

IBM BPM V8.5 Standard

Consistent Document Management

Lab Exercises – Version 1.0

Author: Sebastian Carbajales



An IBM Proof of Technology

Catalog Number

Contents

LAB 9	IBM BPM V8.5 STANDARD – CONSISTENT DOCUMENT SUPPORT	7
9.1	LAB SETUP	7
9.2	PART I – ADD DOCUMENT EVENTS	7
9.3	PART II – COMPLETE COACHES TO DISPLAY BPM DOCUMENT	23
9.4	PART III – TEST THE IMAGE SUBMISSION PROCESS APP	37
9.5	SUMMARY	42

Overview

In V8.5 the IBM BPM document support has been significantly enhanced. You can use Enterprise Content Management (ECM) tools to work with IBM BPM documents in the embedded IBM BPM document store. The IBM BPM document store is a CMIS-enabled embedded document repository that is used to store IBM BPM documents. As a result it supports CMIS operations and inbound events. That means that now you can use the Content Integration node, found in the service editor pallet, to work with IBM BPM documents.

In addition, the Document List and Document Viewer coach views, shipped in the Content Management toolkit, can now also be used to work with IBM BPM documents and replace the Heritage Document Attachment and Document Viewer controls.

Lastly, event subscriptions can also be created against the embedded IBM BPM document store.

In this hands-on lab you will explore the new IBM BPM document capabilities available in Process Designer.

Lab Introduction

You will use IBM Process Designer throughout this lab. The process app you will be working on is partially built. The end result will be to provide functionality to submit and review an image document. The process app uses document events to coordinate the work between two processes. It also uses the coach views, in the Content Management toolkit, to implement the various UIs.

In the first part of the lab, you will work on adding the document events to the two pre-built processes.

The second part of the lab focuses on completing the UI for the various tasks. You will also build a dashboard to show submitted images.

Finally you will use the Inspector view in Process Designer to test your solution.

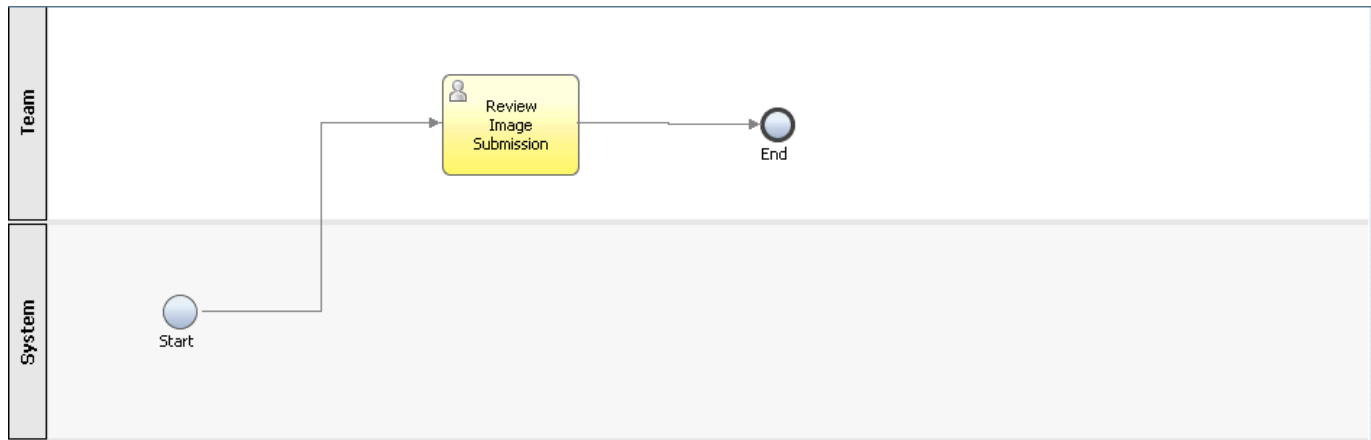
Scenario Introduction

In this lab you will be completing the implementation of two processes: Submit New Image and Review Image Submission.

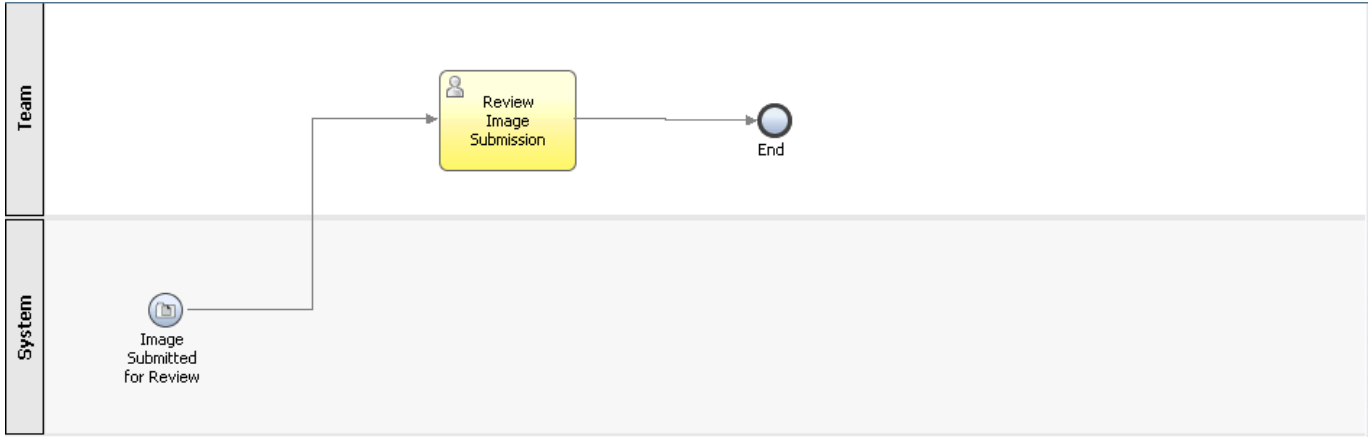
The Submit New Image process can be started by any user and provides a task to capture the new image to be reviewed as an IBM BPM document.



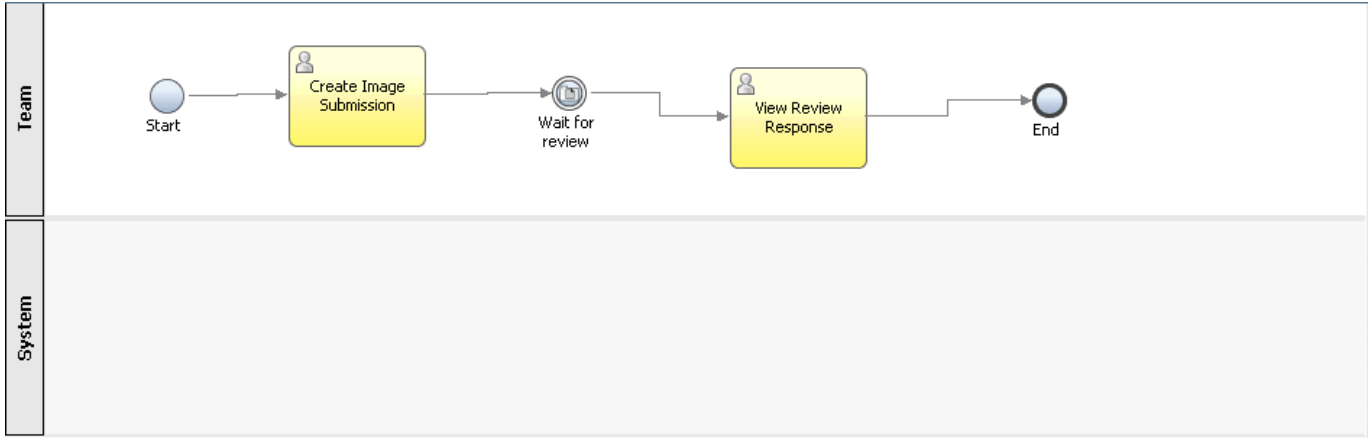
The Review Image Submission process is to be invoked as a result of a new image submission and provides the task to review the image document. The user can approve the submission and also add a comment. The pre-build process, however, has no way to start or get the image document to review.



The objective of the lab is to add a content start event to the Review Image Submission process so that it can be automatically started as soon as a new image document is submitted for review. The image below shows the to-be version of this process.



Finally, the Submit New Image process is modified with an intermediate event to wait for the review to be completed and then allow the submitter to see the response using a new task. The to-be version of this process is shown below.



Lab 9 IBM BPM V8.5 Standard – Consistent Document Support

9.1 Lab Setup

9.1.1 Start Process Server

If you have already started Process Server a previous lab, please move on to the next section.

- __1. Start Process Server
- __2. Wait for the Command window to close before proceeding to the next step

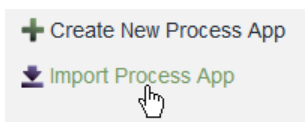
9.1.2 Start Process Designer

If you have already started Process Designer a previous lab, please move on to the next section.

- __3. Start Process Designer
- __4. For both *User Name* and *Password* enter **admin** and click **Login**

9.1.3 Import Consistent Document Support Lab Process App

- __1. Click **Import Process App**



- __2. Navigate to **Image_Submission - Start.twx** and click **Ok**



9.2 Part I – Add Document Events

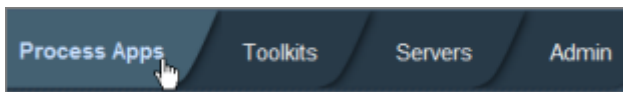
In this part of the lab you will add the event handling to the **Submit New Image** and **Review Image Submission** processes. You will also create the supporting artifacts for each of the events handled in these processes.

You will start working on the **Submit New Image** process by adding an intermediate content event that will wait for the image to be reviewed. You will also define the event subscription that triggers the intermediate content event.

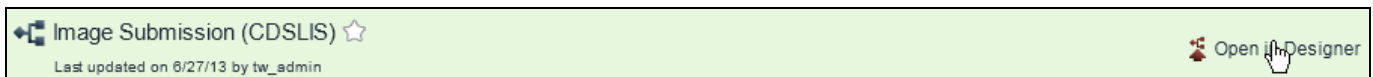
Then you will move on to the **Review Image Submission** process and add a content start event to kick off the process when a new image document has been submitted for review. As before, you will also create the event subscription to link to the start event.

9.2.1 Open the Image Submission Process App

- __1. Switch to **Process Designer**
- __2. Click the **Process Apps** tab



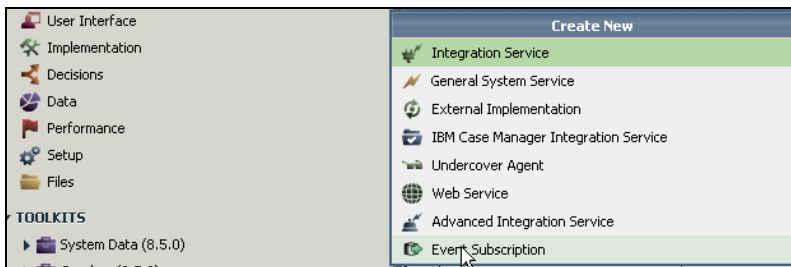
- __3. Select **Image Submission** Process App and click **Open in Designer**



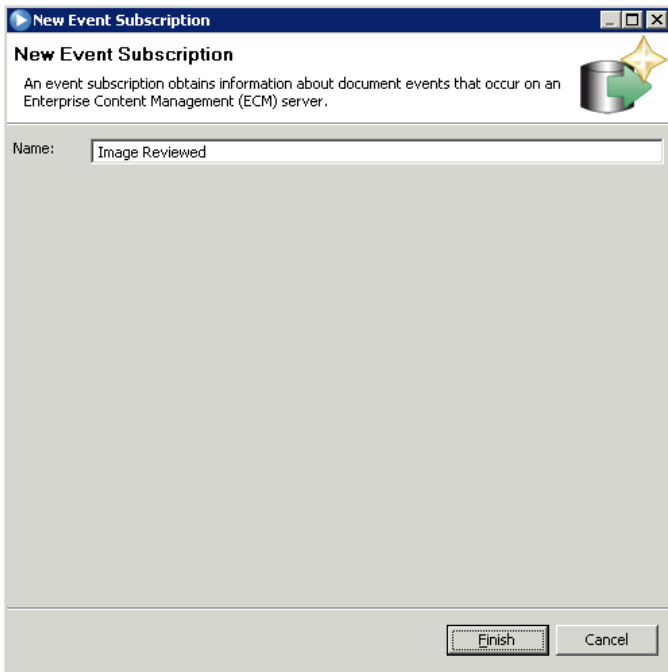
9.2.2 Subscribe to Update Document Events

In order to trigger a content event in a BPD an event subscription must be defined.

- __1. Hover over the **Implementation** category and click the add button. Select **Event Subscription**.

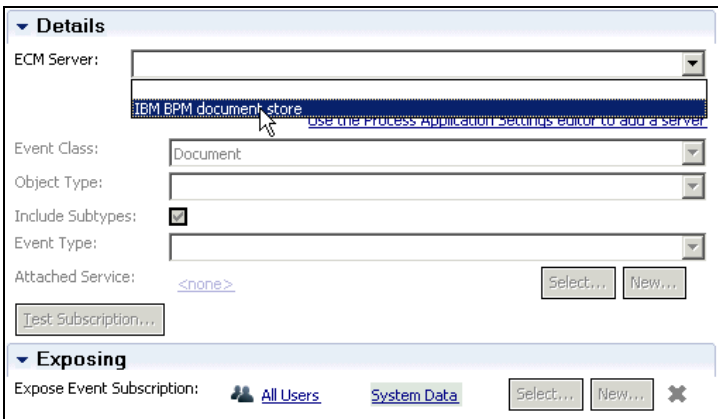


__2. Enter **Imaged Reviewed** for name and click **Finish**.



The Event Subscription editor opens. You are now ready to start defining the event subscription attributes.

