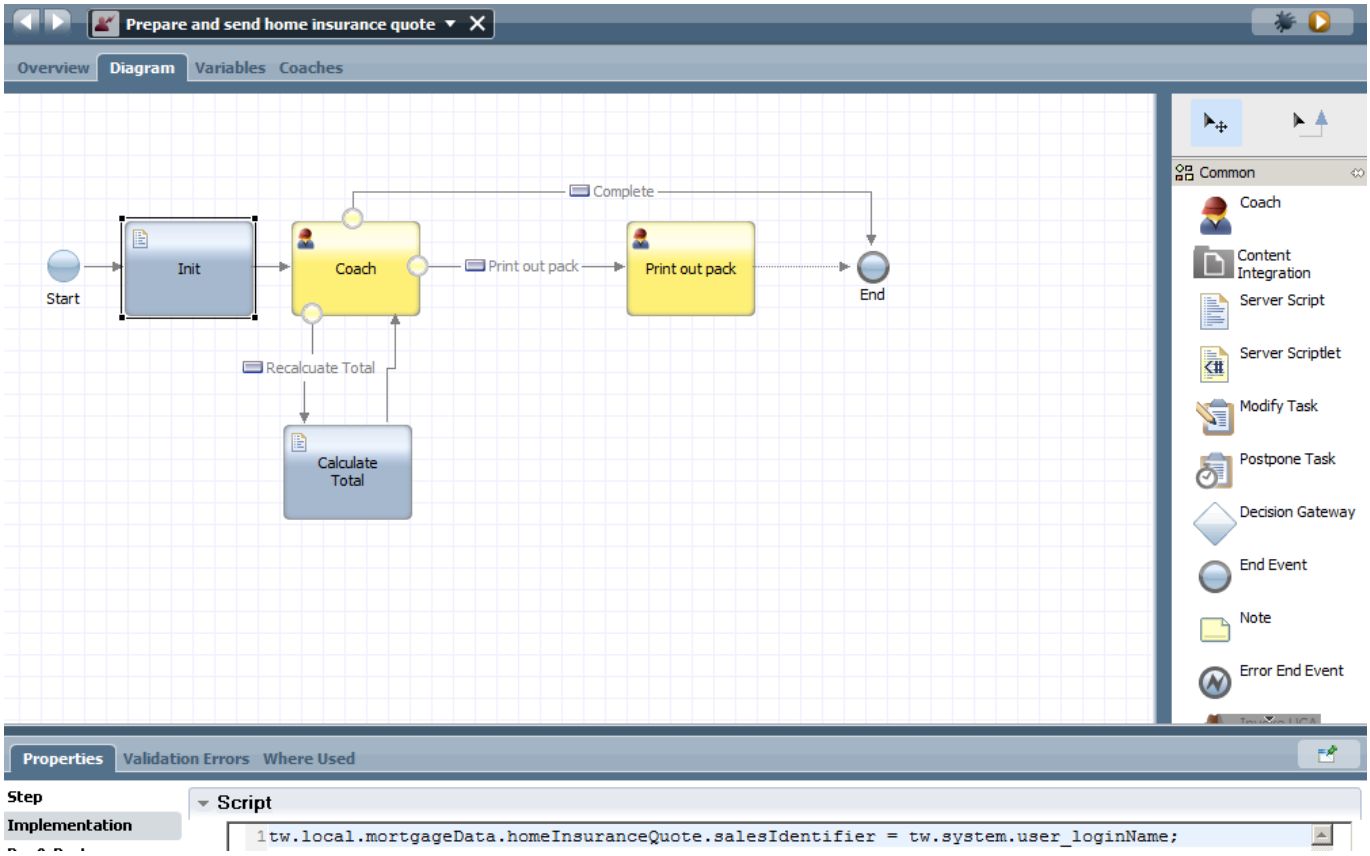
__12. Drag and drop the variable elements of repaymentAgreement and homeInsuranceQuote and re-arrange them using horizontal and vertical sections into the following format. Add a button and name it 'Recalculate Total'



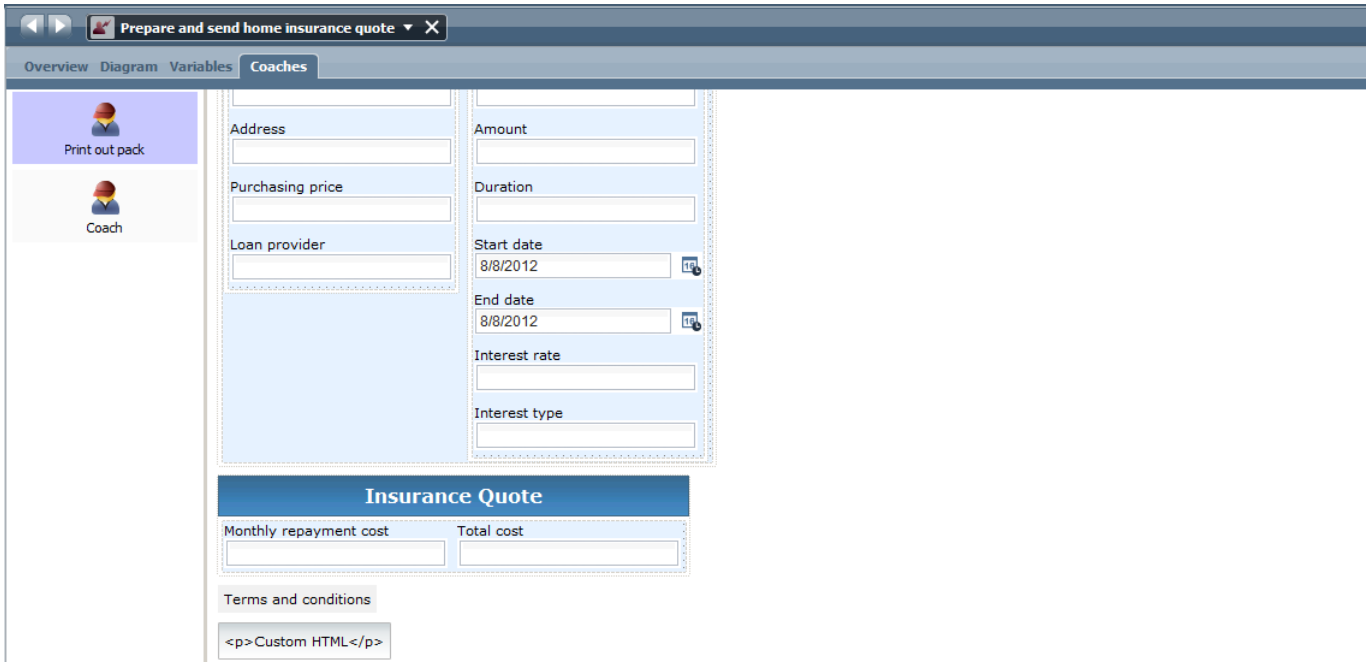
__13. Switch to the diagram view and add a server script to the canvas with the following calculation:
tw.local.mortgageData.homeInsuranceQuote.totalCost =
tw.local.mortgageData.repaymentAgreement.monthlyRepaymentAmount +
tw.local.mortgageData.homeInsuranceQuote.monthlyRepaymentCost;



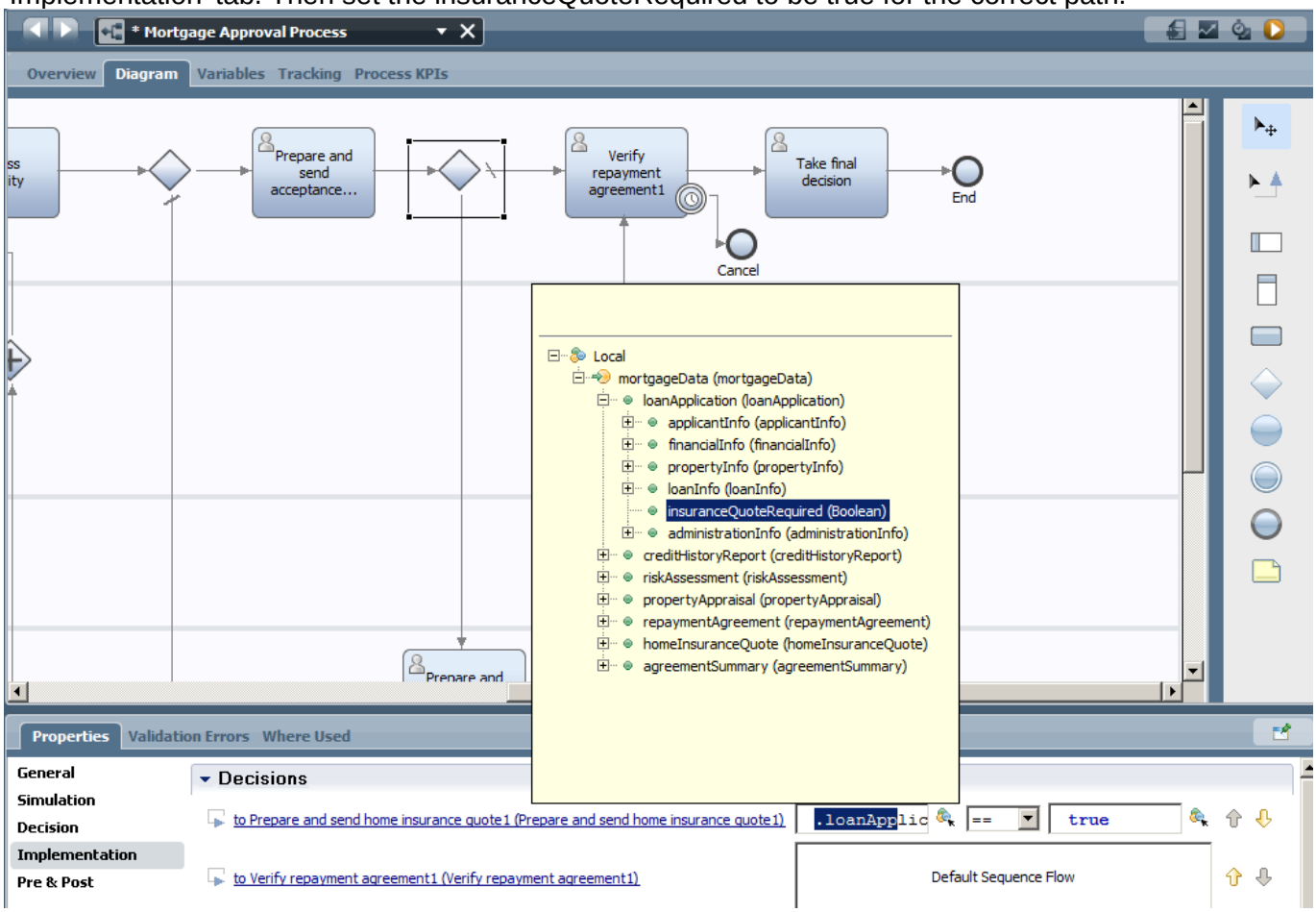
__14. Add an init field to the diagram to set the HomeInsurance salesIdentifier to the current logged in user:



