IBM BUSINESS PROCESS MANAGER 7.5.1 – LAB EXERCISE

Developing business processes with Advanced Integration Services, the process developer's perspective

A top-down approach

What this exercise is about

IBM Business Process Manager Advanced (BPM) V7.5.1 is a single BPM solution that combines human-centric and integration-centric capabilities into a unified platform.

When developing a business process that uses services to integrate functionality from other systems there are several different approaches that can be taken. With one approach, the process developer defines the interface for the service to be called and the integration developer will then create the implementation based on the interface provided. This *top down* approach works well when the service does not already exist.

Another approach is for the integration developer to create the service interface and the service implementation and then make them available to the process developer in a Toolkit. This *bottom up* approach works well when there is already a mature SOA architecture with reusable services.

In this exercise you will learn how to use the *top down* approach for integrating Advanced Integration Services into your human centric business process.

Lab requirements

These products below are required for this exercise:

- IBM Business Process Manager V7.5.1 Server with IBM Process Center V7.5.1 stand-alone profile
- IBM Process Designer V7.5.1
- IBM Integration Designer V7.5.1
- DB2 V9.7.200.358 or higher

Throughout the lab, references to these products drop the IBM prefix and the version, calling the Process Center, Process Designer and Integration Designer.

What you should be able to do

At the end of this lab you should be able to:

- Import an existing business process definition into Process Center
- Implement a simple business process definition in Process Designer
- Define an Advanced Integration Service in Process Designer
- Configure Integration Designer to connect Process Center
- Implement the Advanced Integration Service in Integration Designer
- Publish and synchronize changes between Integration Designer and Process Center
- Test a business process definition that is associated to an advanced integration service in Process Designer

Introduction

This lab exercise has three parts to it. As you work through the lab you will be switching roles from *process developer* to *integration developer* and then back to *process developer*.

The business process used for this exercise models a work request order where the request can be handled using either the standard process or the expedited process. The expedited process requires human interaction while the standard process can be automated using existing services.
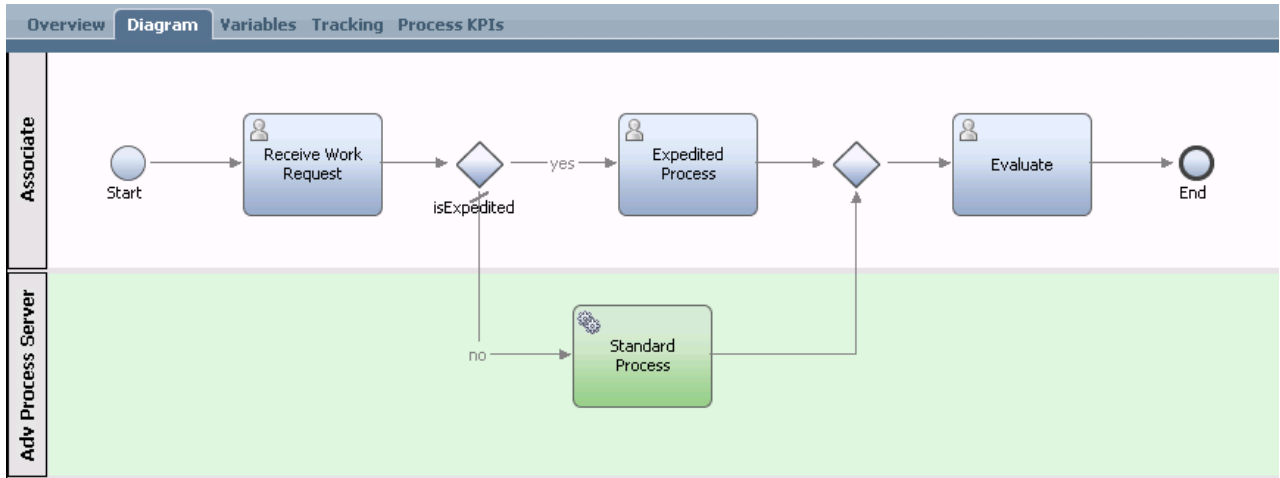
As the *process developer*, you will import a **Work Request** business process that was created with WebSphere Lombardi edition V7.1. Then you will define an *advance integration service* called **Standard Process AIS**, and associate it with the **Standard Process** system task.