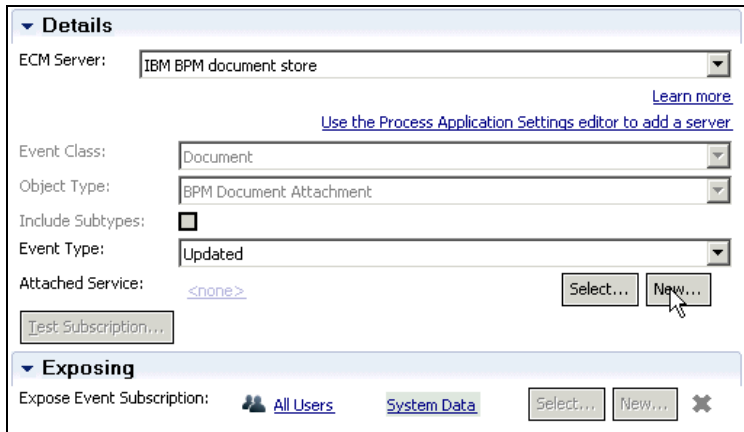
__3. Set the **ECM Server** in the **Details** section. Click on the selection box and pick **IBM BPM document store**.



Notice that many of the fields in the **Details** section are disabled. These fields are not configurable for the **IBM BPM document store**. These become enabled if you target an external ECM server. At this point however, you are only able to set the **Event Type** and **Attached Service**.

__4. Click the **Event Type** selection box and then pick **Updated**.

__5. Click the **New** button for **Attached Service**.

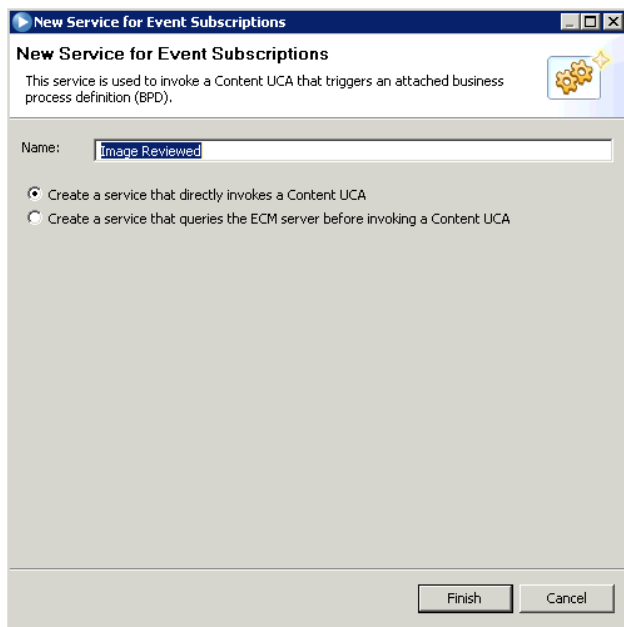


You will create a new service which will be invoked when an update document event occurs, as selected in the previous step.

__6. Accept the defaults and click **Finish** on the wizard.

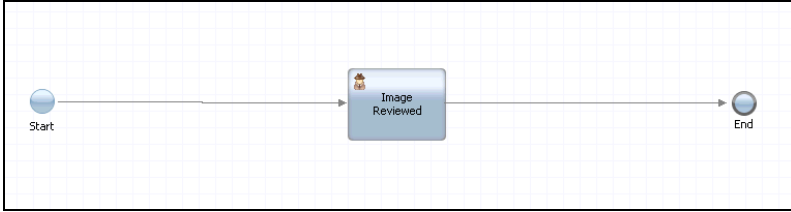
The new service wizard opens up prefilled with a name to match the event subscription. It offers two flavors of the service to be created: a simple service that directly invokes an UCA, or a service that first evaluates the event contents to determine if the event should be emitted.

In this case the first option is acceptable. This event will be consumed by an intermediate content event which will filter based on correlation, so there is no need to pre-process the event contents at this point.

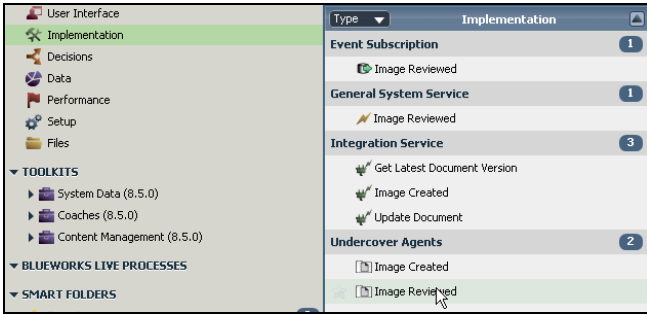


__7. Examine the **Image Reviewed** service.

A simple service with a single **Invoke UCA** node that will invoke the **Image Reviewed** UCA, also created along with this service.

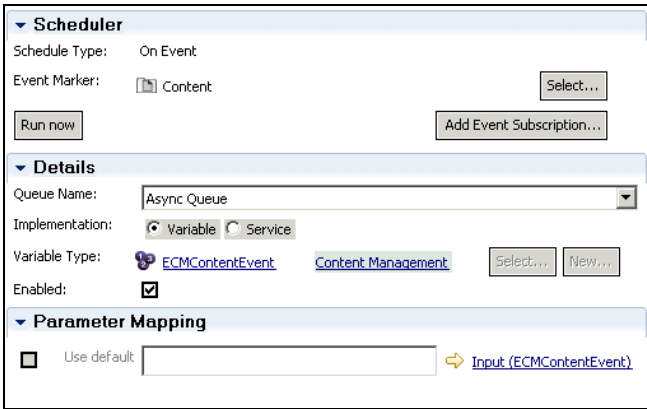


__8. Click on the **Implementation** category and double click on the **Image Reviewed** UCA.

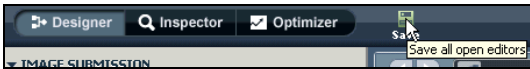


__9. Examine the **Image Reviewed** UCA.

Notice, in the **Scheduler** section, this is marked as a **Content** UCA. In the **Details** section, the UCA is automatically associated with an **ECMContentEvent** type. A property of this type will be later used for correlation in the **Submit New Image** process.

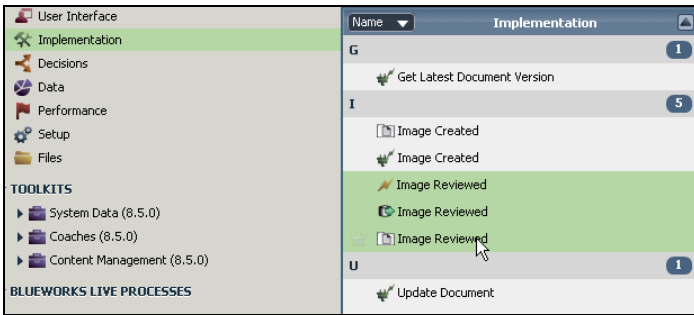


__10. Click the **Save** button to save the **Image Reviewed** event subscription.



__11. The event subscription definition is done. Click on the **Implementation** category to see the artifacts.

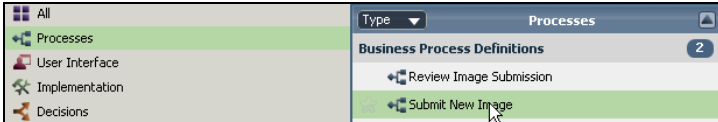
Notice you now have three artifacts named Image Reviewed: an Event Subscription, a General System Service and an UCA.



9.2.3 Add Intermediate Event to the Submit New Image Process

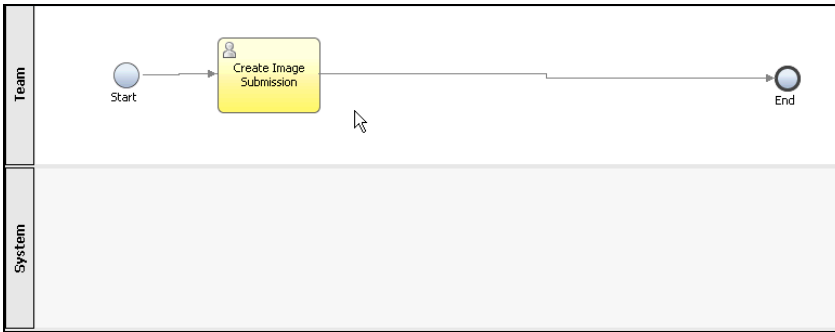
At this point we have defined the event subscription and UCA required to consume the content event. Now we need to add the intermediate event to our process.

1. Click the **Process** category and double click **Submit New Image** process.



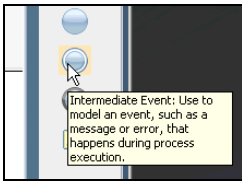
__2. **Examine** the process.

You will add a new intermediate content event and a new task between the **Create Image Submission** and **End** nodes. The event will trigger when the document, created in the **Create Image Submission** node, is updated in the **Review Image Submission** process. The new task will then become available to review the response.

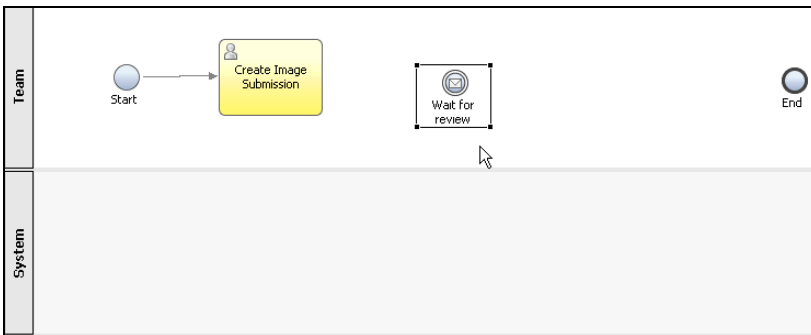


__3. Remove the link between **Create Image Submission** and **End** nodes.

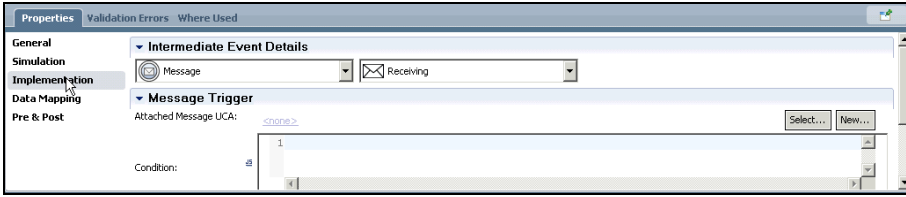
__4. Drop an **Intermediate Event** node from the palette onto the canvas.



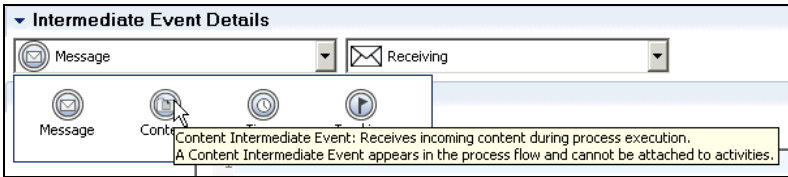
By default, a Message Event is created. Name the event node **Wait for review**.



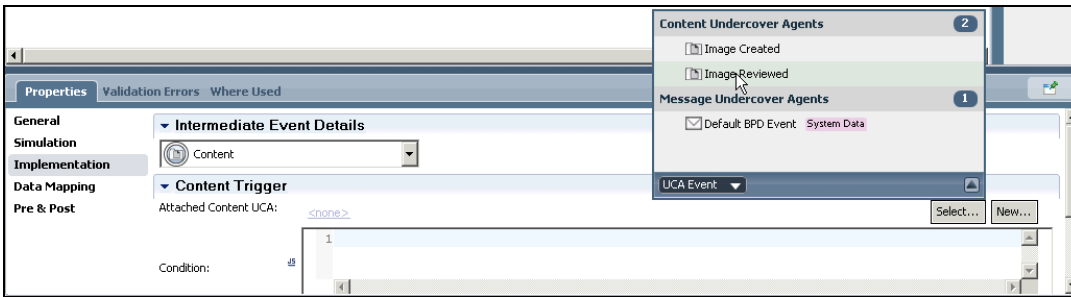
__5. Select the new Wait for review event node. Click on the **Properties** tab and then **Implementation** tab



__6. Click the first selection box under **Intermediate Event Details** and select **Content**.

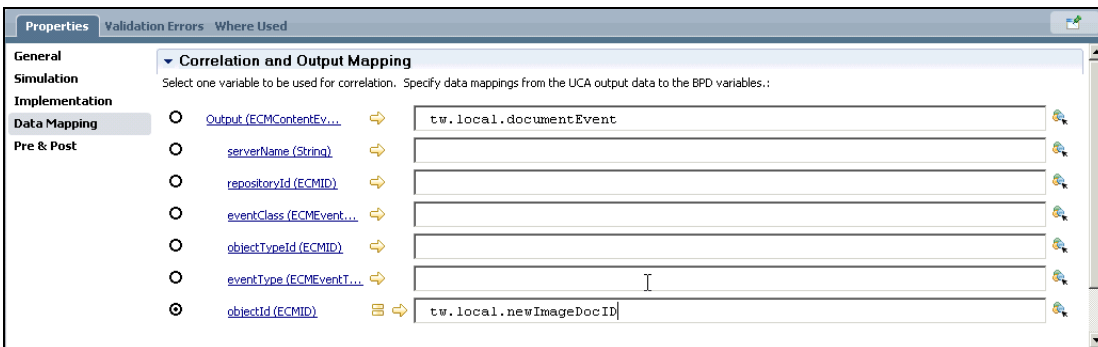


__7. Click the Select button in the **Content Trigger** section, to select an UCA, and pick **Image Reviewed**.



__8. Click on the **Data Mapping** tab and assign the variables as shown below.

The top level Output is mapped to the **tw.local.documentEvent** variable so that we can use the event contents later. However, correlation is done on the **objectId** (notice the radio button selected on the left), and the event will only fire if the incoming value of **objectId** matches the content of **tw.local.newImageDocId**. This variable is set to the output of the first activity, **Create Image Submission**.