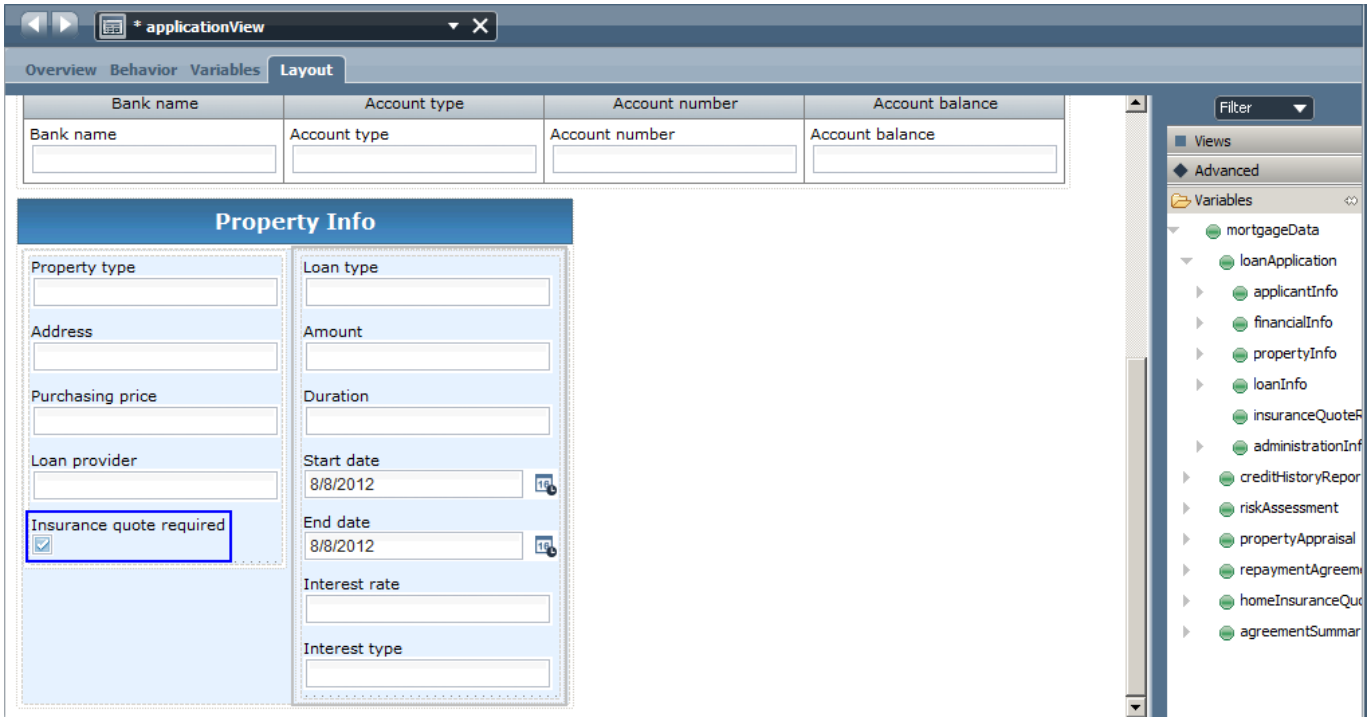
__15. Adjust the print out pack coach to have the fields as shown below and adjust the Custom HTML text to reflect insurance content, change the Terms and Conditions field to be an output field:



__16. Implement the decision gate logic to determine if the flow should go to insurance quote step or directly to the verify repayment agreement. Click on the decision diamond and select the 'Implementation' tab. Then set the insuranceQuoteRequired to be true for the correct path.



__17. We haven't added the insuranceQuoteRequired boolean to the applicationView coachView yet. Open the applicationView coachView and drag the insuranceQuoteRequired variable onto the canvas. Save and close the view. All coaches which have implemented this coachView will now be automatically updated.

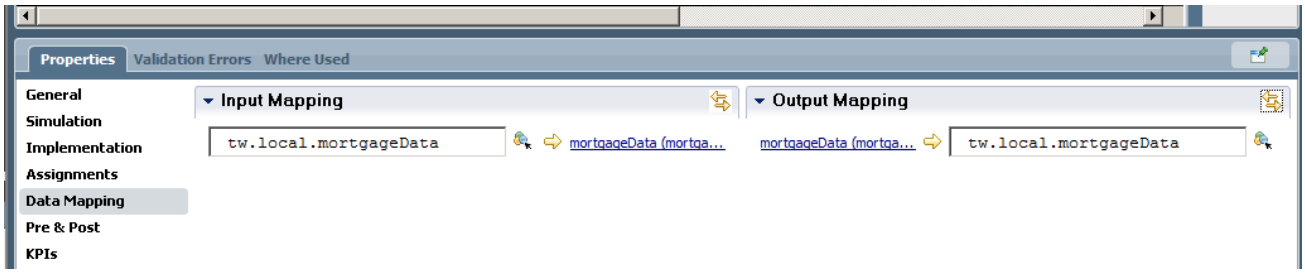


2.6.2 Verify Repayment agreement coach

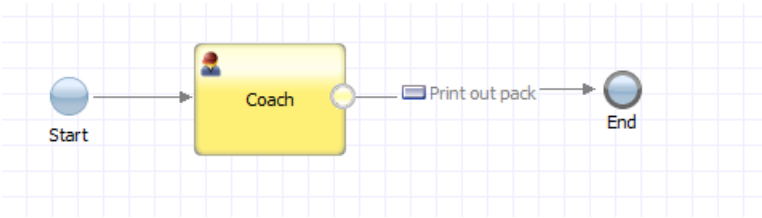
Duplicate the Prepare and send acceptance pack coach.

__18. From the **Process** diagram, right click the Prepare and send acceptance pack coach and select 'Duplicate..'. Drag the new coach onto the canvas. Connect the path from the original coach to the new one and delete the old one.

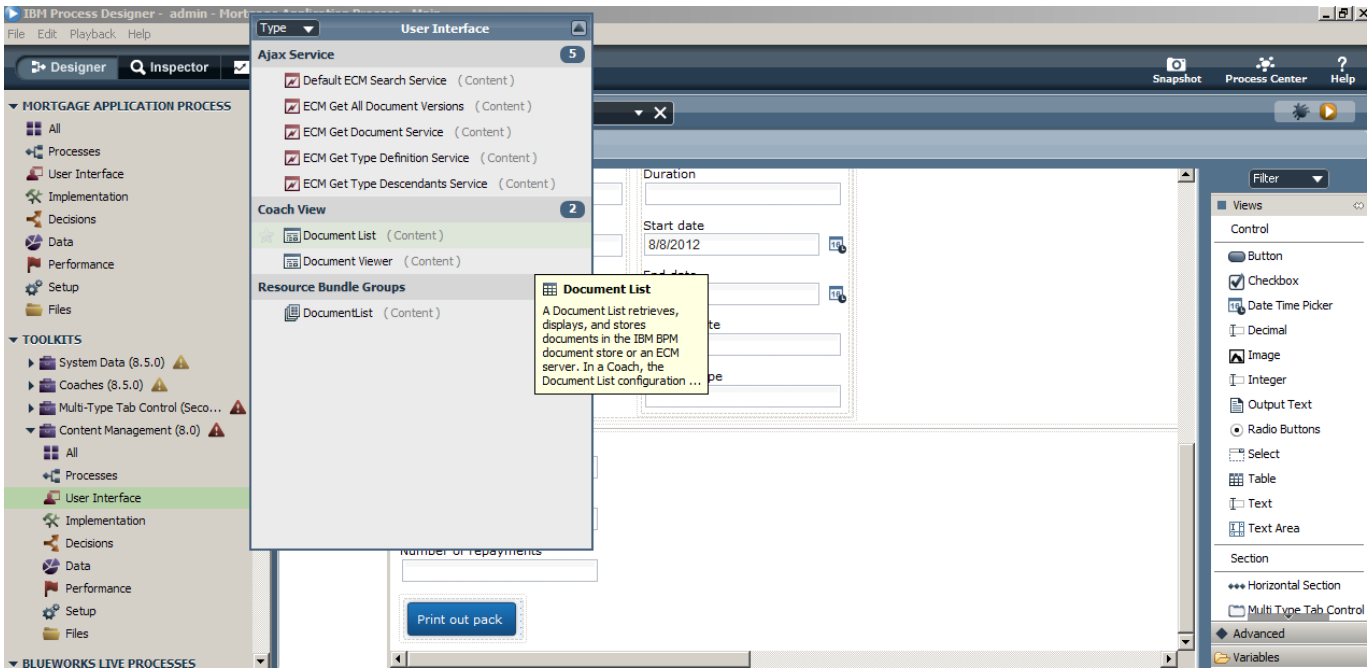
__19. Map the IN/OUT parameters of the coach to mortgageData:



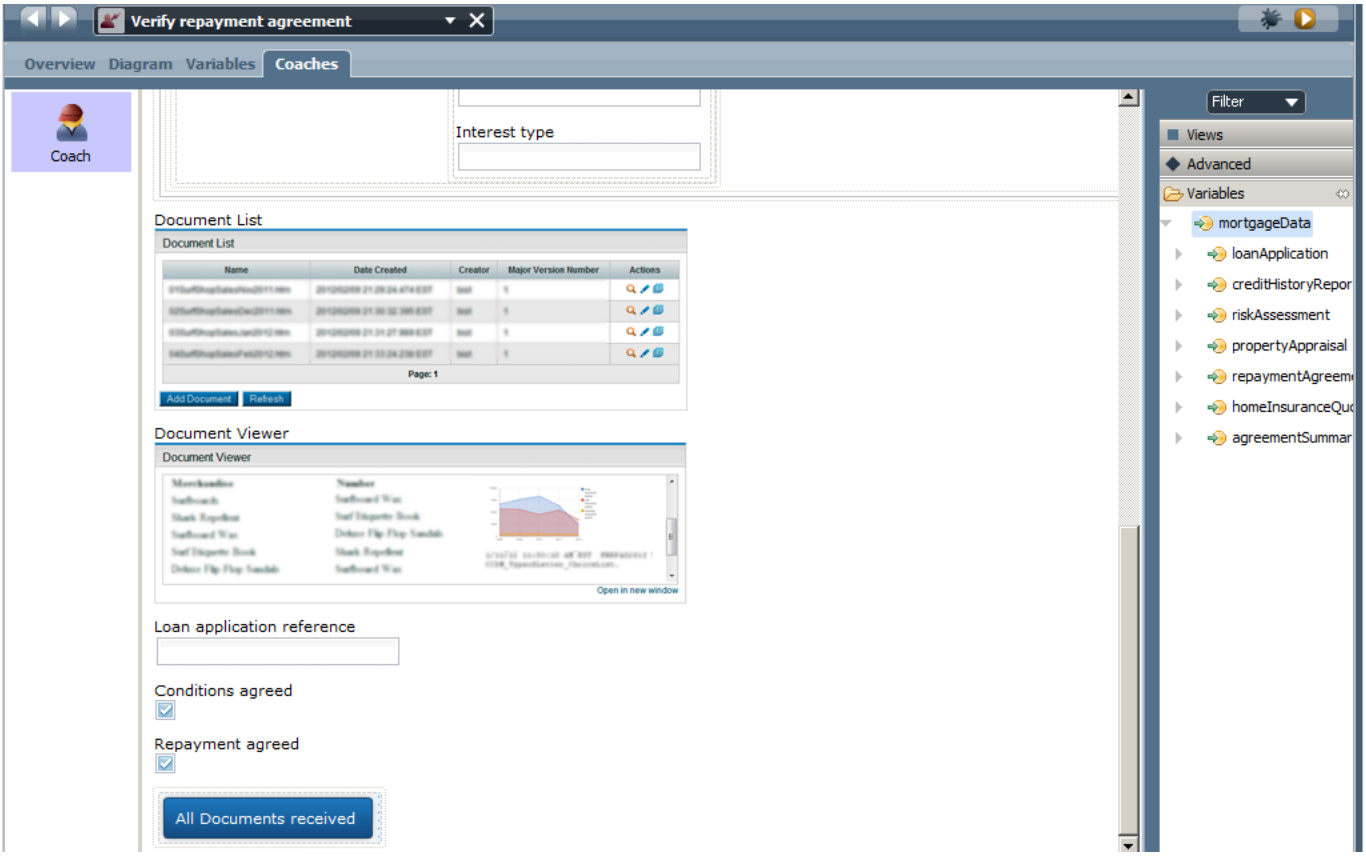
__20. Switch to the diagram view of the new coach and delete the 'Print out pack' coach.