In the second part you switch roles and become the *integration developer*. Using Integration Designer, you will first import the business process components into your workspace. Then you will implement the **Standard Process AIS** service as a Java component. When the implementation is complete, you will publish it to the *Process Center* repository, where it can then be used by the *process developer.*

Returning to the *Process Designer;* as a *process developer* you will step through the Work Request business process in the Process Designer Inspector, and test the newly created **Standard Process AIS** *implementation* in the context of the business process flow.

Part 1: Process Designer - Defining the Advanced Integration Service

The first part of this lab will focus on importing as existing business process into **Process Designer** and defining an **Advanced Integration Service** called **Standard Process AIS** which will eventually be implemented using **Integration Designer**.

____ 1.   Start **Process Center**
   __ a. Select **Start → All Programs → IBM → BPM Advanced 7.5 → Profiles → ProcCtr01 → Start the Server**
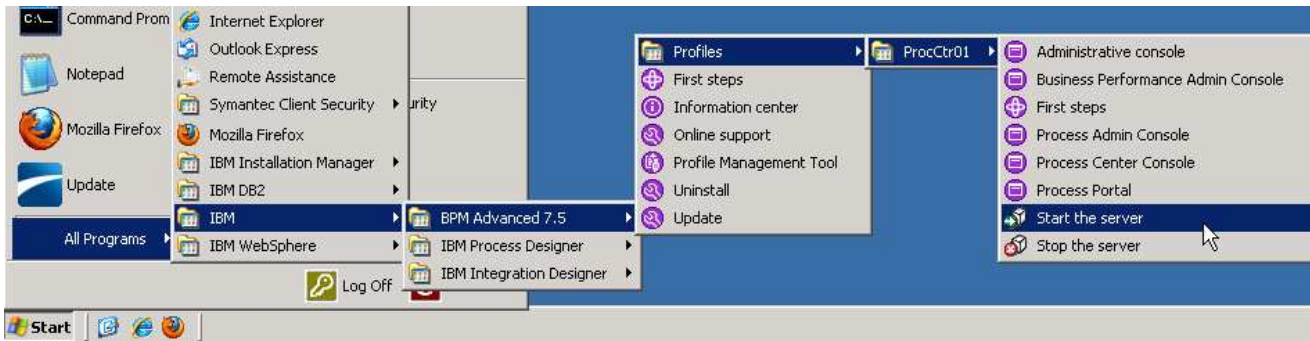
    __ b. Wait for the server to start, which can take a few minutes. While it is starting, you should see this window:
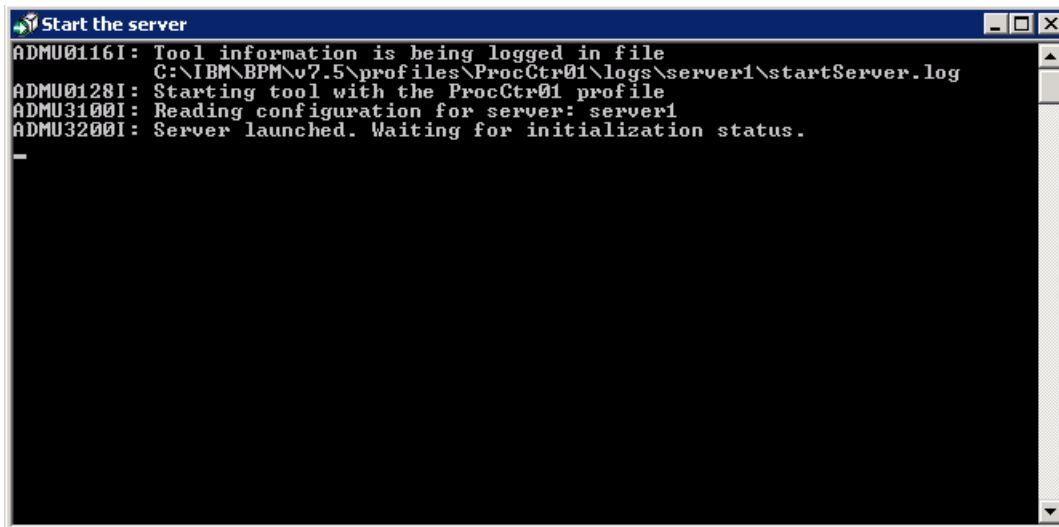
    __ c. When the server startup completes, you will see some additional messages in the window and then it will then close automatically. When the window disappears, you are ready to continue