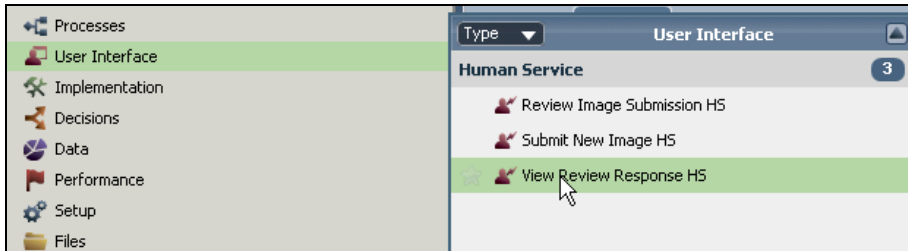


__9. **Save** the process.

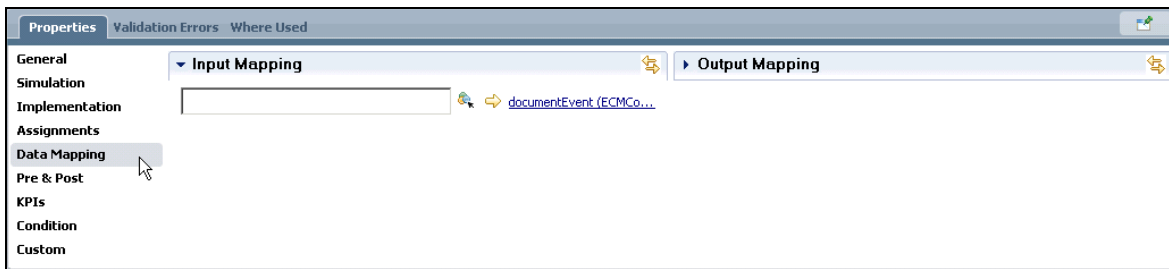
9.2.4 Add Review Task to Submit New Image Process

You complete the process by adding the task used to review the response.

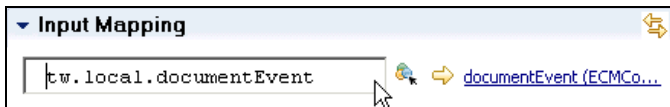
1. Click on the **User Interface** category in the library, drag and drop the **View Review Response HS** human service onto the process canvas.



2. Select the **View Review Response HS** node. Click on the **Properties** tab and then the **Data Mapping** tab.

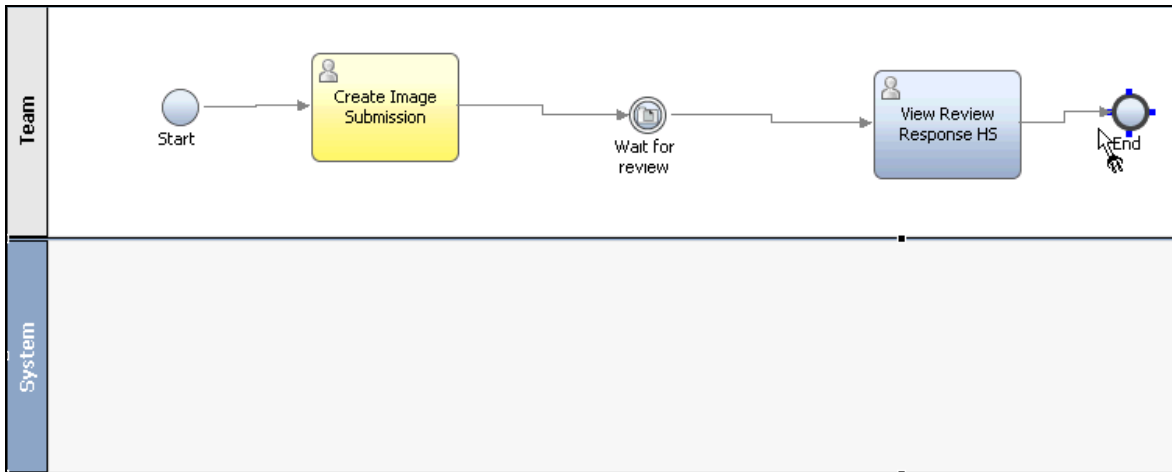


3. Set the input mapping to **tw.local.documentEvent**.



4. On the process canvas, create a link between **Create Image Submission** and **Wait for review** nodes.
5. Create another link between **Wait for review** and **View Review Response HS** nodes.
6. Create one last link between **View Review Response HS** and **End** nodes.

At this point the **Submit New Image** process is complete.

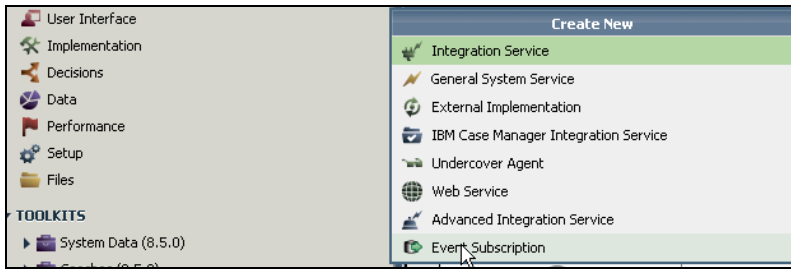


__7. **Save** the process.

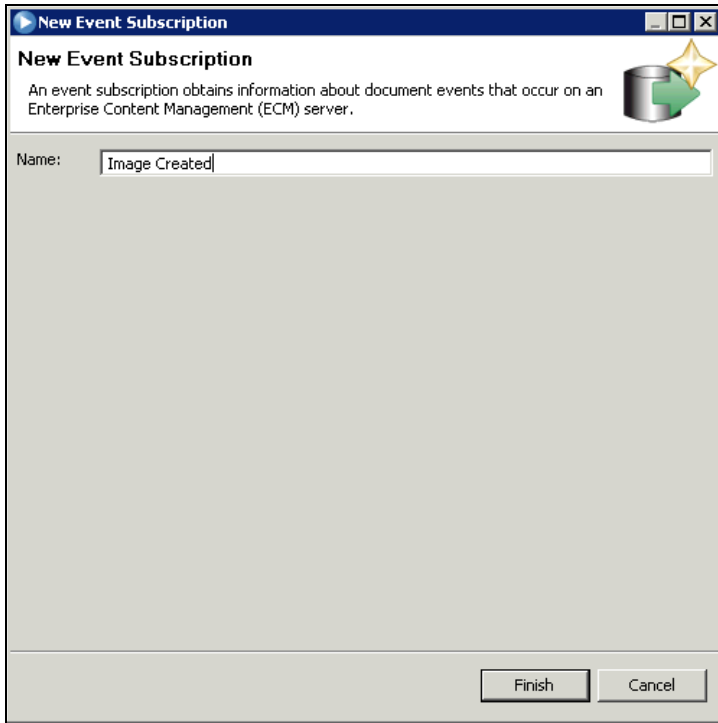
9.2.5 Subscribe to Create Document Events

You will now focus on the **Review Image Submission** process. It needs to be modified so that it can be started by a create document event. As in the previous case, the first step is to create an **Event Subscription**.

__1. Hover over the **Implementation** category and click the add button. Select **Event Subscription**.

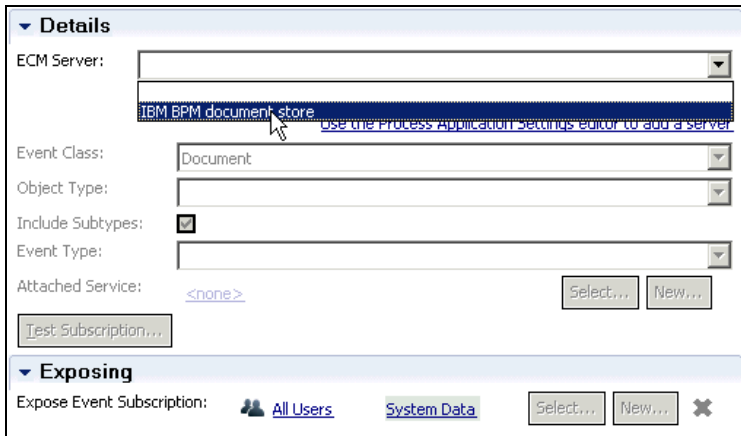


__2. Enter **Imaged Created** for name and click **Finish**.



The Event Subscription editor opens. You are now ready to start defining the event subscription attributes.

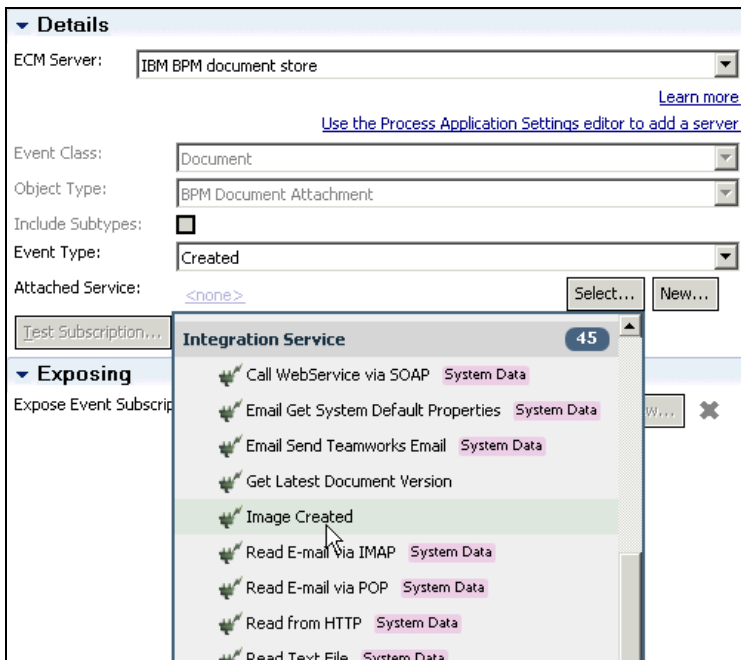
__3. Set the **ECM Server** in the **Details** section. Click on the selection box and pick **IBM BPM document store**.



__4. The **Event Type** is set to **Created** by default.

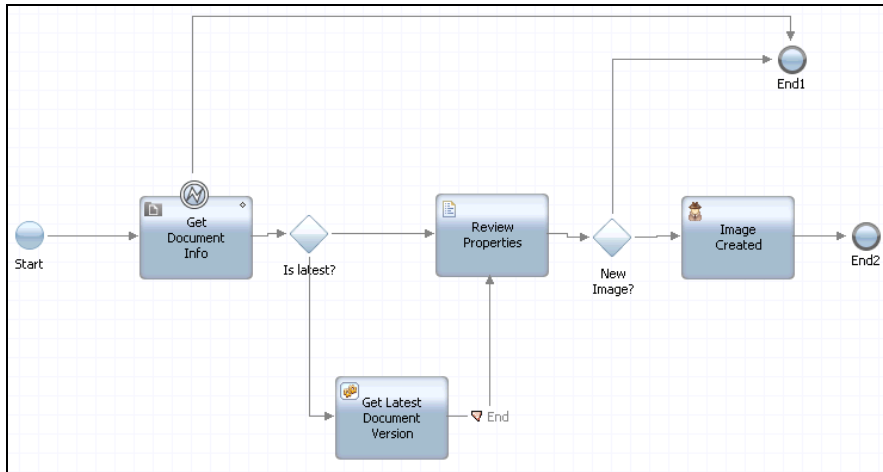
__5. Click the **Select** button for **Attached Service**. Pick **Image Created** service.

The created event will be handled by a pre-built service.

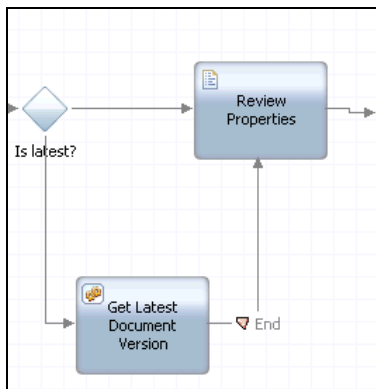


- __6. **Save** the event subscription.
- __7. Click the **Image Created** link to open the editor.
- __8. **Examine** the service.

This service was originally created by selecting the second pattern in the new service wizard.



The generated service was modified by adding the invocation to get the latest version of the document and the node to retrieve its properties.