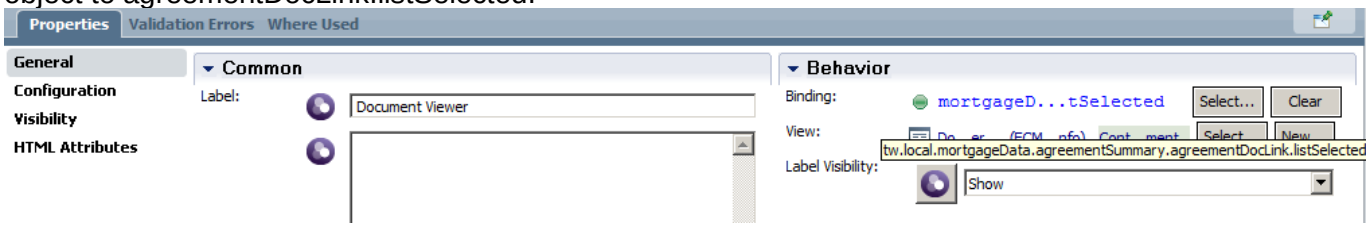
__21. Switch to the coach layout view and add a Document List and Viewer from the Content Management toolkit on the canvas under the existing coach views, if the Content Management toolkit is not visible, click on the '+' symbol next to TOOLKITS to add it, right click on the toolkit to upgrade its version to 8.5.0 :



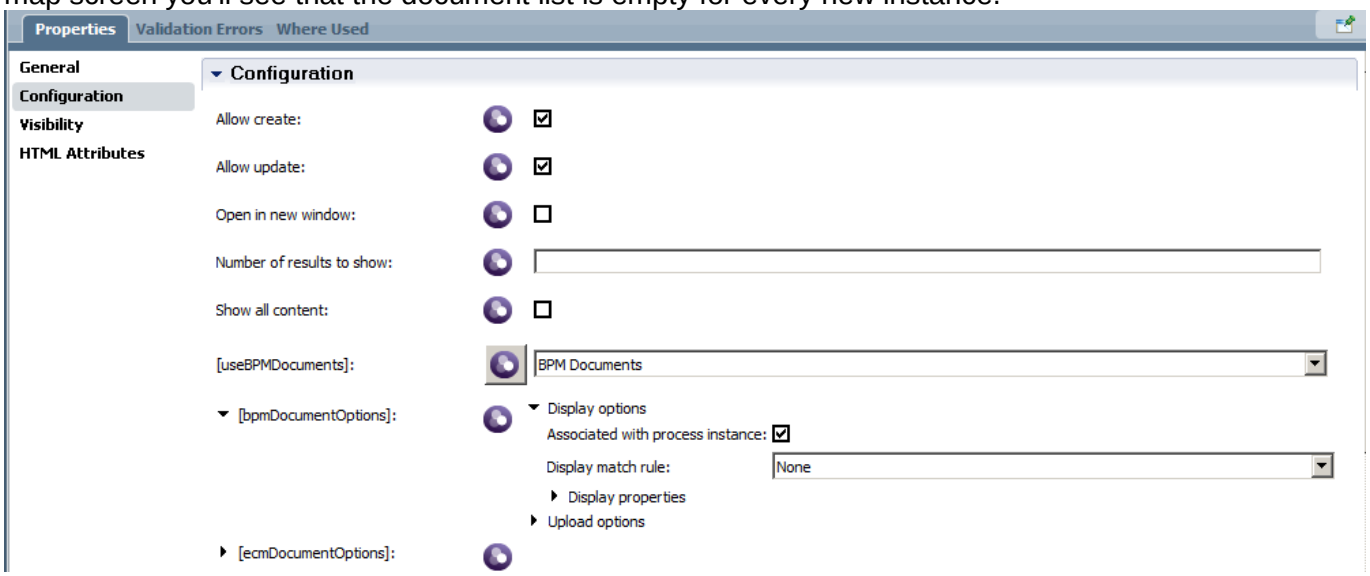
__22. Delete the existing text fields under the coach views and replace them by the agreementSummary variables. As we use the build in attachment functionality we don't need the agreement doc link text field, delete it:



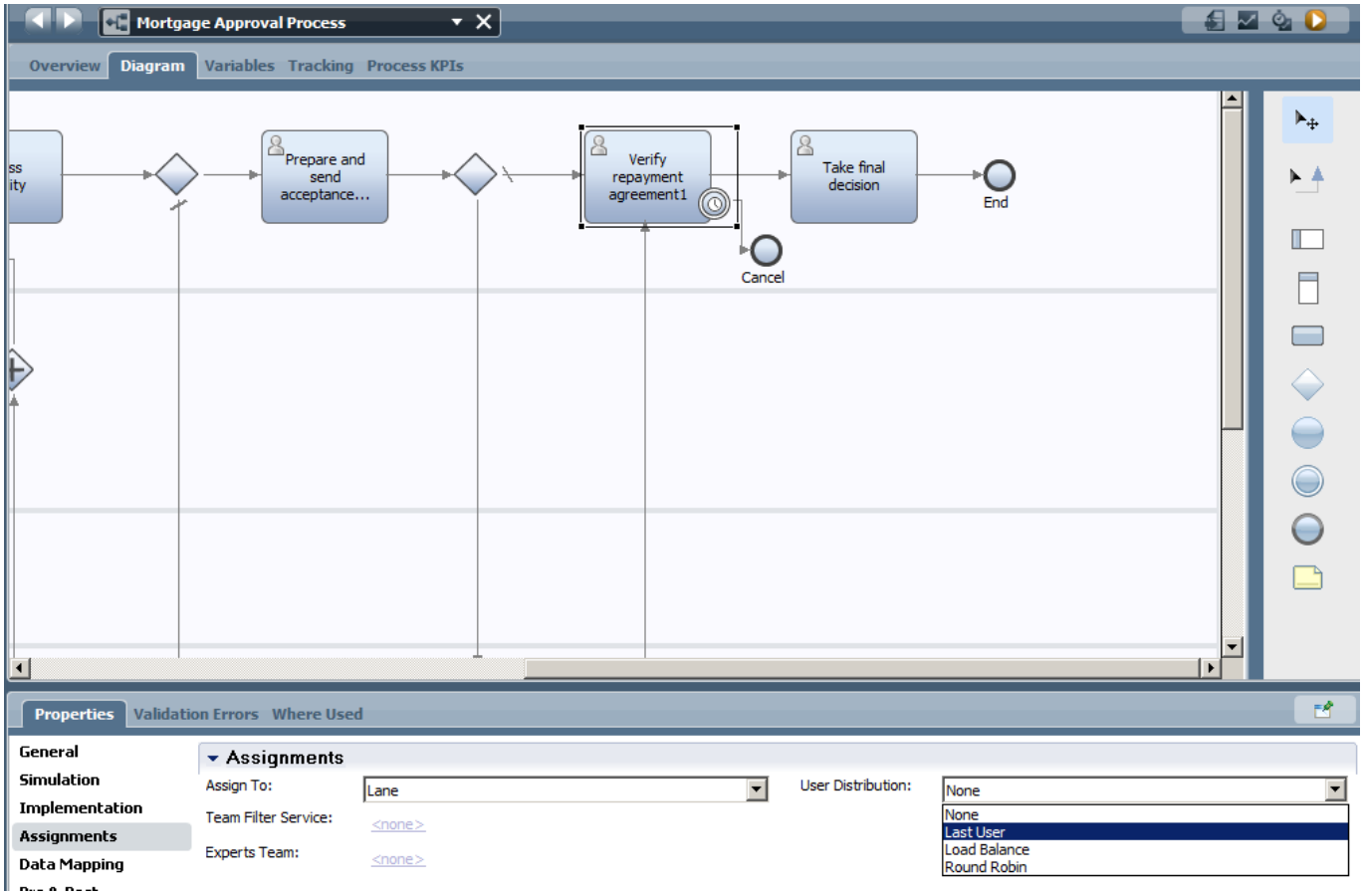
__23. We now need to bind the Document List object to agreementDocLink and the Document Viewer object to agreementDocLink.listSelected.



__24. Switch to the properties view of the Document List object and set the following options, make sure to set useBPMDocuments to 'BPM Documents'. As you test the coach you might notice that documents previously uploaded are also visible in the test screen. This is because the single coach playback test uses a generic instance ID. If you use the playback button on the process map screen you'll see that the document list is empty for every new instance.



__25. Assign work to previous loan worker. To provide consistency across the case we'll assign the task to the same loan officer as in previous steps. Switch to the process map view and click on the verify repayment agreement task. Click on the Assignments tab and set 'user distribution' to 'Last User'.

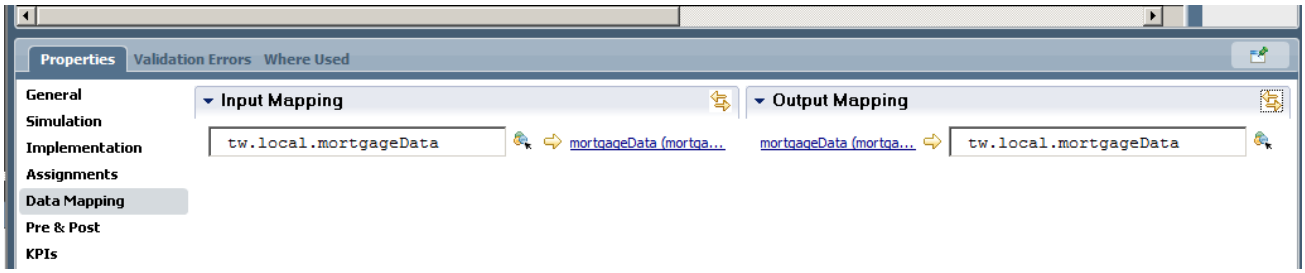


2.6.3 Take Final Decision coach

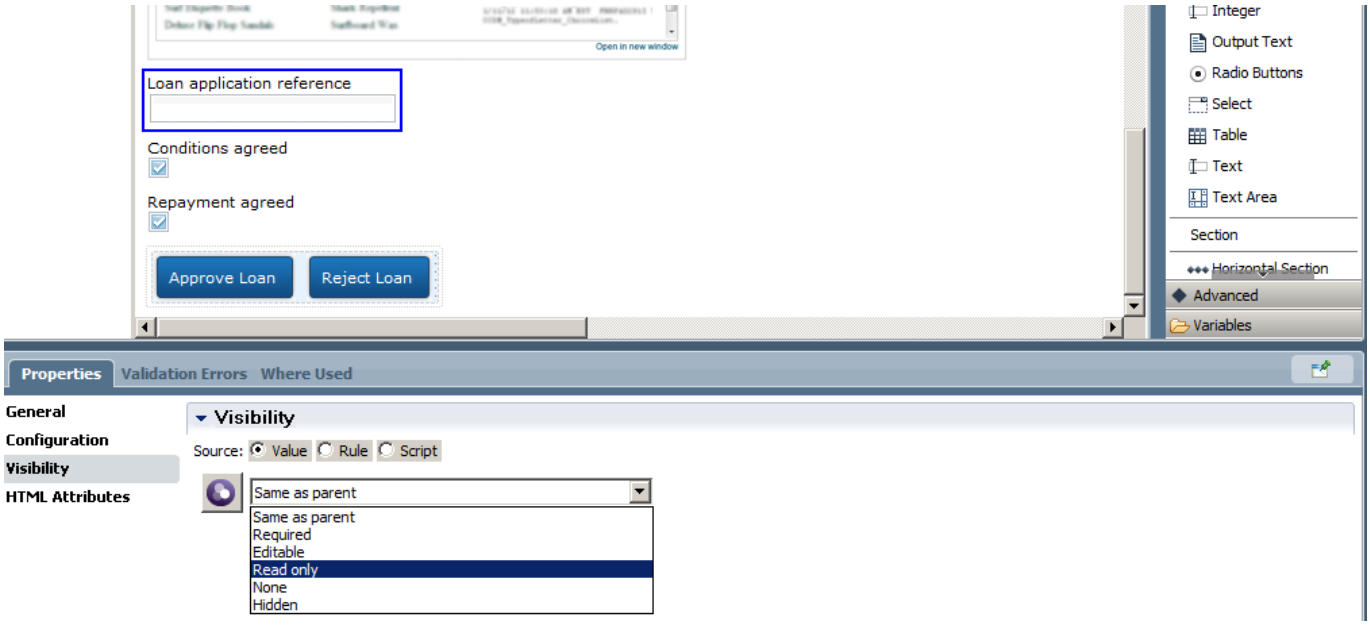
Duplicate the Verify repayment agreement coach.

__26. From the **Process** diagram, right click the Verify repayment agreement coach and select 'Duplicate..'. Drag the new coach onto the canvas. Connect the path from the original coach to the new one and delete the old one.