The finished service does the following:

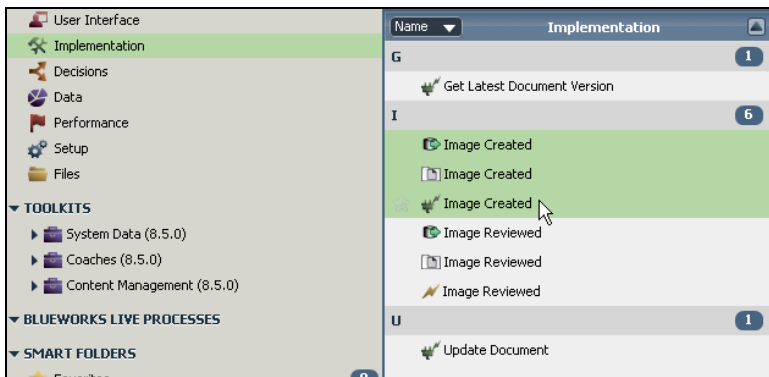
- The **Get Document Info** node retrieves in document from the **IBM BPM document store** using the document ID received from the content event input.
- The **Is Latest?** decision node checks a property of the document to determine if it is the latest version. If it is, it continues to the **Review Properties** node. If it's not, it calls the **Get Latest Document Version** service to get it before proceeding.
- The **Review Properties** node will parse the **IBM BPM document properties** and look for the 'status' property. If the value of it is 'new', it will set the **tw.local.newImage** to true will cause the **Image Created** node to be executed. Otherwise the process ends. Spend some time

examining the code in the **Implementation** tab of this node to understand how these properties are retrieved.

The **Image Created** node will trigger the **Image Created** UCA which will be hooked up to the **Review Image Submission** start event.

___9. The event subscription definition is done. Click on the **Implementation** category to see the artifacts.

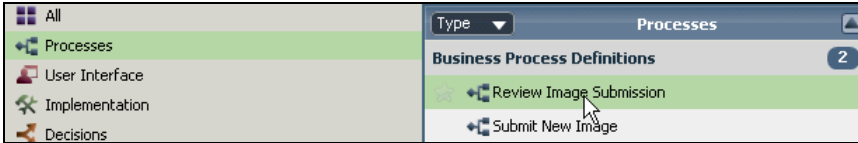
Notice you now have three artifacts named Image Created: an Event Subscription, an Integration Service and an UCA.



9.2.6 Add Content Start Event to the Review Image Submission Process

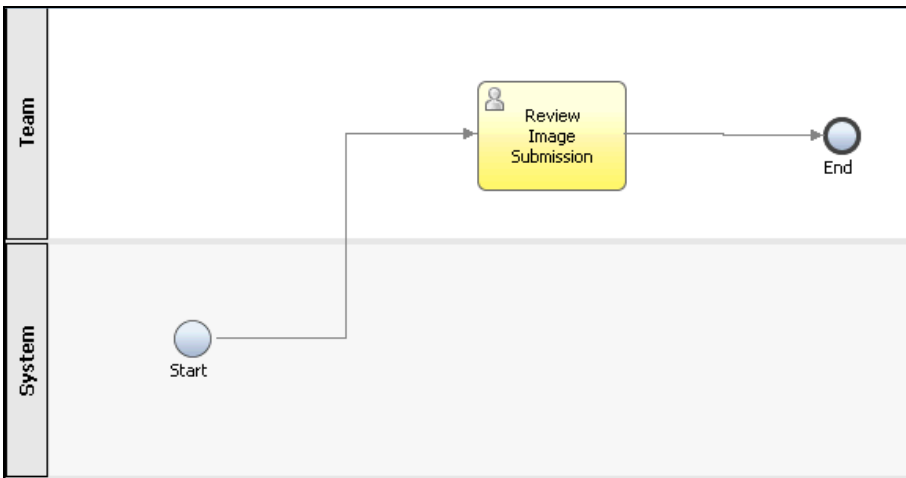
The created content event subscription has been defined. We now add the start event that will trigger the review process.

___1. Click the **Process** category and double click **Review Image Submission** process.



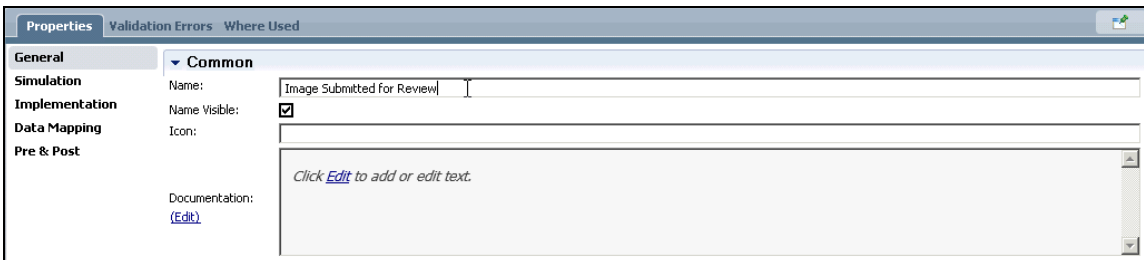
__2. **Examine** the process.

You will change the **Start** event type to be a content event and associate it with the right UCA.

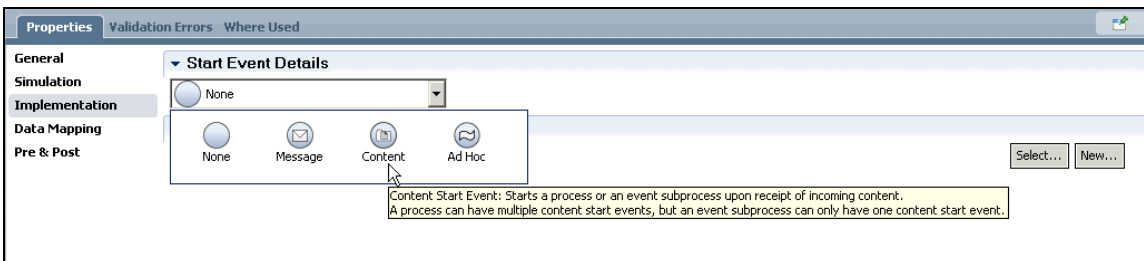


__3. Select the **Start**. Click the **Properties** tab and then **General**.

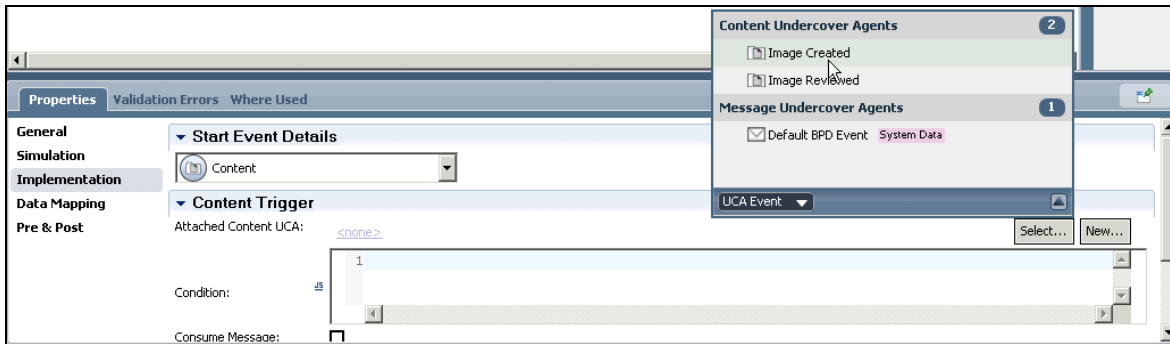
__4. Update the name to **Image Submitted for Review**.



__5. Still in the **Properties** tab, click on the **Implementation** tab and click on the selection box in the **Start Event Details** section. Select **Content**.

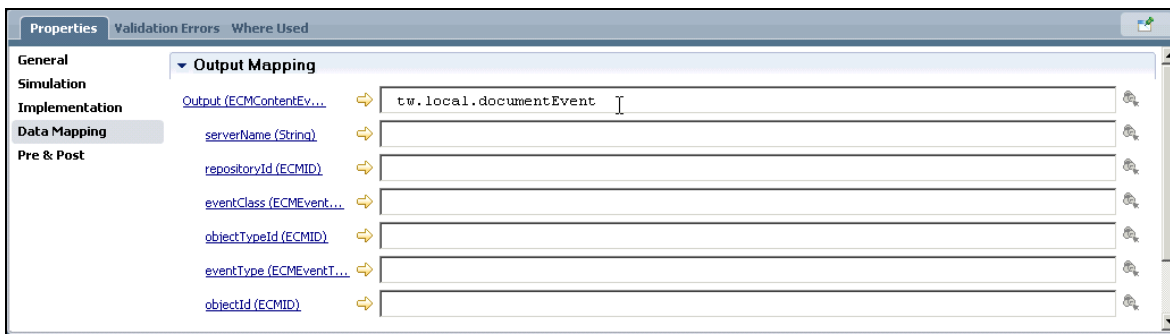


__6. Click the **Select** button in the **Content Trigger** section, to select an UCA, and pick **Image Created**.



__7. Click on the **Data Mapping** tab and assign the variables as shown below.

The top level **Output** is mapped to the **tw.local.documentEvent** variable so that we can use the event contents in the invocation of the **Review Image Submission** node.



__8. Select the **Review Image Submission** node. Click on the **Properties** tab and then on the **Data Mapping** tab.

__9. Set the input mapping to **tw.local.documentEvent.objectId**.

The content event object that was received by the **Image Submitted for Review** start event carries the new document's ID. This is required by the **Review Image Submission** activity to show the details of the document.



__10. **Save** the process.

The process changes are now complete.

9.3 Part II – Complete Coaches to Display BPM Document

In this part of the lab you will add missing bits to the various coaches in the two processes that you modified in the previous part.

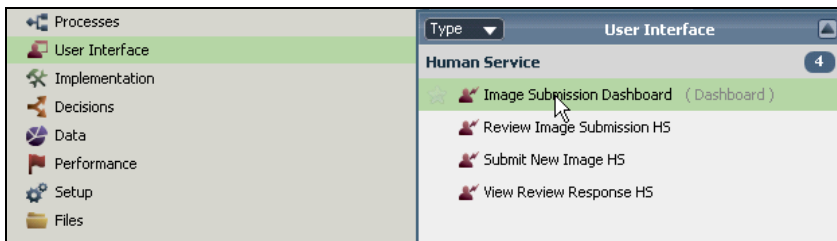
Specifically, the exercise will focus on adding and customizing the Document List and Document Viewer coach views as well as using the Content Integration node for IBM BPM document operations.

9.3.1 Creating an Image Submission Dashboard

Given that this is an image submission application, the first UI component you will add is a dashboard that will be able to display a list of documents in each state: to be reviewed, approved, rejected.

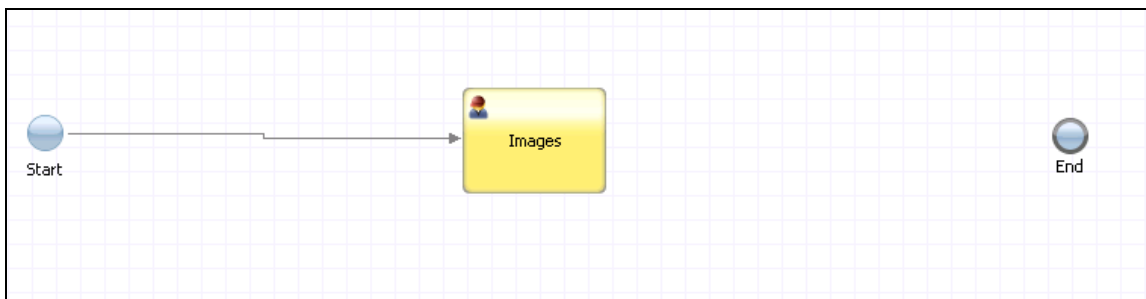
The **Image Submission Dashboard** has been partially built. It does not yet contain any of the document UI components.

1. Click the **User Interface** category and double click **Image Submission Dashboard** human service to open the editor.



2. **Examine** the service.

This is a human service exposed as a dashboard. It contains a single coach node which will implement the dashboard's UI.



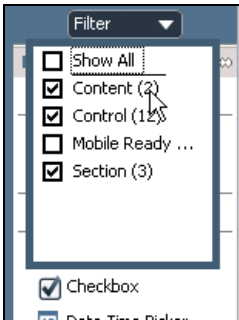
3. Double click on the **Images** node to bring up the coach editor.

The coach layout contains a **Tab** view which will host the three different lists of documents.

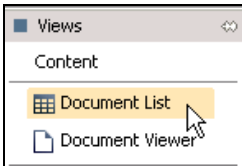


__4. Click on **Filters** on the palette and ensure **Content** is checked.

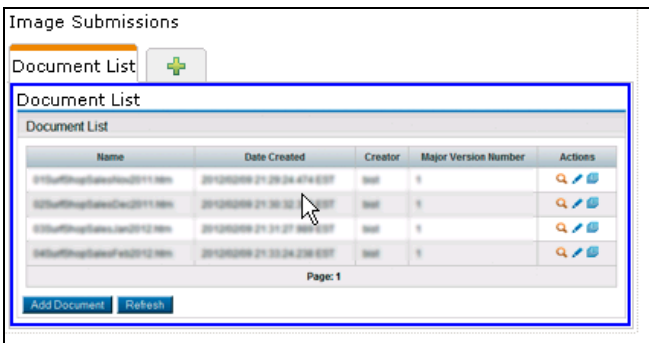
The **Content** views are not visible by default in the coach editor as the **Content Management toolkit** may not always be referenced.



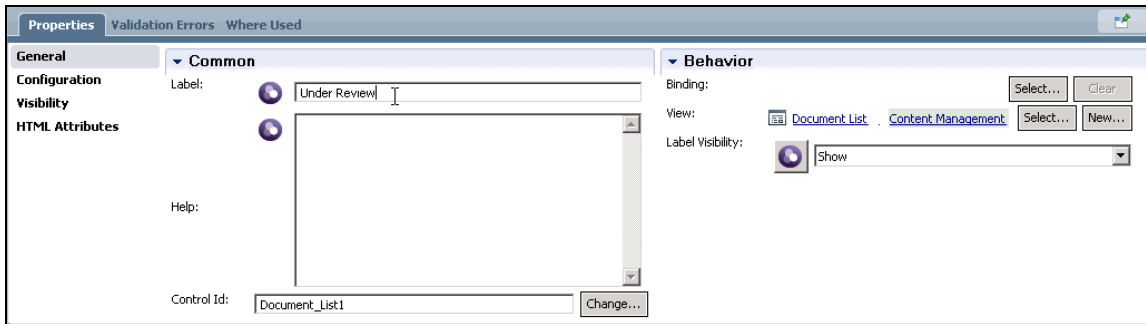
__5. From the palette, drag and drop a **Document List** view onto the content box in the tab view.



Once you drop it the coach layout should see the document list view in the first tab.



__6. Select the document list view then click **Properties** tab. Change the name to **Under Review**.

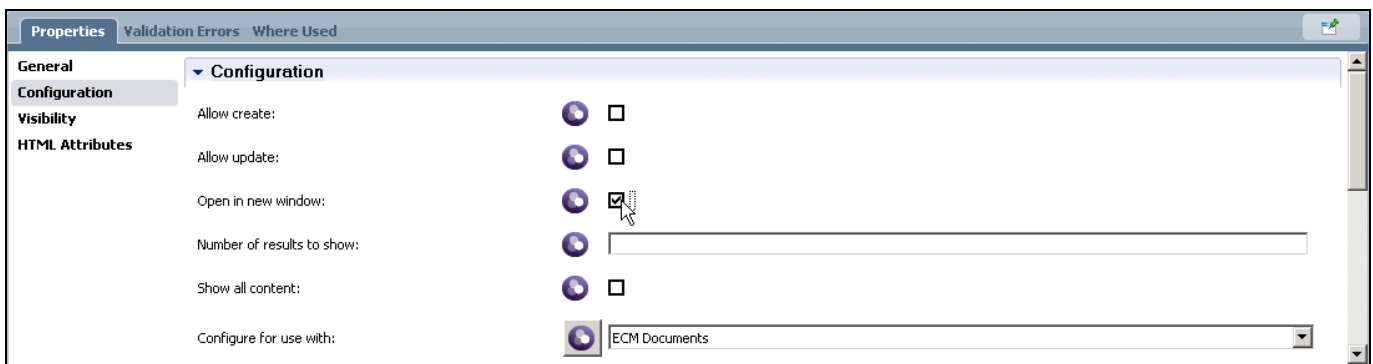


__7. Click on the **Configuration** tab.

You will now configure the view to display all image submissions that have not been reviews yet. Since this will be a read-only view we only need to configure it to display documents. As a result, we will not enable the create/update capabilities.

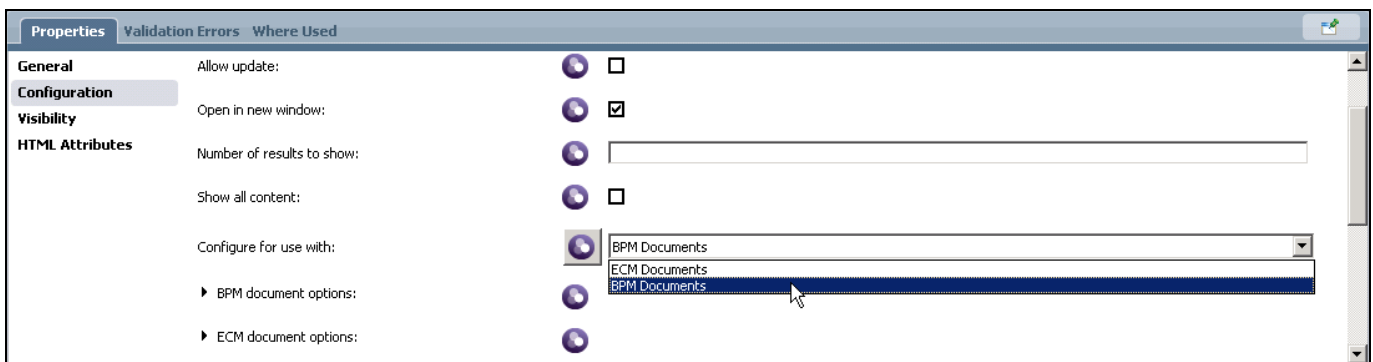
__8. Select the **Open in new window** checkbox.

This will create a **'view'** icon for each document in the list so that the user can open the contents and see them in a new window.



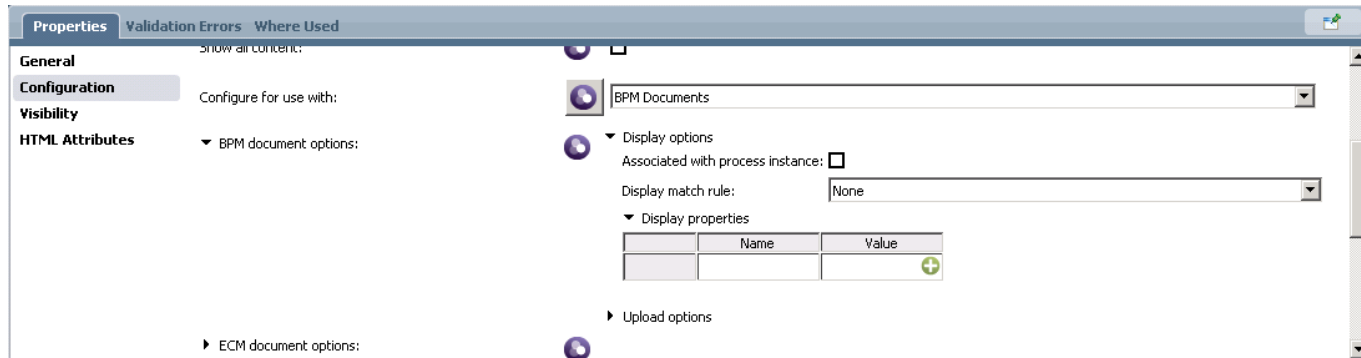
__9. Click on the selection box for the **Configure for use with** option and select **BPM Documents**.

Selecting either value does not change what other options you can specify in the UI. This choice will tell the coach view, at runtime, what other configuration settings to honor. In this case, the value of **BPM document options** will be used and **ECM document options** ignored.



___10. Expand the **BPM document options** option, then expand **Display options** and **Display properties** within it.

We will now set the display criteria for the view. We don't need to specify **Upload options** as we are not enabling this view to create, or update, documents.

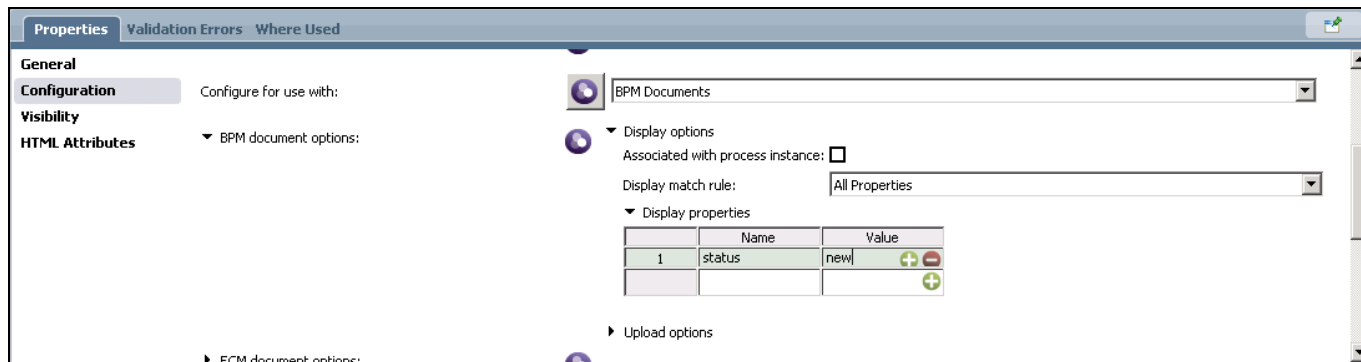


___11. Click the **Display match rule** selection box and select **All Properties**.

This ensures that all criteria specified in the **Display properties** section is met for a document to show up on the list.

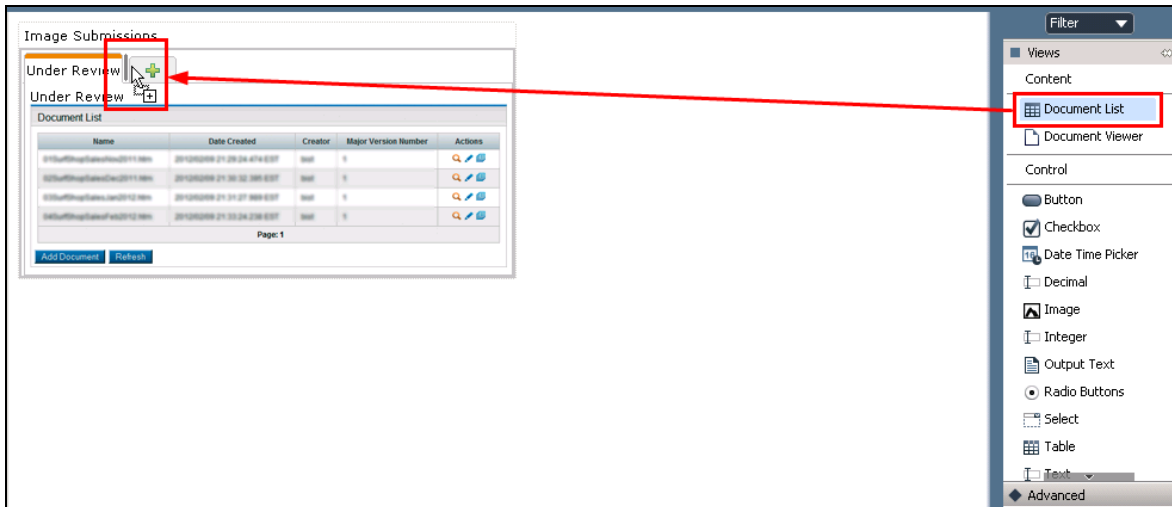
___12. Click on the plus sign in the **Display properties** table and type in 'status' for **name** and 'new' for **value**.

This tells the view to only display all documents that have an **IBM BPM document property** named **status** with a value of **new**. This property is created and set by the Document List view in the **Submit New Image HS**. You will review this later in the lab.



At this point the first view is configured. You will now configure two additional views, in the same way, to display approved and rejected image documents.

___13. Drag and drop two additional **Document List** views onto two new tabs.



__14. Repeat steps 6 through 11 to configure each view as described below.

Approved Document List:

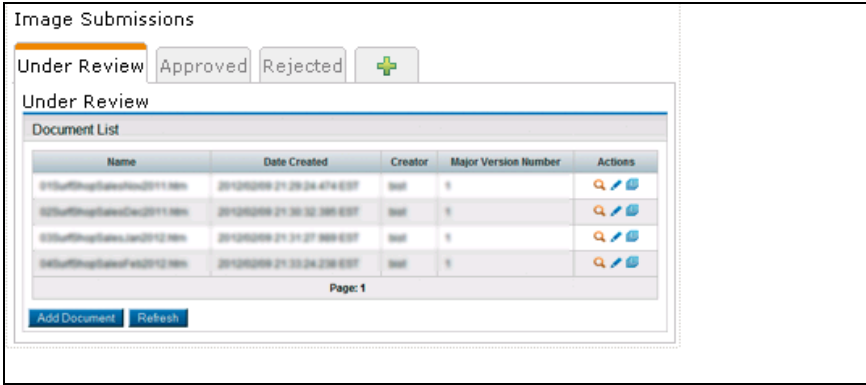
- **Name:** Approved
- **Display properties:**
 - status = reviewed
 - approved = true

Rejected Document List:

- **Name:** Rejected
- **Display properties:**
 - status = reviewed
 - approved = false

__15. **Save the Image Submission Dashboard.**

At this point the dashboard is completed. Its coach contains three tabs, each with a single **Document List** that will display image documents that are under review, approved and rejected.

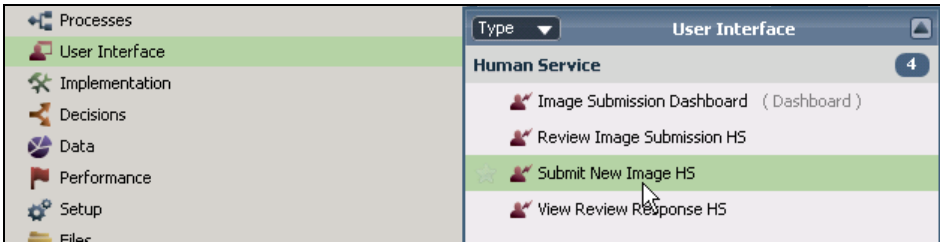


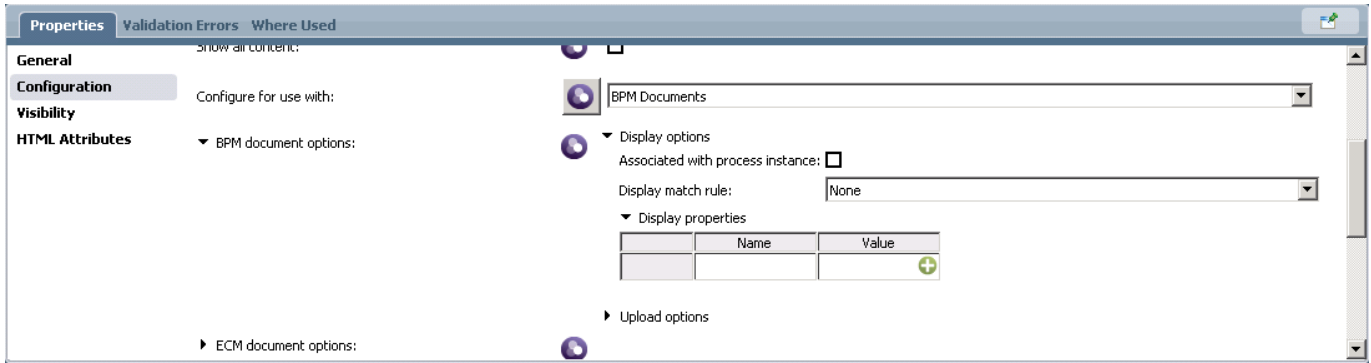
9.3.2 Updating the Image Submission UI

The **Submit New Image** process uses the **Submit New Image HS** human service to provide the interface that captures the submission information and creates the corresponding BPM document.