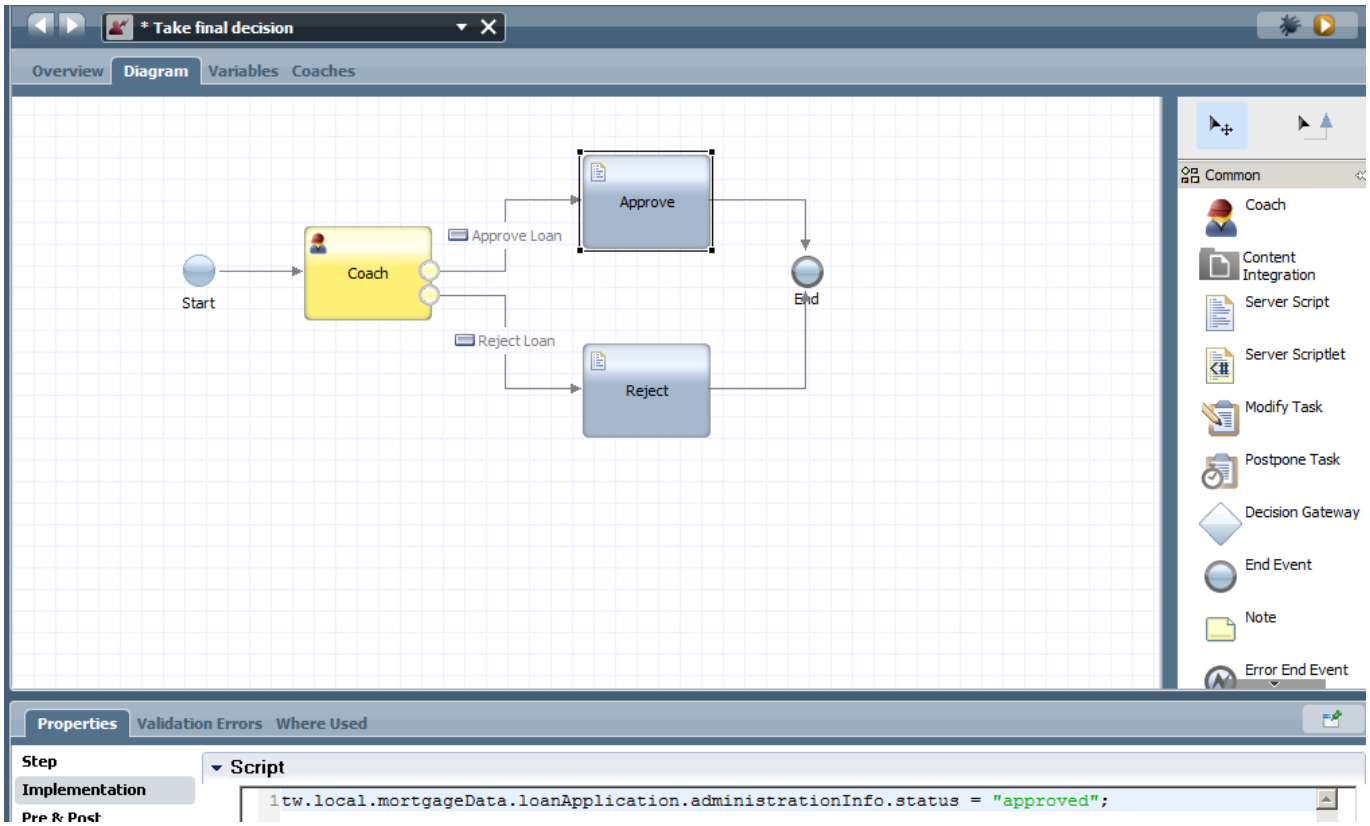
__27. Map the IN/OUT parameters of the coach to mortgageData:



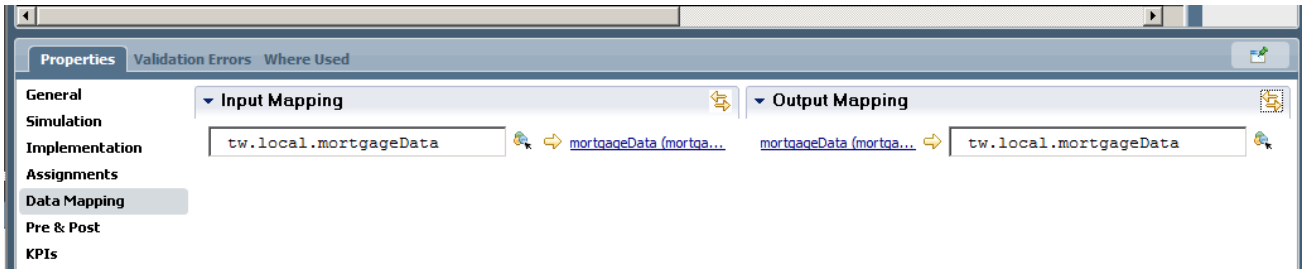
__28. Set all fields to be read-only if required, add two buttons with labels approve and reject loan:



__29. In the Diagram view of the coach, create two server script activities, link them to the coach and add script to set the status to approved or rejected.



__30. Map the IN/OUT parameters of the coach to mortgageData:

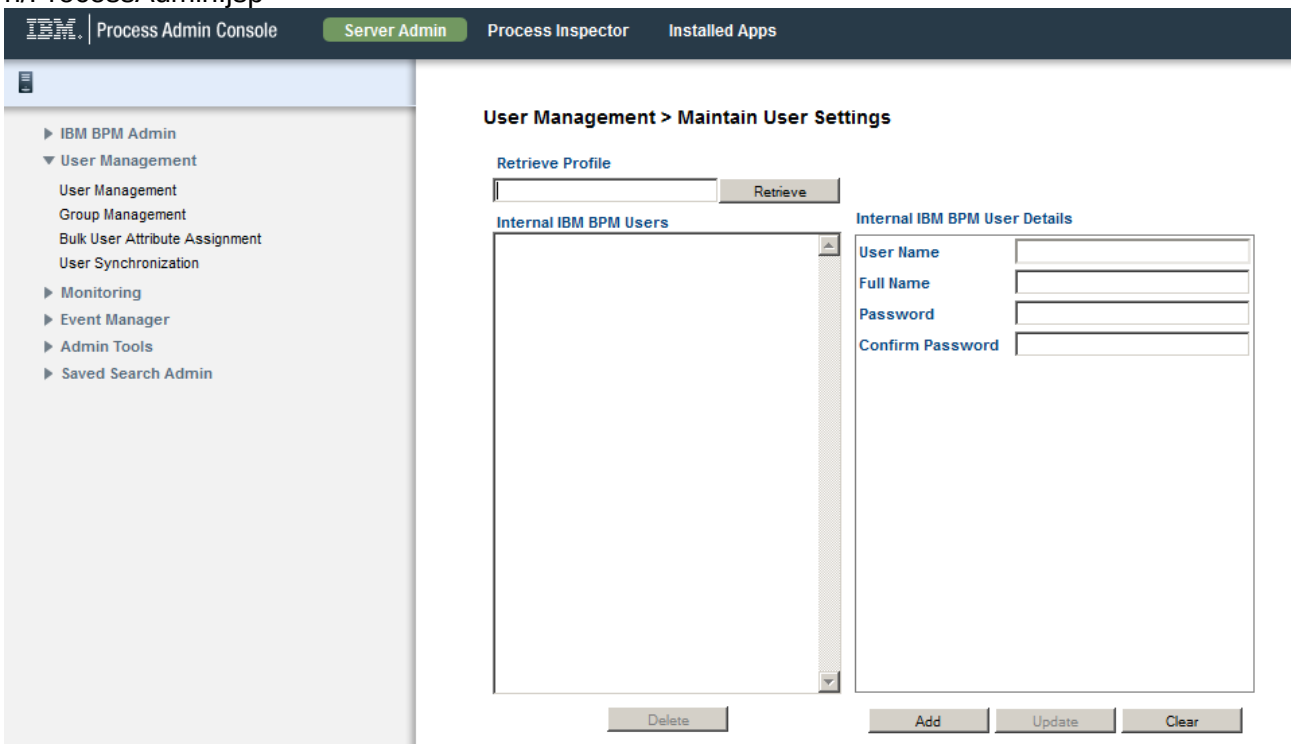


2.7 Assigning tasks to teams

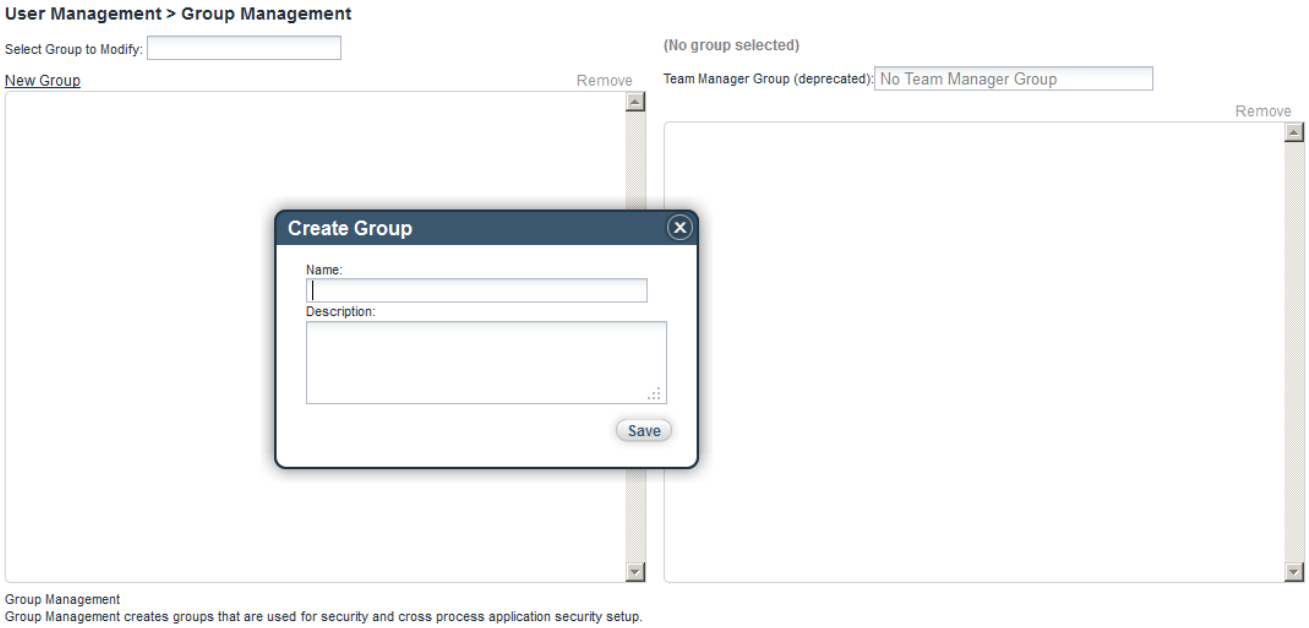
At this stage we have finished the process definition, next step is to create users, assign them to teams and assign teams to process swimlanes.

2.7.1 Create users

Open the Process Admin screen in the browser by going to this link:
https://yourserver:9443/ProcessAdmin/ProcessAdmin/com.lombardisoftware.processadmin.ProcessAdmin/ProcessAdmin.jsp

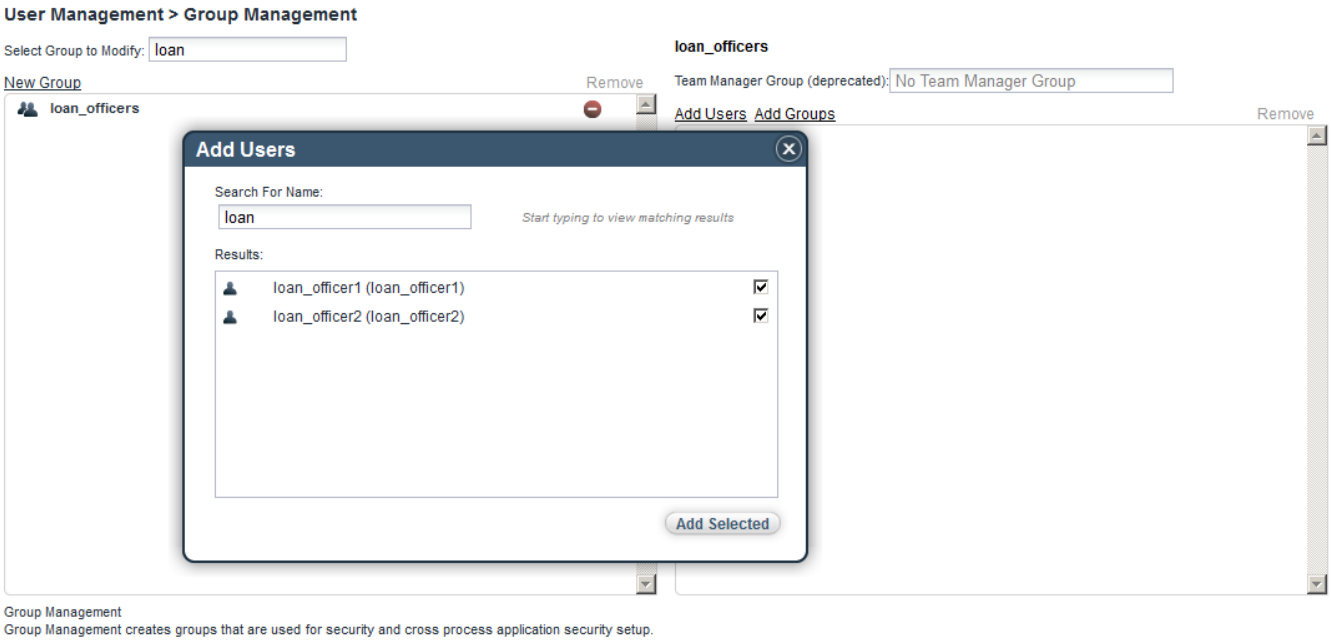


Enter a username, full name and a password (use tw_admin) for each new user, then click the 'add' button to confirm the creation.