The configuration of the **Document List** within this service needs to be updated to add the correct **IBM BPM document properties**.

16. Click the **User Interface** category and double click **Submit New Image HS** human service to open the editor.





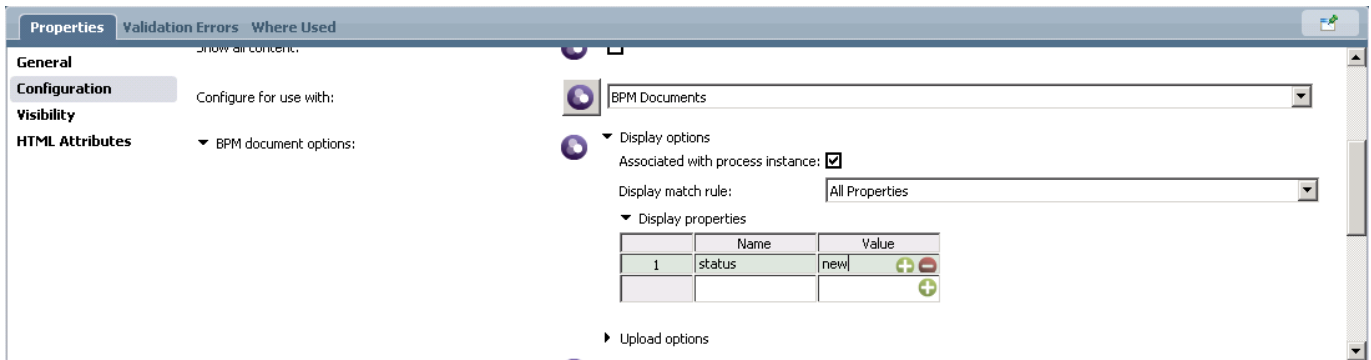
__22. Check the **Associated with process instance** checkbox

This ensures that only documents created within the process instance that launched this coach are displayed.

__23. Click the **Display match rule** selection box and select **All Properties**.

__24. Click on the plus sign in the **Display properties** table and type in '**status**' for **name** and '**new**' for **value**.

The display properties have been configured.

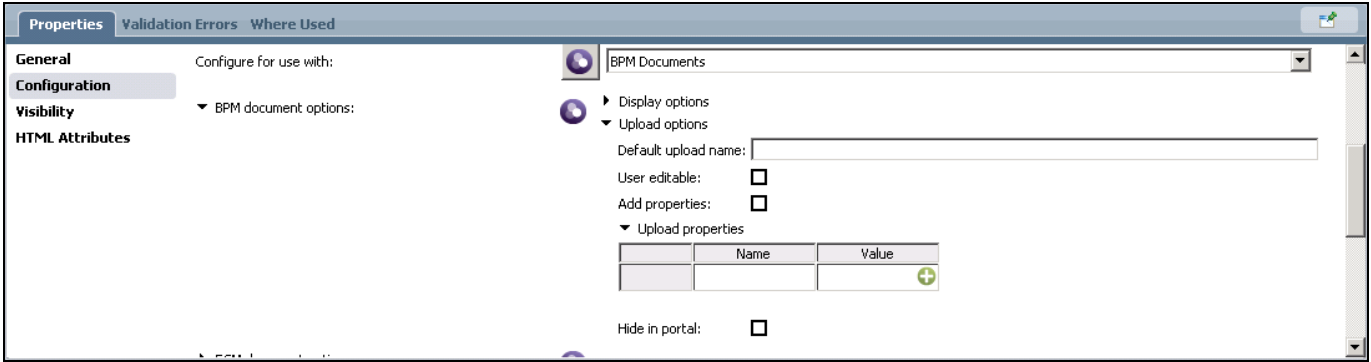


__25. Collapse **Display options** and expand **Upload options** and **Upload properties** within it.

You will now configure the upload options used when a new document is created.

The create capability is enabled by the **Allow create** option at the top of the **Configuration** tab. In this case, this option is bound to a **Boolean** variable, **tw.local.enableCreate**, that is set by the visibility rules associated with the view.

In our particular scenario, this service should only allow the creation of a single document. Once a document is created, the script in the visibility rule will set the **Boolean** to false effectively disabling the **Create Document** button.



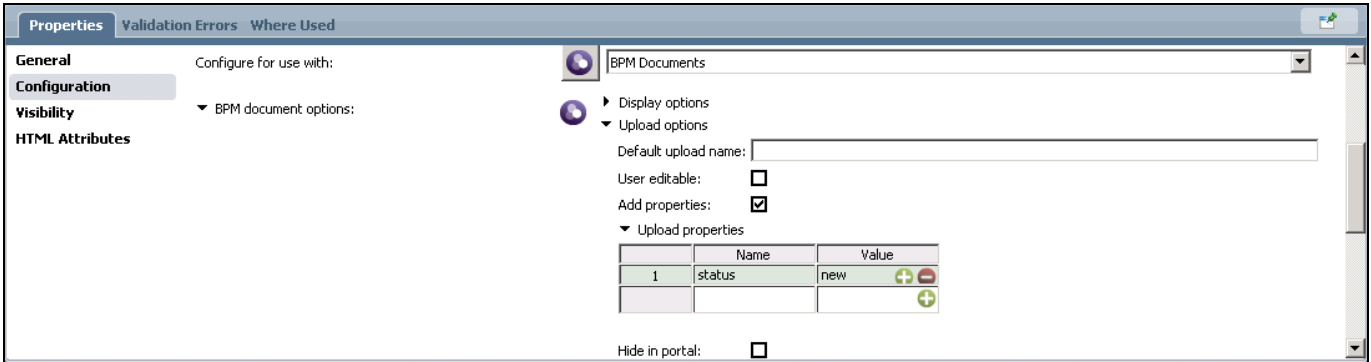
__26. Check the **Add properties** checkbox

This table holds the property name-value pairs that need to be added to all new documents created by this view.

__27. Click on the plus sign in the **Upload properties** table and type in '**status**' for **name** and '**new**' for **value**.

When a new document is created, it will automatically include the **status** property with a value of **new**.

The upload options have been completed.



__28. **Save** the service

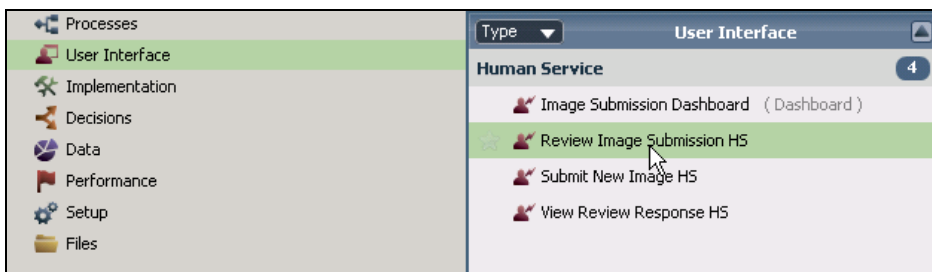
The service is complete. When a new document is created, an **ECMDocumentInfo** object will be stored in the binding of the **Document List** view (**tw.local.docInfo**). The **Extract Doc ID** node will take the first element in this list and extract the document ID to be returned by the service.

9.3.3 Updating the Review Image Submission UI

The **Review Image Submission** process uses the **Review Image Submission HS** human service to provide the interface that captures the review of a new image document. It allows the user to see the image, add a comment and approve/reject it.

The service needs to be completed by adding the ability to retrieve the document being reviewed. Its coach node also needs to be updated to add the Document Viewer that will actually show the image being reviewed.

- __1. Click the **User Interface** category and double click **Review Image Submission HS** human service to open the editor.



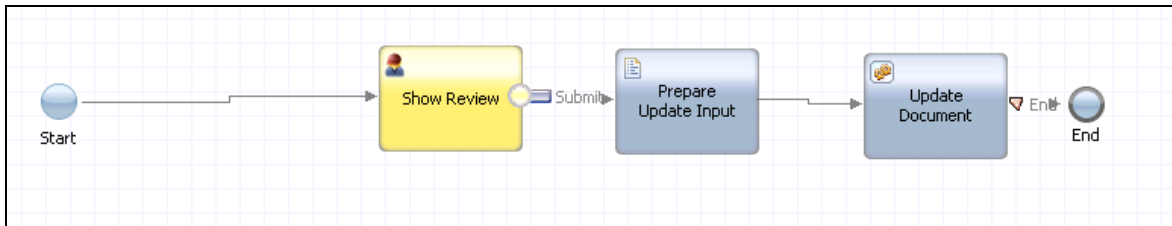
__2. **Examine** the service.

This service takes a single input, document ID, and does not return an output.

The service currently has three nodes. The **Show Review** coach holds the UI used to review the image submission. The **Prepare Update Input** script node uses Javascript to build the **IBM BPM document properties** with the values captured by the coach. Finally, the **Update Document** node invokes a service that will update the image document in the **IBM BPM document store** with the given properties.

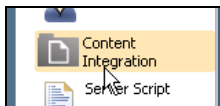
Take time to examine the **Prepare Update Input** node implementation. It will update the **'status'** property and set its value to **'reviewed'**. It will also add a new property **'approved'** and will set its value according to what was picked in the coach (**true** or **false**).

The missing component is the node that will actually retrieve the document from the **IBM BPM document store** and provide it as input to the coach. You will add this next.

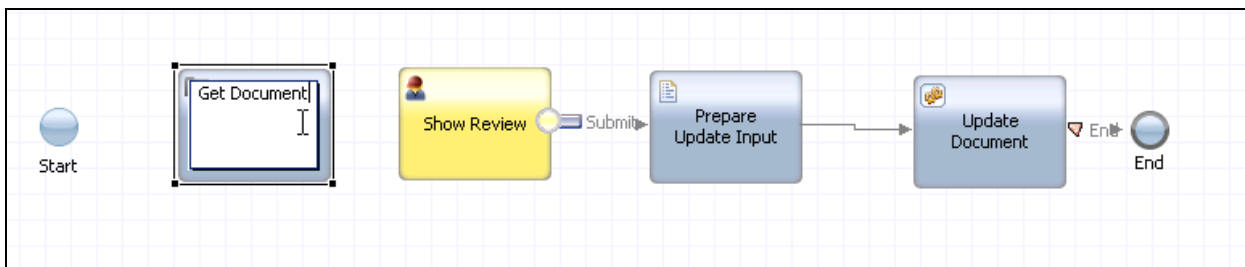


__3. Remove the link between the **Start** node and the **Show Review** coach node.

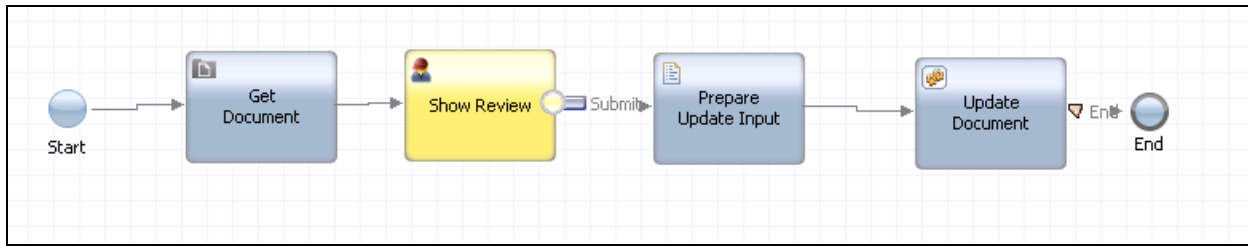
__4. In its place, drag and drop a **Content Integration** node from the palette.



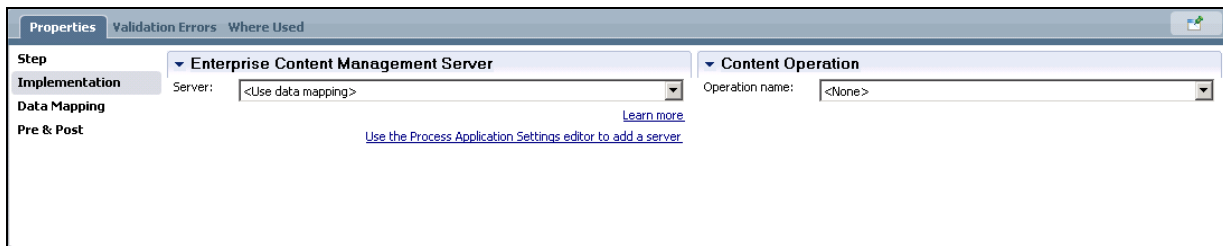
__5. Name the node **Get Document**.