On the group creation page, click 'New Group' and enter the name of a new group.

After creating a group, enter the group name in the 'Select Group to Modify' field, then click on the found result and select 'Add Users' to populate the group.



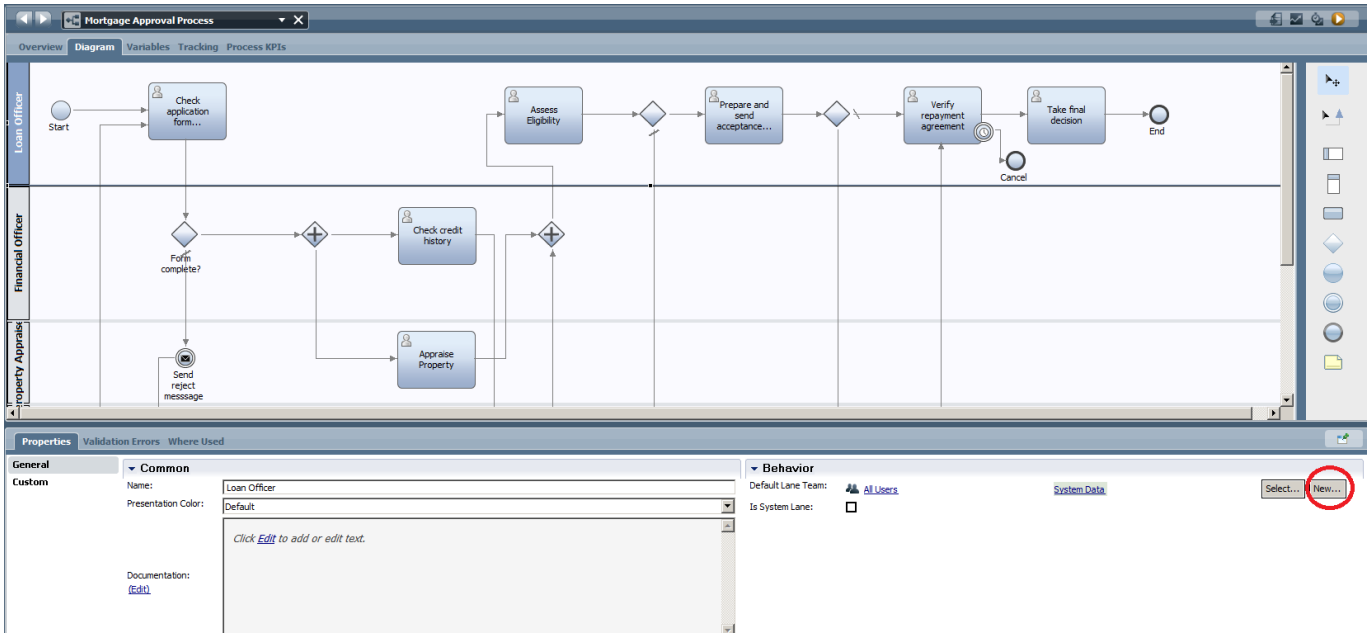
In the popup screen, type the name of the users that you want to add, check the box and add the selected members. (there is no Save button on this page, each action will trigger an autosave)

Repeat the process for all other users in the table below.: use tw_admin as the default password for all of them.

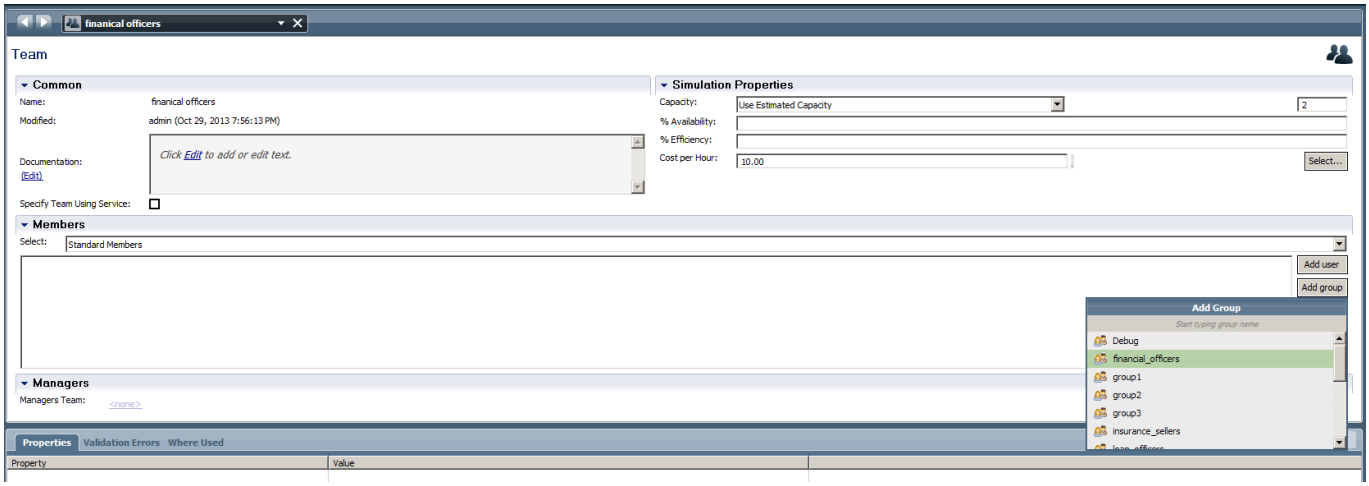
User	Full name	Group	Team
loan_officer1	loan_officer1	loan_officers	Loan officers
loan_officer2	loan_officer2		
financial_officer1	financial_officer1	financial_officers	Financial officers
property_appraiser1	property_appraiser1	property_appraisers	Property appraisers
insurance_sales1	insurance_sales1	insurance_sellers	Insurance sellers

2.7.2 Create teams

Switch back to the Process Designer and click on the Loan Officer swimlane. In the property screen, you can now create a new lane team, use the table above to create the team for each swimlane. After creating a team, associate the corresponding group with that team.



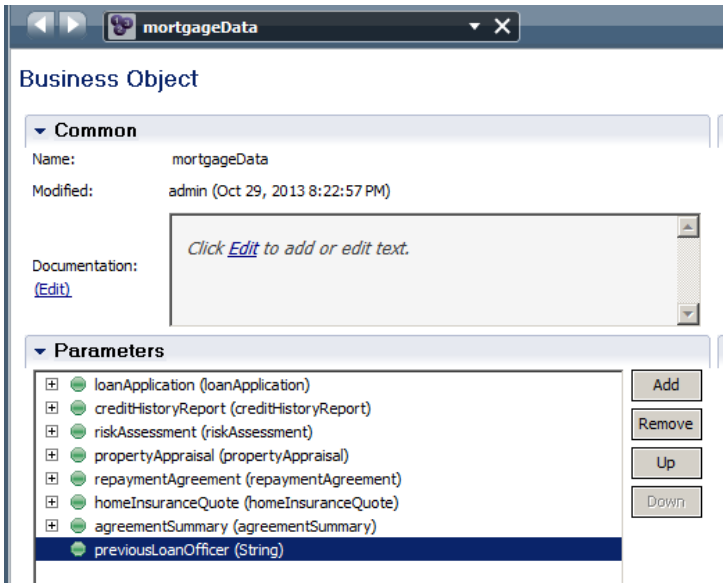
Click on each swimlane and create the corresponding teams.



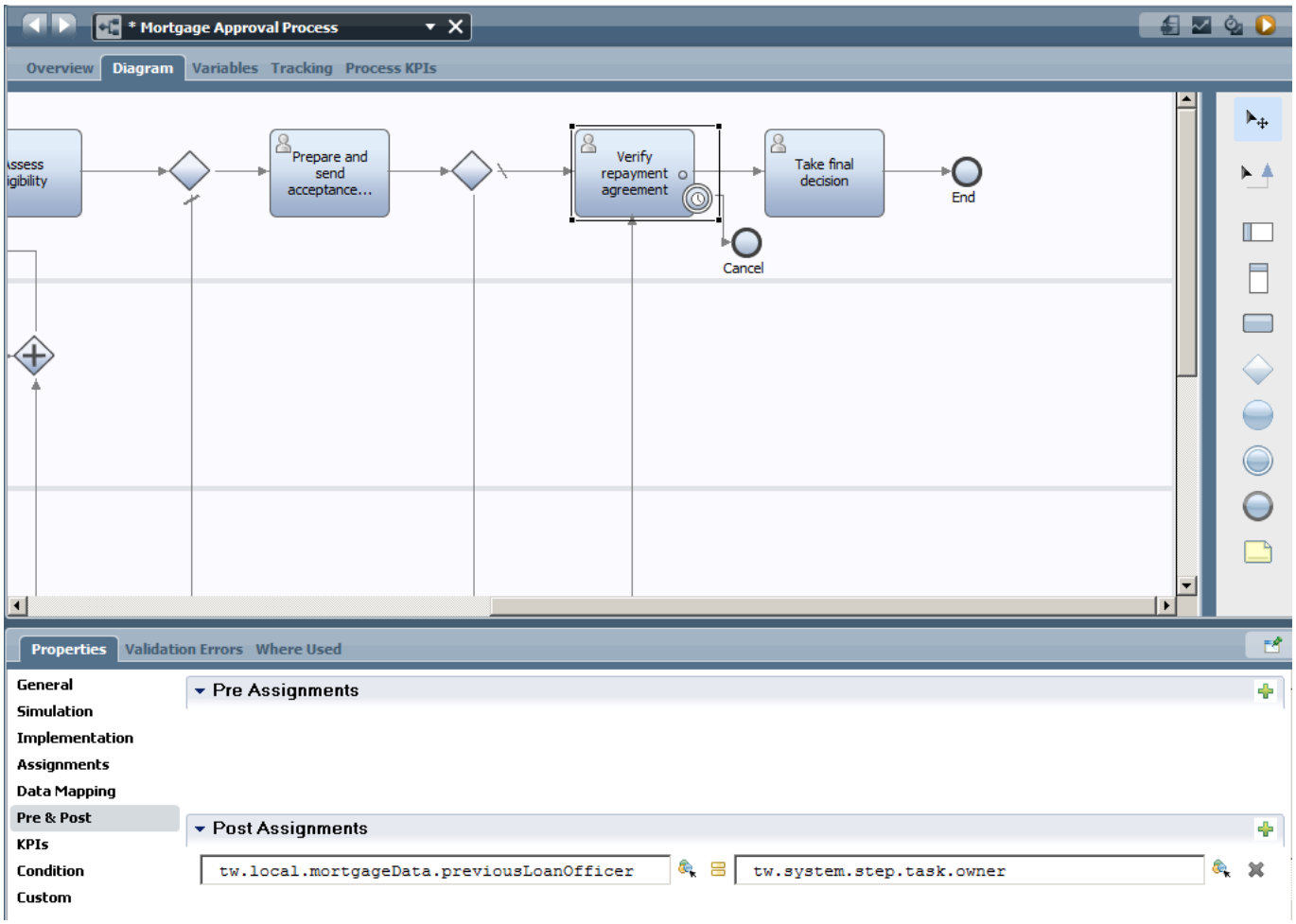
At this stage each lane has a team associated with it, the distribution of tasks among team members can be further refined per individual activity. In the scenario it has been specified that the final decision task cannot be performed by the same loan officer that verified the agreement.