- __6. Connect the **Start** node to the new **Get Document** node.
- __7. Connect the new **Get Document** node to the **Show Review** coach node.

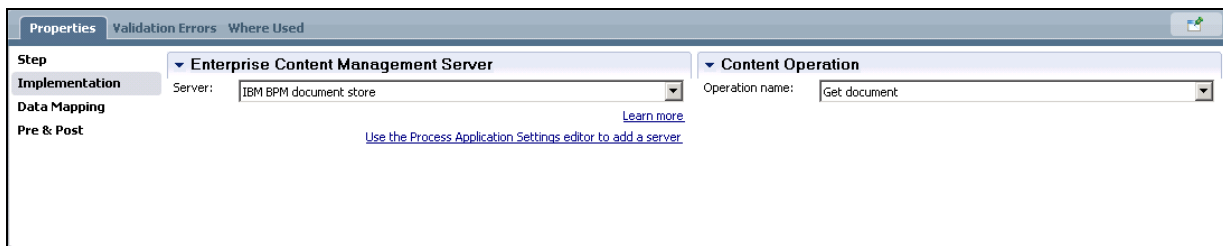


- __8. Select the **Get Document** node. Click the **Properties** tab and then **Implementation** tab.



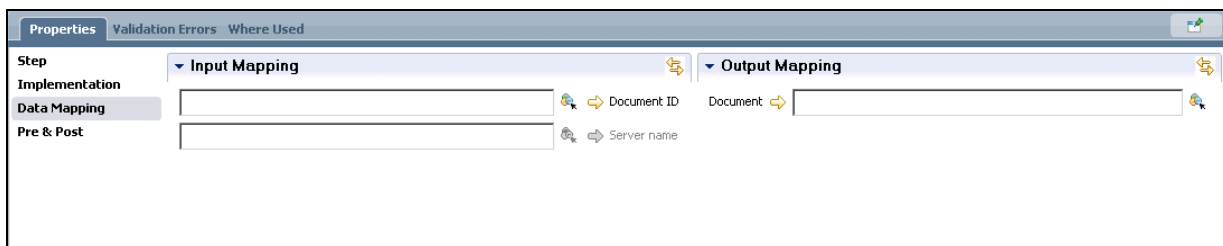
- __9. Select **IBM BPM document store** for the **Server** field.
- __10. Select **Get document** for **Operation name**.

These settings indicate that the node will invoke the **Get document** operation against the **IBM BPM document store**. Now that the operation has been selected, the data mapping can be specified.



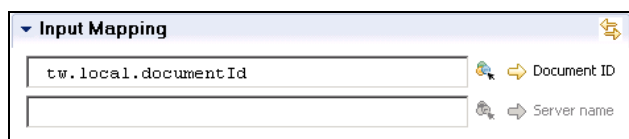
- __11. Click the **Data Mapping** tab.

Corresponding input and output variables have already been defined for you to use in the corresponding fields.



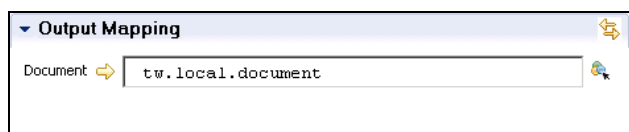
- __12. Set **Document ID** input mapping to **tw.local.documentId**.

This variable is the input to the service and carries the ID of the document we want to retrieve.



__13. Set **Document** output mapping to **tw.local.document**.

This is a private variable defined to hold the document itself. This variable will be used to populate the coach.

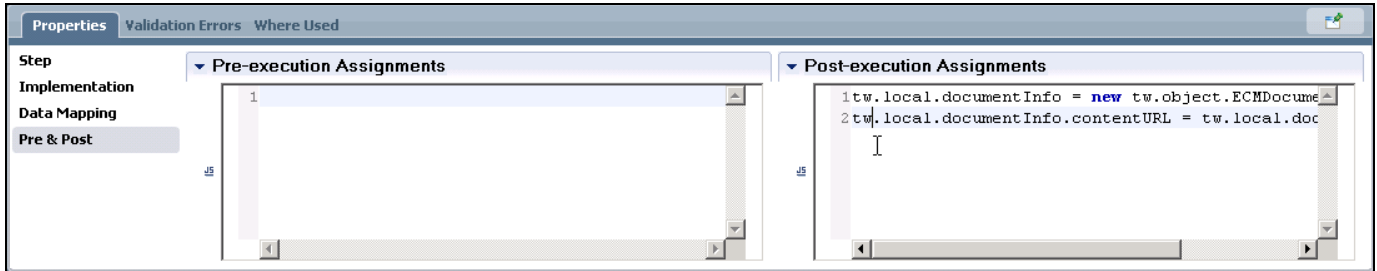


__14. Click the **Pre & Post** tab

__15. Add the following Javascript to the **Post** pane.

```
tw.local.documentInfo = new tw.object.ECMDocumentInfo();  
tw.local.documentInfo.contentURL = tw.local.document.contentURL;
```

Once the node executes, this code will be invoked to take the URL stored in the returned document object and place it in a new instance of **ECMDocumentInfo**. This particular type can be bound to a Document Viewer coach view. The result is that the view can resolve the URL and show the image content.



You can now move on to complete the coach UI.

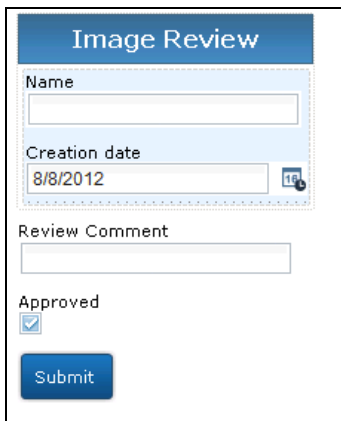
__16. Double click on the **Show Review** coach node to bring up the coach layout.

The coach itself is mostly complete.

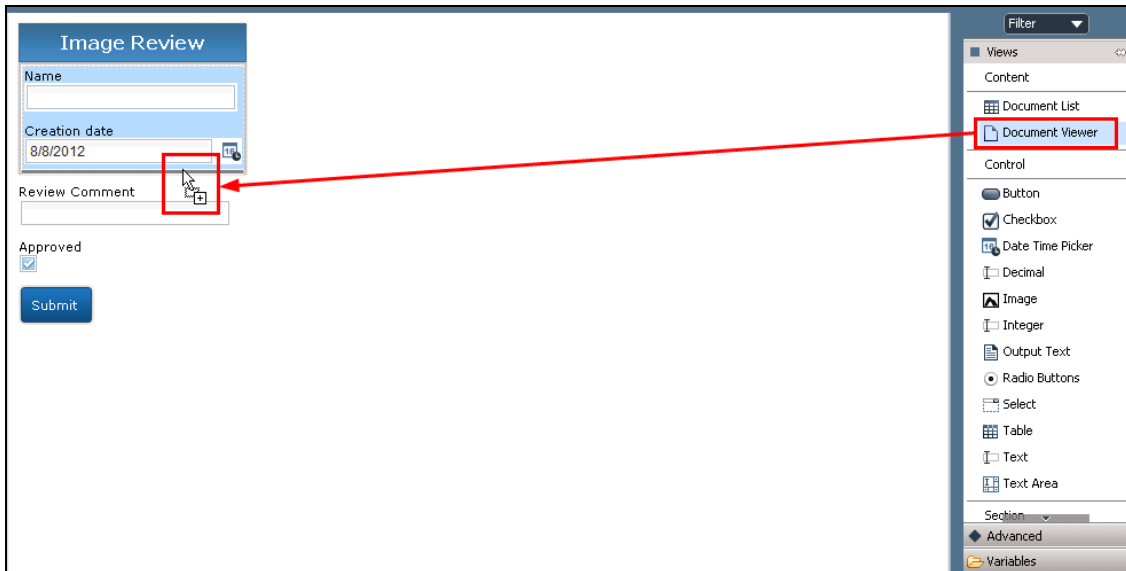
It contains a **name** and **creation date** fields. Both of these are bound to the properties of the document (**tw.local.document**).

It also contains a **comment** and **approved** widgets. These are bound to private variables that will later be used to construct the **IBM BPM document properties** object in the **Prepare Update Input** node.

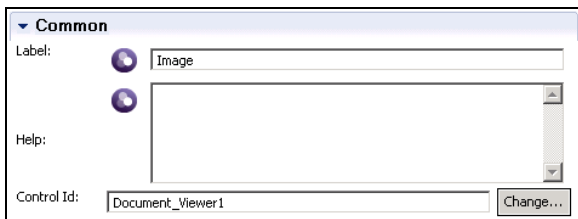
The only remaining work is to add a **Document Viewer** coach view to display the image content.



__17. Drag and drop a **Document Viewer** element into the **Image Review** section's content box.

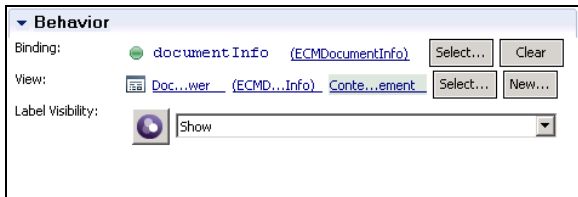


__18. Select the new **Document Viewer**. Click on the **Properties** tab and rename the view to **Image**.



__19. In the **Behaviour** section to the right, bind the view to the **tw.local.documentInfo** variable.

This variable was set in the **Post** pane of the **Get Document** node. The **contentURL** property contains the URL that the **Document Viewer** will use to retrieve and display the image.



__20. **Save** the service.

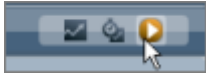
The application is complete. You can now test your application.

9.4 Part III – Test the Image Submission Process App

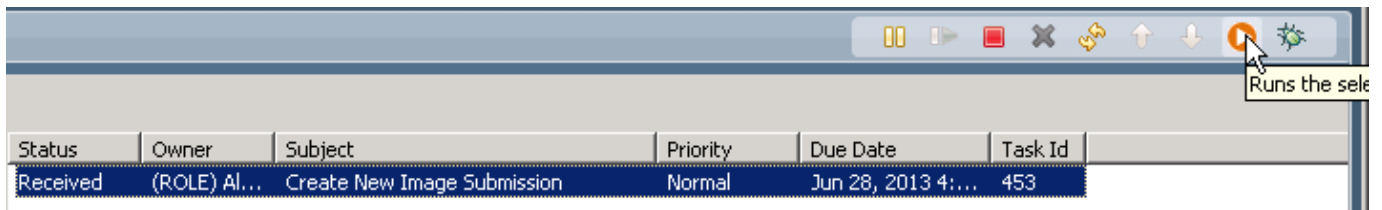
You will now run some test cases to ensure documents are created and modified as expected.

9.4.1 Test Image Review

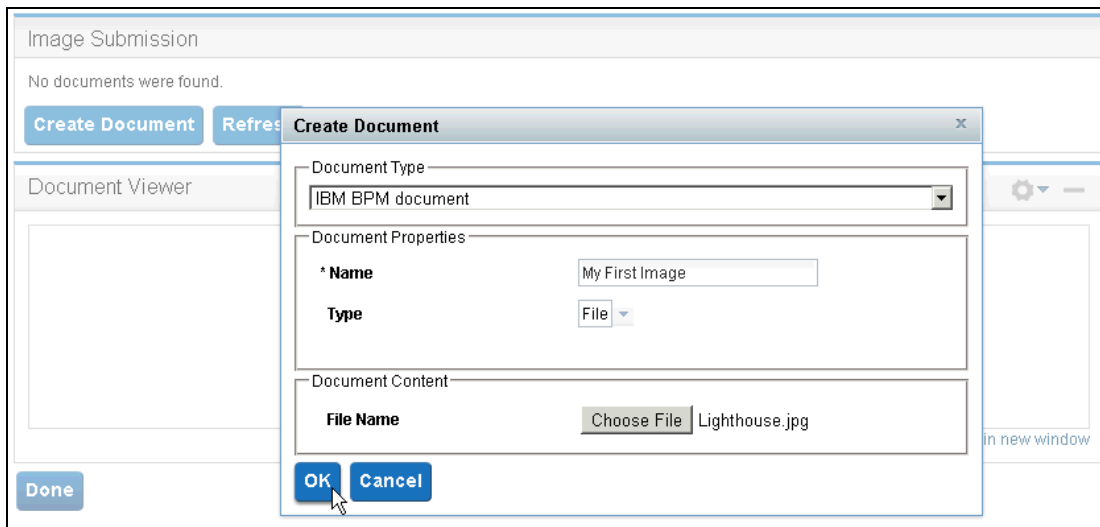
1. Open the **Submit New Image** process
2. Click the orange **Run Process** button



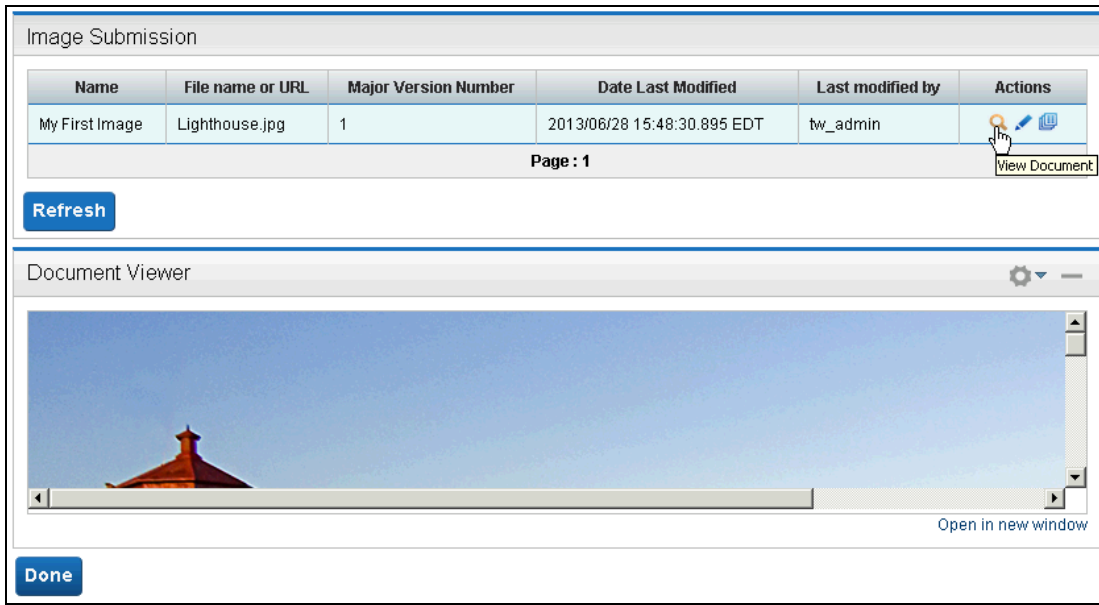
3. In the **Inspector**, select the **Create New Image Submission** step and click the orange **Run** button



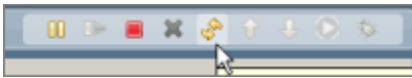
4. From *Pick Users From Role* window select user **admin** and click **OK**
5. Switch to web browser
6. Click **Create Document** button.
7. On the **Create Document dialog** enter the image name and select a file.
8. Click **OK**.