In order to implement this logic we'll need to create a variable to store the previous user ID and apply a team filter for the 'Take Final Decision' task.

__1. On the process diagram, switch to the Variable tab: click on mortgageData and add a new variable called: previousLoanOfficer (String).



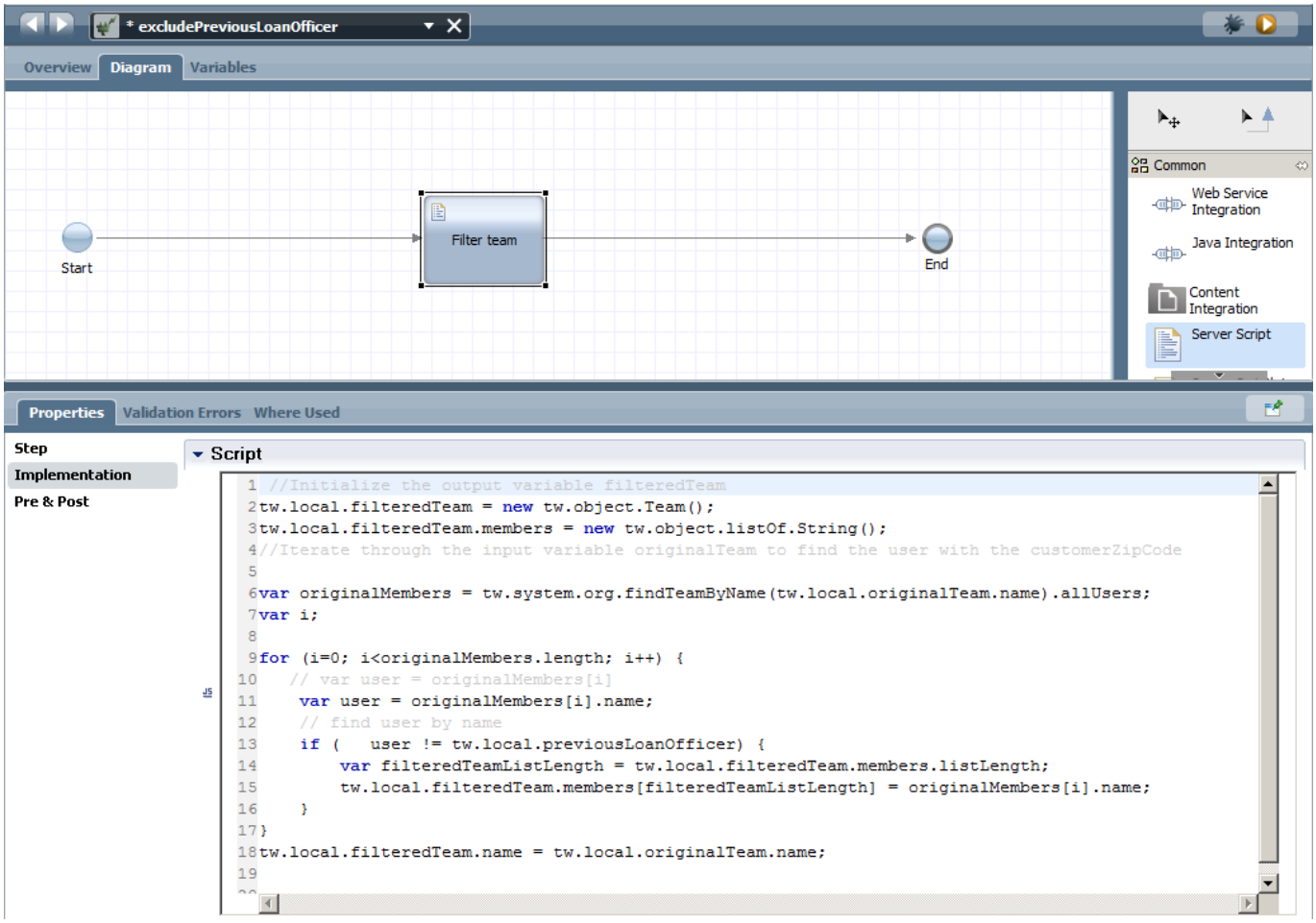
__2. On the **Process** diagram, click on the 'Verify repayment agreement' task and switch to the 'Pre & Post' tab in the properties view. Assign the tw.system.step.task.owner value to the tw.local.mortgageData.previousLoanOfficer variable.



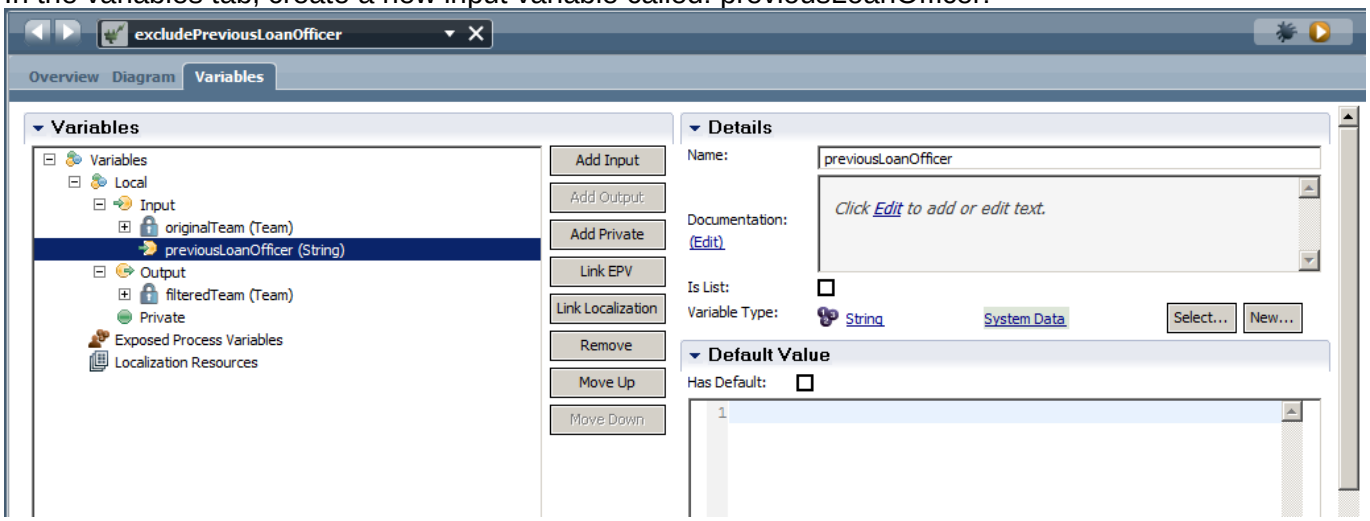
__3. We now need to create a team filter: Click on the "Assignments" tab and create a new Team Filter service: name the service excludePreviousLoanOfficer.

__4. Create a server script on the canvas, connect it and insert the following code in the Implementation tab:

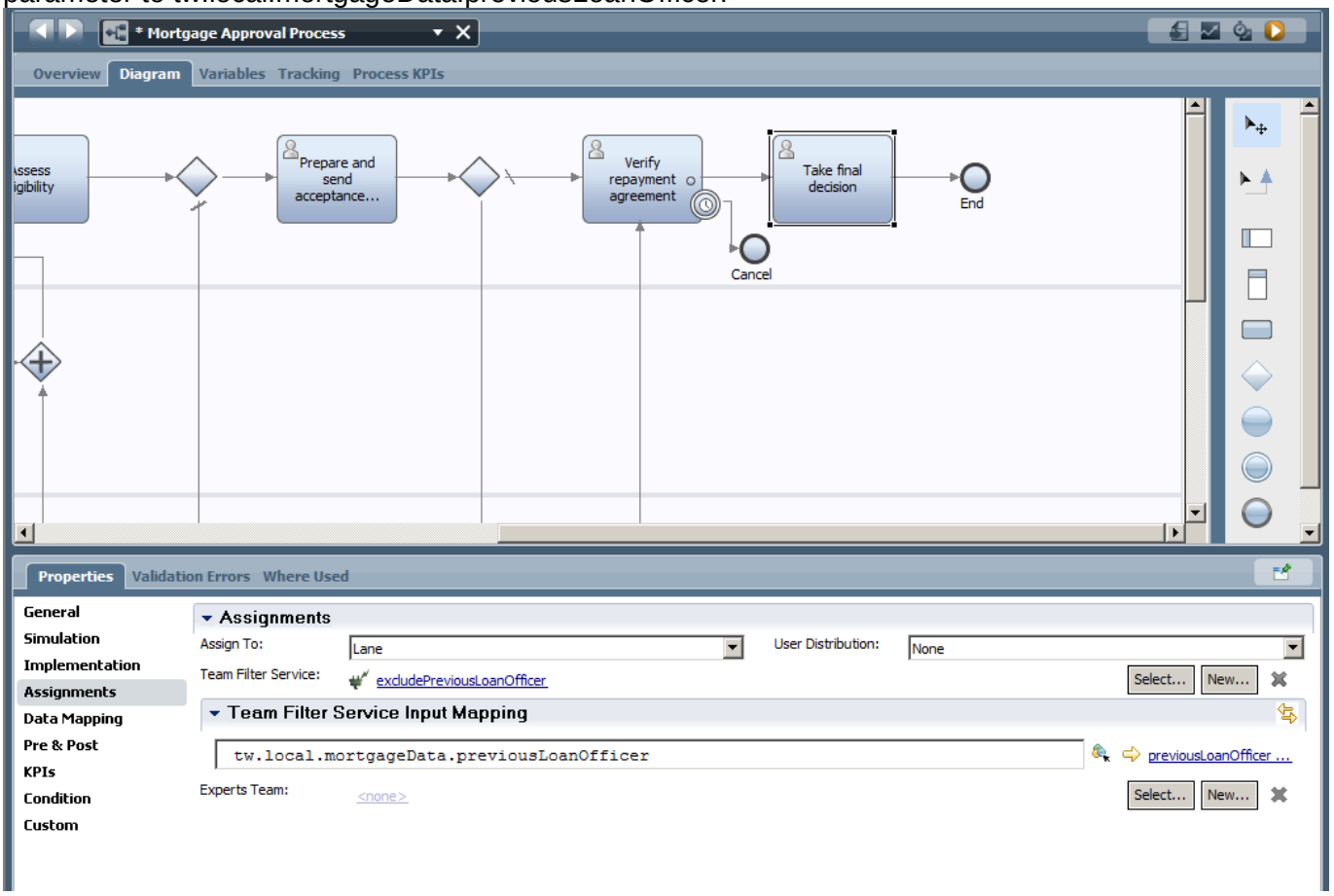
```
//Initialize the output variable filteredTeam
tw.local.filteredTeam = new tw.object.Team();
tw.local.filteredTeam.members = new tw.object.listOf.String();
//Iterate through the input variable originalTeam
var originalMembers =
tw.system.org.findTeamByName(tw.local.originalTeam.name).allUsers;
var i;
for (i=0; i<originalMembers.length; i++) {
    var user = originalMembers[i].name;
    // filter user by name
    if (   user != tw.local.previousLoanOfficer) {
        var filteredTeamListLength = tw.local.filteredTeam.members.listLength;
        tw.local.filteredTeam.members[filteredTeamListLength] =
originalMembers[i].name;
    }
}
tw.local.filteredTeam.name = tw.local.originalTeam.name;
```



__5. In the variables tab, create a new input variable called: previousLoanOfficer:



__6. Save and close the service. In the previous assignment screen link the Team Filter service input parameter to tw.local.mortgageData.previousLoanOfficer.