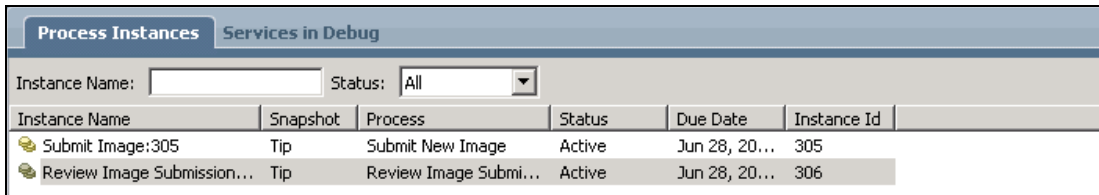
9. Click on the **view icon** to verify the image content.



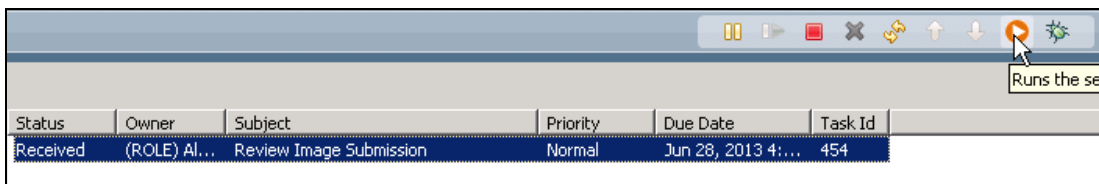
- __10. Click **Done** to finish the work.
- __11. Switch back to **Process Designer**
- __12. Keep clicking the **yellow refresh** button until you see a new instance of the **Review Image Submission** process in the window below.



Notice you will see the **Submit New Image** process still active. Selecting it will show that it is waiting on the intermediate event that will fire once the review is completed.

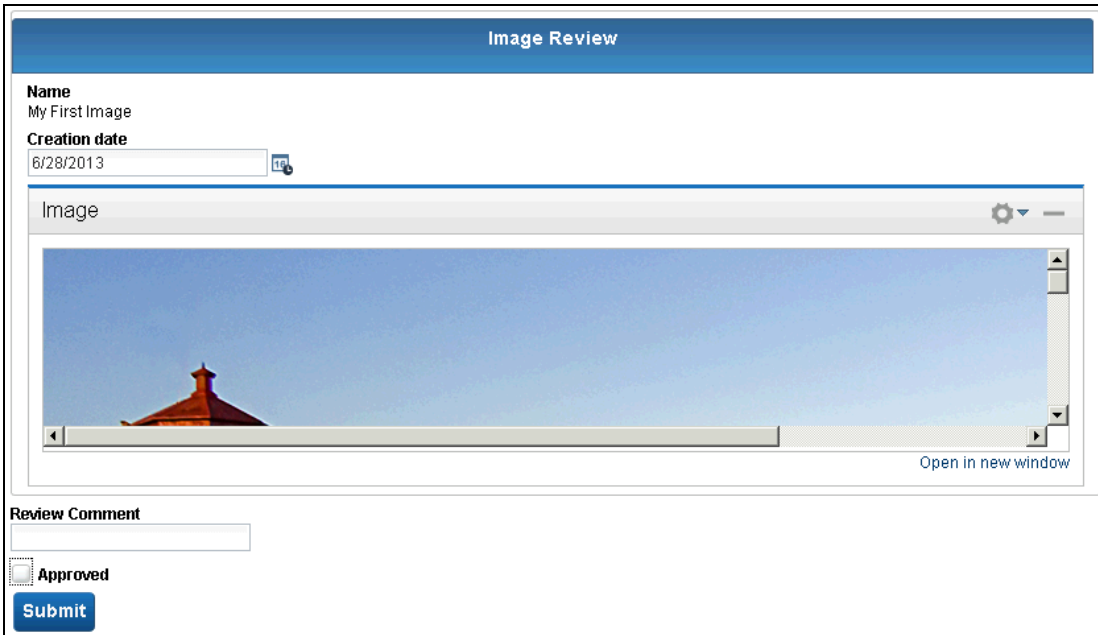


- __13. Select the **Review Image Submission** process.
- __14. On the right pane select the **Review Image Submission** step and click the orange **Run** button



- __15. From *Pick Users From Role* window select user **admin** and click **OK**

__16. Switch to web browser



This coach holds the information of the document retrieved using the document ID carried by the start event.

__17. Enter **'Nice shot!'** for **Review Comment** and check the **Approved** check box.

__18. Click **Submit**.

__19. Switch back to **Process Designer**

__20. Keep clicking the **yellow refresh** button until you see the **Review Image Submission** process has been **Completed** in the window below.



The **Review Image Submission** has run to completion. The **Submit New Image** process is still active but now has moved to the **View Review Response** node as the intermediate event was fired after the review was completed.

Process Instances		Services in Debug			
Instance Name	Snapshot	Process	Status	Due Date	Instance Id
Submit Image:308	Tip	Submit New Image	Active	Jun 29, 20...	308
Review Image Submission...	Tip	Review Image Submi...	Completed	Jun 29, 20...	309

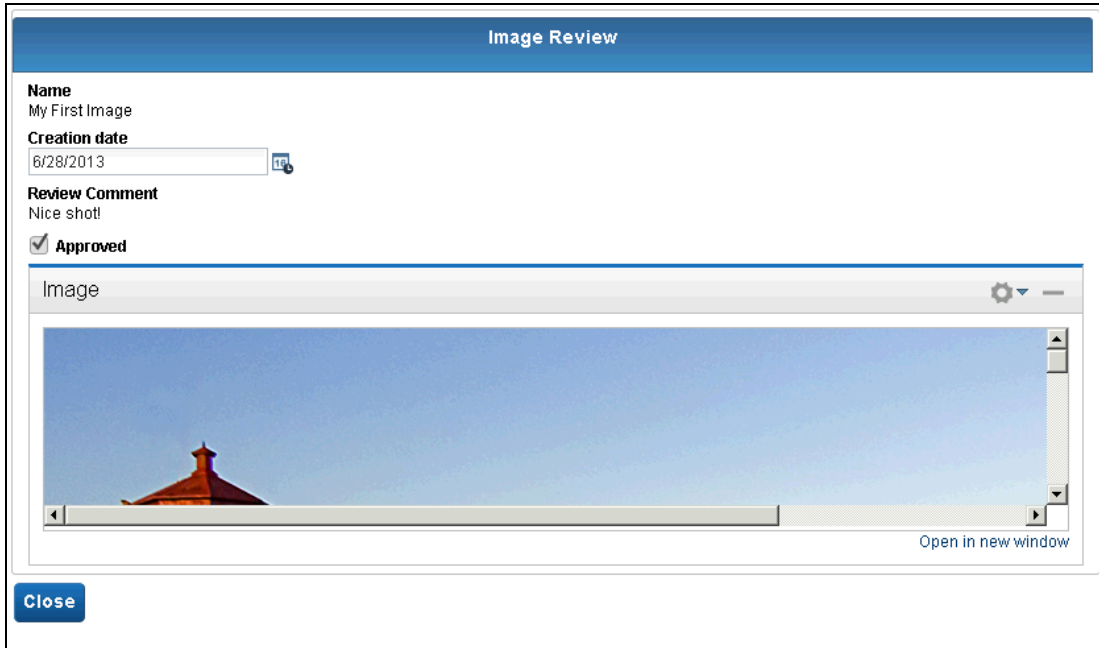
__21. Select the **Submit New Image** process.

__22. On the right pane select the **View Image Submission Review** step and click the orange **Run** button

Status	Owner	Subject	Priority	Due Date	Task Id
Closed	tw_admin	Create New Image Submission	Normal	Jun 28, 2013 5:...	457
Received	tw_admin	View Image Submission Review	Normal	Jun 28, 2013 5:...	459

__23. From *Pick Users From Role* window select user **admin** and click **OK**

__24. Switch to web browser



This coach shows the document content (image) and also the response fields: comment and approved. These are configured to be read only and are populated with the values set by the reviewer.

__25. Click **Close**.

9.4.2 Test Image Submission Dashboard

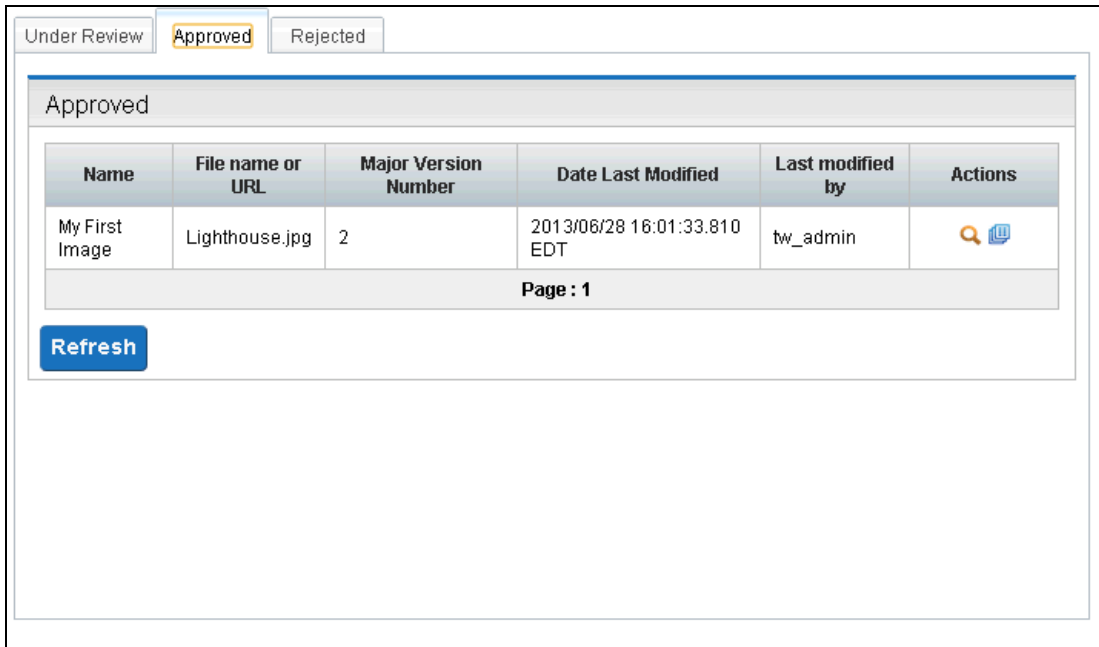
In this section you will test the dashboard that shows the image documents that exist in the system within each category: to be reviewed, approved and rejected.

You should first re-run the test in the previous section two more times. Once where a document is rejected and another up to the point where a document is submitted, but not reviewed.

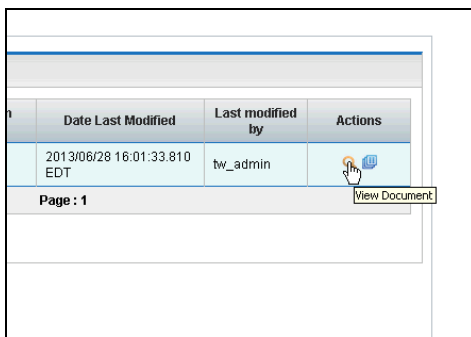
__1. Open the **Image Submission Dashboard**.

__2. Switch to web browser

The image below shows the approved image document created in the previous section. It also shows the other two tabs that would be populated according to the state of the other documents in the system.



__3. Click the **view icon** to see the image that has been approved.



__4. The image opens up in a new window.

9.5 Summary

In this lab you've explored all the ECM tools, available in Process Designer, to work with IBM BPM documents.

You created start and intermediate events to respond to event subscriptions registered against the IBM BPM document store. You also used the Content Integration node to execute one of several CMIS operations available. Finally, you used the Document List and Document Viewer coach views, shipped in the Content Management toolkit, to create and display a list of documents, and their content, in various coaches.

Finally, you used the Inspector view in Process Designer to test your solution.