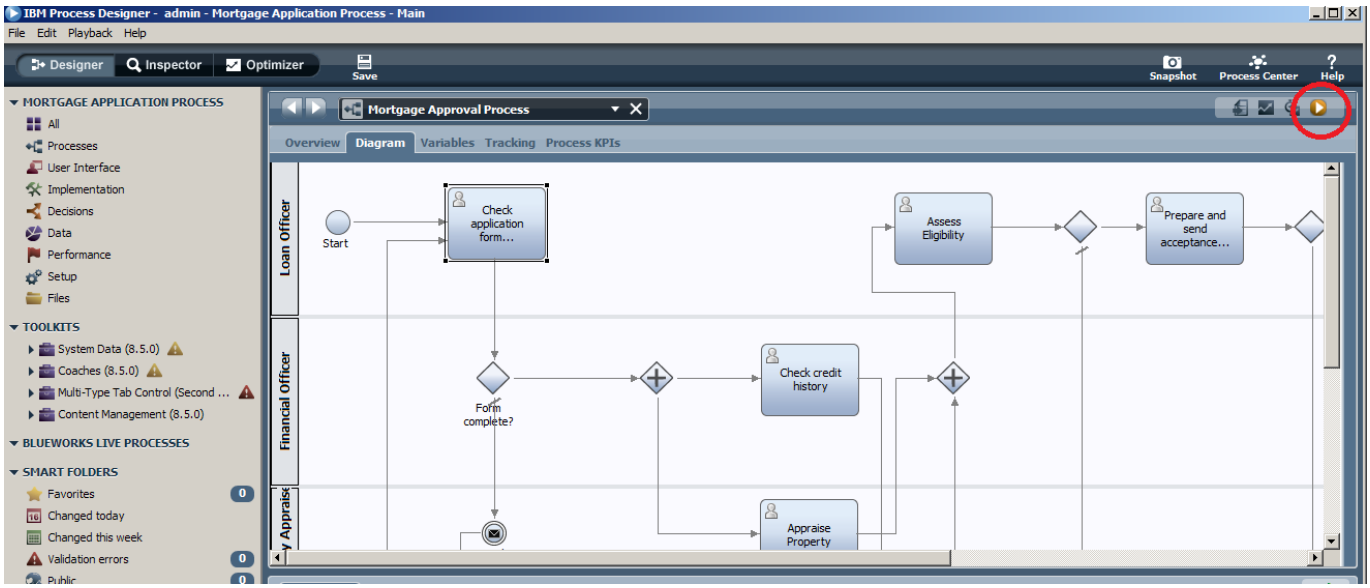
__7. Your process is now finished and ready for testing.

Snapshot: Final process diagram

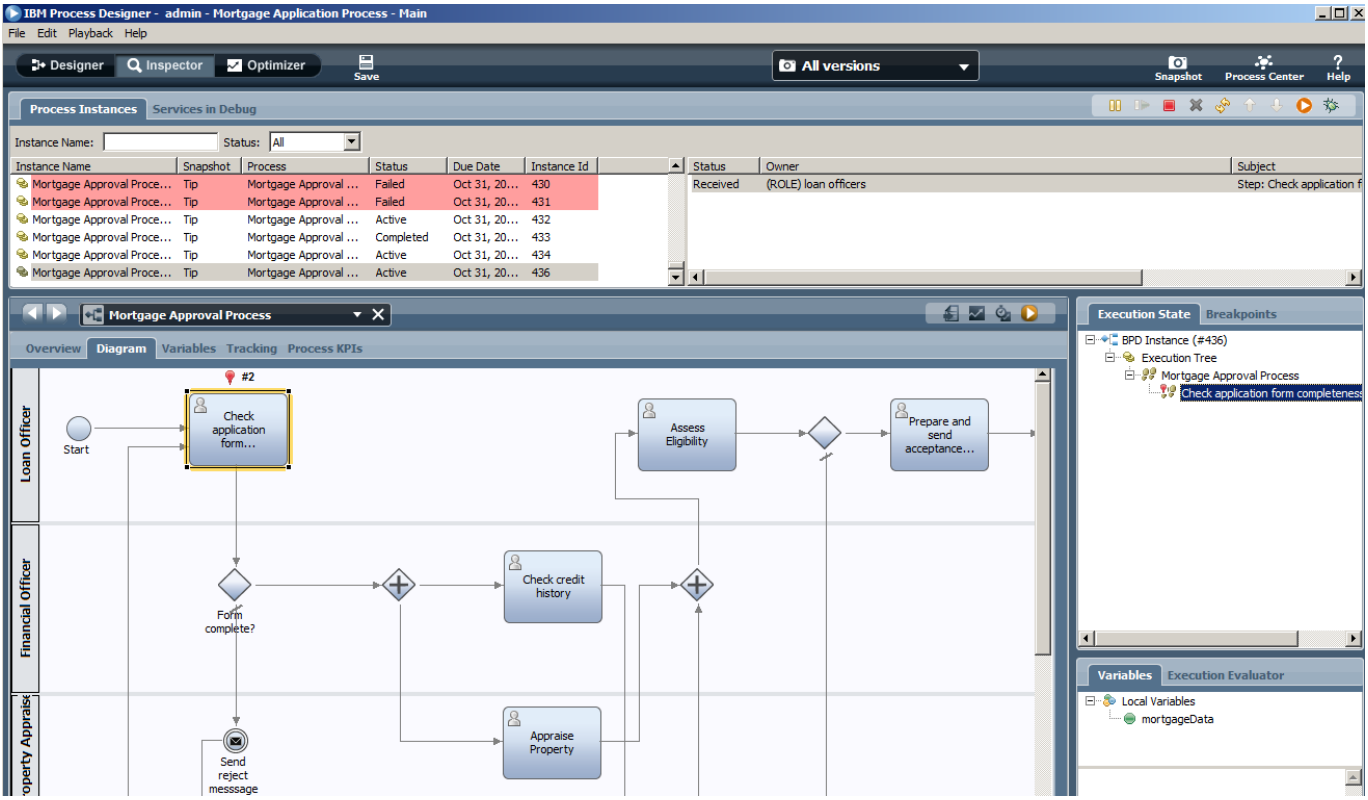
Mortgage_Application_Process - Part_3_Completed_Process.twx

2.8 Testing & Deployment

At this stage we have finished the process definition and we're ready for testing. Testing can be done locally as the Process Centre includes a Process Server runtime. Alternatively we can deploy the process to test on a remote Test server. For quick debugging purposes we'll now perform a local test. On the process map view, click the orange Play button.

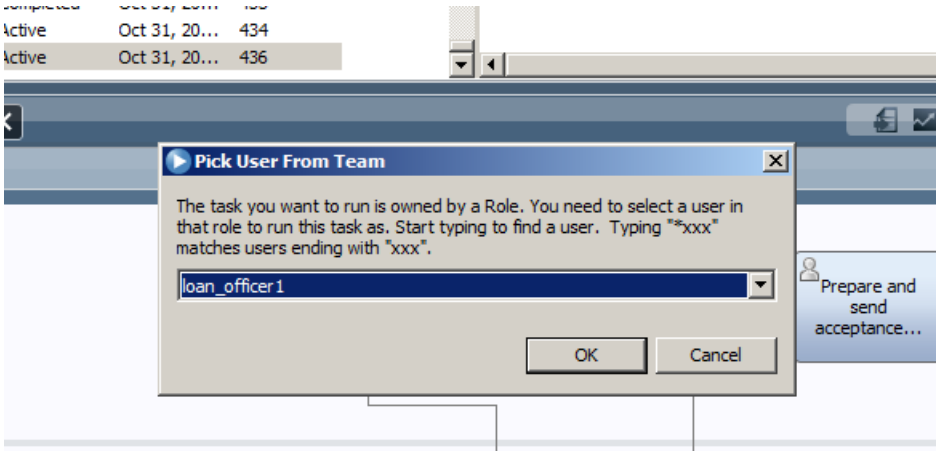


This will switch the designer into the "Inspector" view where we can step through the process.

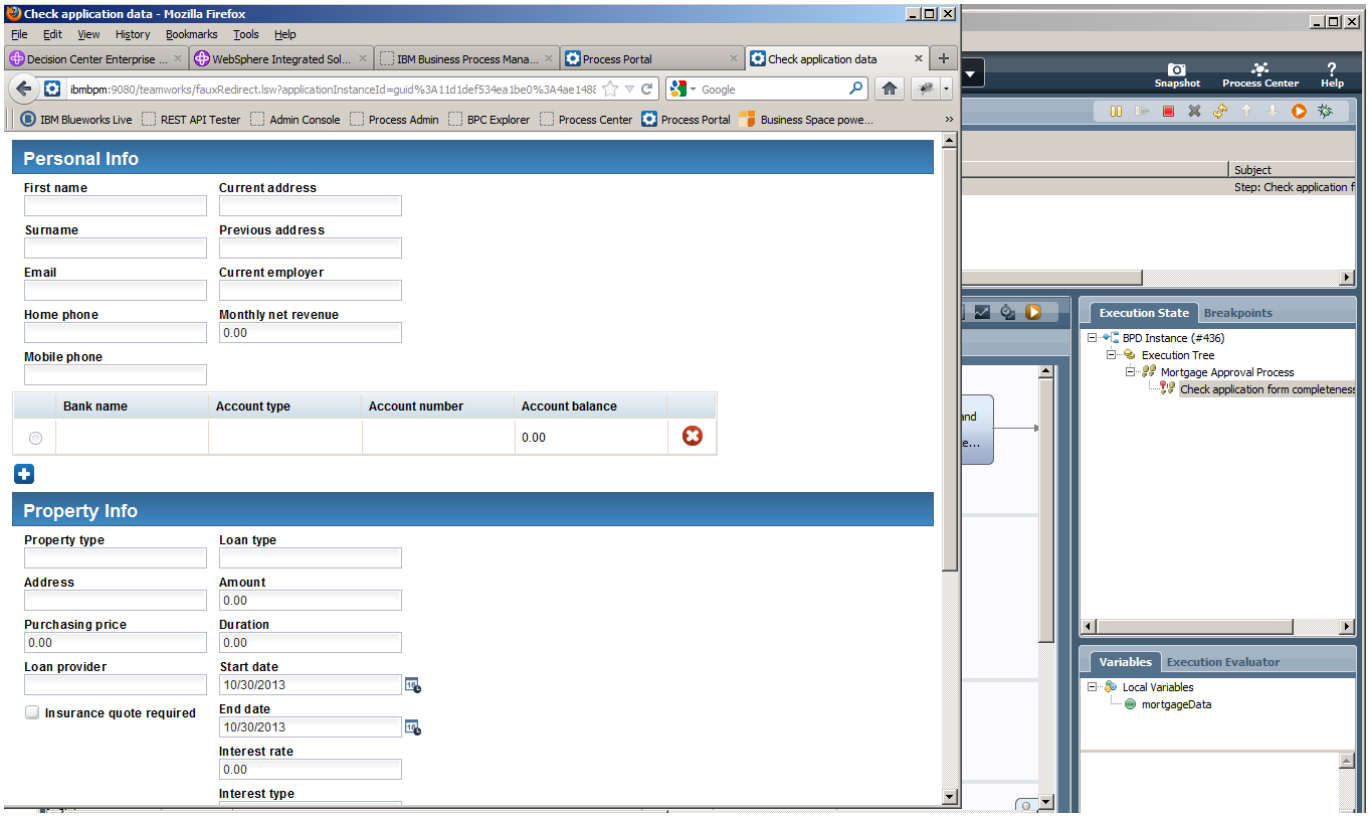


The list of instances appears on the right, click on the active instance to show the individual steps on the right. The process diagram will appear at the bottom. At the bottom right the process variables can be inspected.

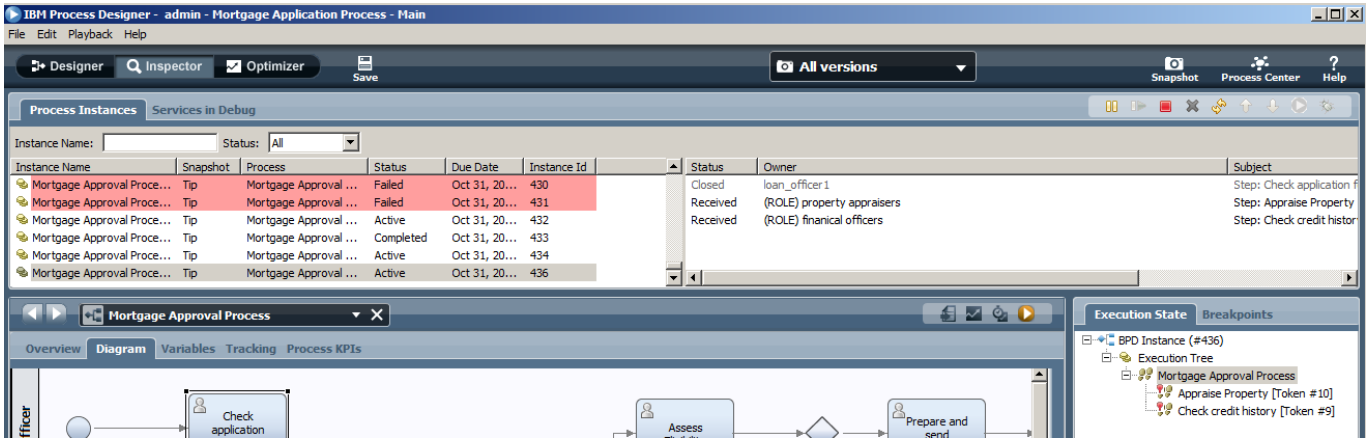
After selecting an activity on the top-right, click the orange play button to launch the associated task:



If a team has been associated with a task, a user picker will show up. This allows the tester to assume the roles of multiple people without logging in and out all the time.

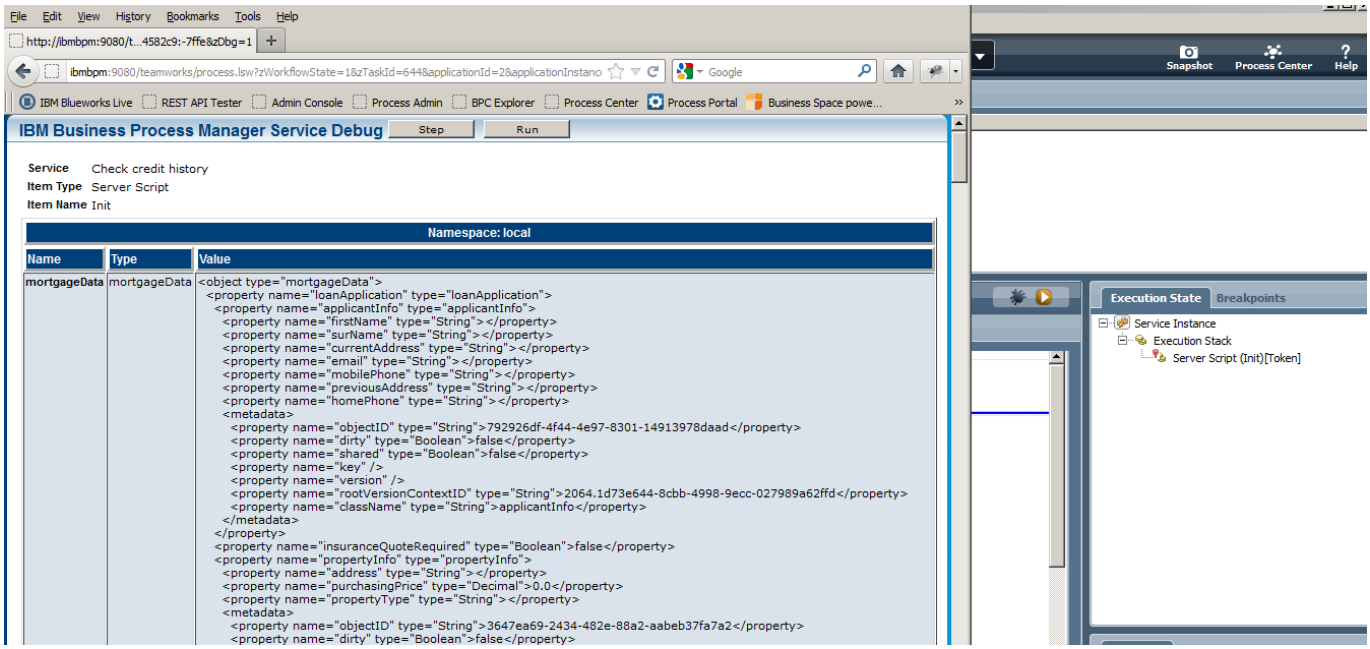


The coach will launch in the browser and the tester can complete the coach as normal.

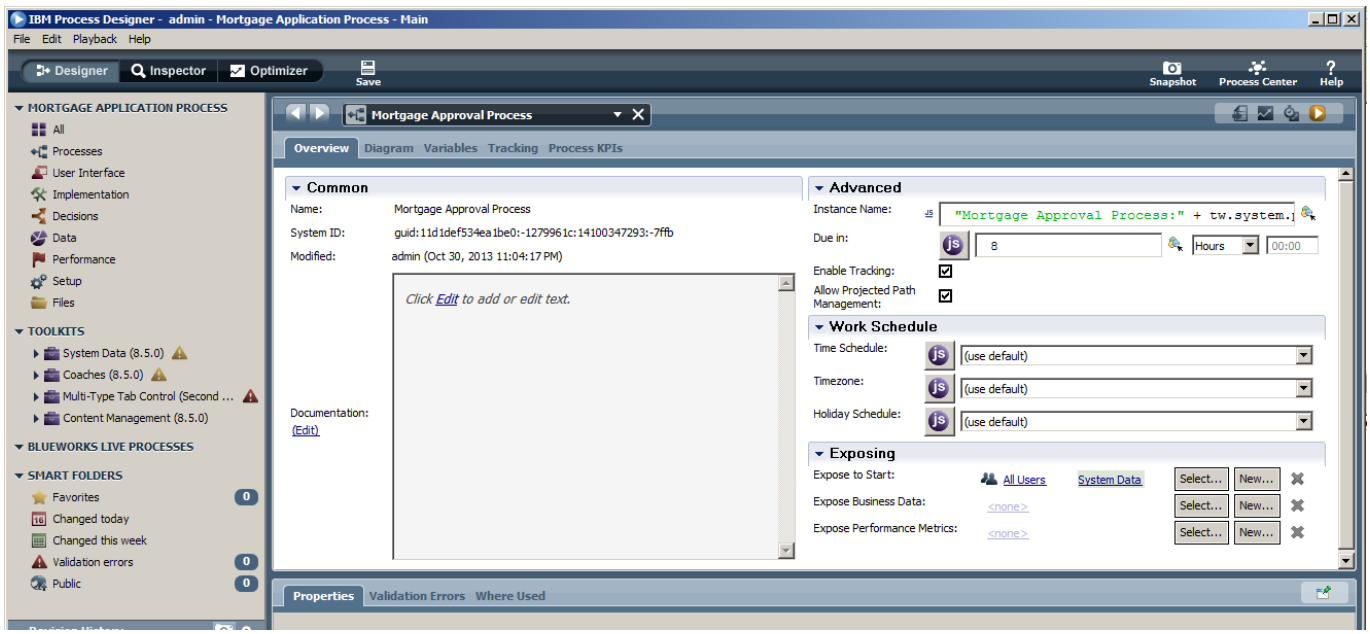


Press the yellow refresh button to make the next steps appear. Clicking on each task and playing each task back will allow the tester to go through the entire process.

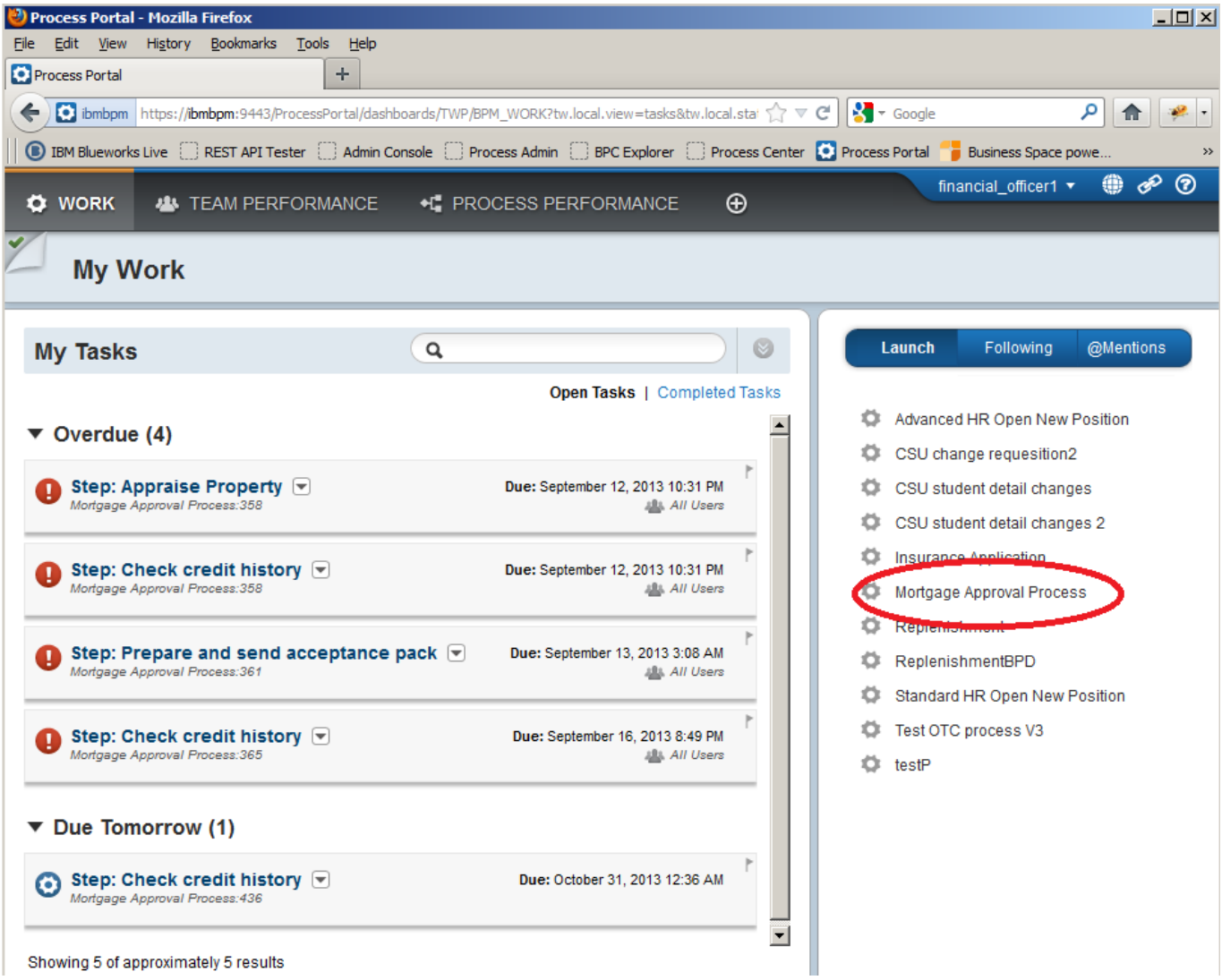
Instead of the playback button, the 'debug' button can be used to execute the tasks via the debug screens:



The debug view shows a detailed view of the variable at each step of the execution. It allows the tester to do a step by step run of the process.



The process can be deployed to the Process Portal, by specifying who can start the process: select the value "All Users" to the "Expose to Start" field.



After logging into the process portal, the Mortgage Approval Process is now visible and can be started by clicking on it.

Congratulations, you've finished the lab!

2.9 Further reading

2.9.1 Business Resources

IBM BPM 8.5 youtube video's
https://www.youtube.com/playlist?list=PLBC07B35CC4847FF7

BPM for Dummies ebook
http://public.dhe.ibm.com/software/in/events/softwareuniverse/resources/BPM_for_Dummies.pdf

Creating a BPM Centre of Excellence ebook
www.redbooks.ibm.com/abstracts/redp4898.html

2.9.2 Technical Resources

Developerworks technical forum:
https://www.ibm.com/developerworks/community/forums/html/forum?id=11111111-0000-0000-0000-000000002382

IBM BPM Wiki Sample Exchange:
http://bpmwiki.blueworkslive.com/display/samples/SAMPLE+EXCHANGE+HOME

PDF Generation toolkit
http://bpmwiki.blueworkslive.com/display/samples/PDF+Generation+Toolkit

HTML 5 Mobile interface
http://bpmwiki.blueworkslive.com/display/samples/Mobile+BPM+%28Web+Sample%29

BPM 8.5 Detailed System requirements
http://www-01.ibm.com/support/docview.wss?uid=swg27023007

BPM 8.5 Product documentation
http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.main.doc/ic-homepage-bpm.html

IBM Support portal ticket system
http://www-947.ibm.com/support/entry/portal/overview

Neil Kolban BPM ebook
http://www.neilkolban.com/IBM/

Detailed tutorial video's
http://www.neilkolban.com/IBM/videos.html