_____ 2.   Start **Process Designer**

__ a. Start **Process Designer** by clicking on the Process Designer icon from the Windows *Start* menu.
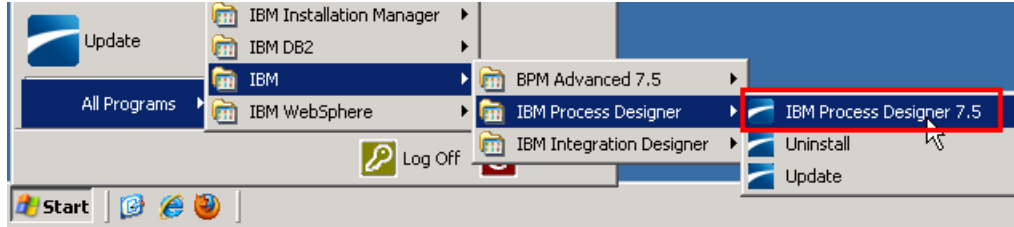
NOTE: The user that will be publishing the Process Application or Toolkit that contains Advanced Integration Services must have either *Deployer* and or *Configurator* administrative privileges in the WebSphere Process Server user registry. This is managed through the WebSphere Process Server administrative console, also known as the *Integrated Solutions Console*.

__ b. Enter the credentials

User Name: <**your user id**>   (default admin user is *admin*)
Password: <**your password>** (default admin password is *admin*)

And then click **Login**.

__ c. If this is the first time you launch the Process Designer, you may see a **SSL SIGNER EXCHANGE PROMPT** dialog pops up. This message occurs when the WebSphere Application Server SSL configuration has changed and the client does not trust the new or changed SSL certificate.
Just click **y** to add singer to the trust store.

__ d. If you are presented with Getting Started screen, you can close it.

____ 3. Import existing process application into **Process Center** and open it in Process Designer
__ a. First, you need to ensure you are in the Process Center view. In the Process Center view, you will see the tabs across the top.

If you do not see this then you are in the Process Designer view.
In the upper right corner of the window, **if you see a Process Center icon, click it** to switch to the Process Center view.

____ 4. Import existing process application into the Process Center and open it in Process Designer
__ a. On the right side task menu select **Import Process App** and click **Browse**
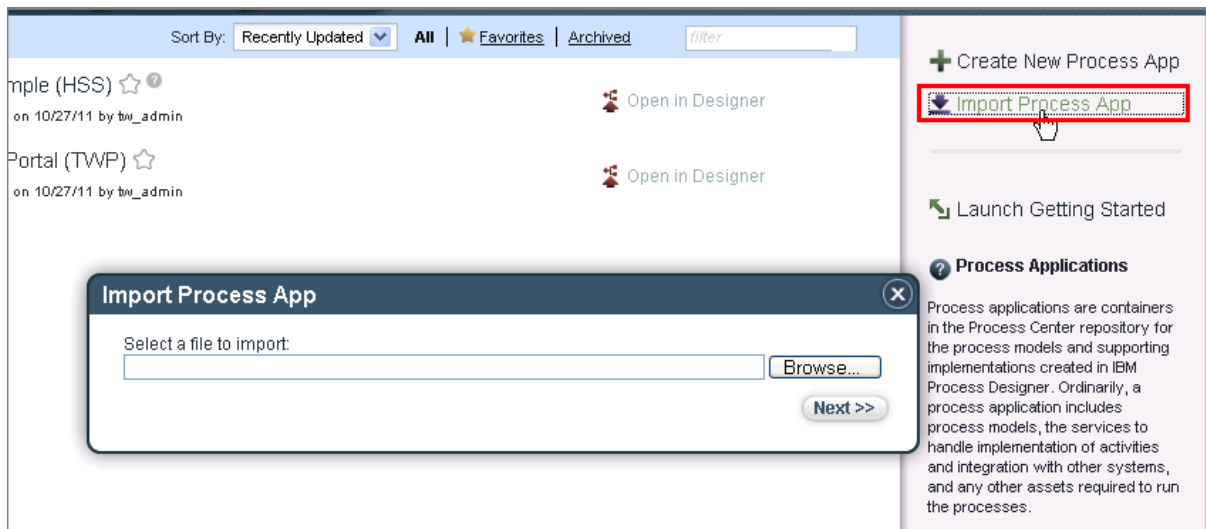
__ b. Navigate to the downloaded lab artifacts directory and select **Request Application–
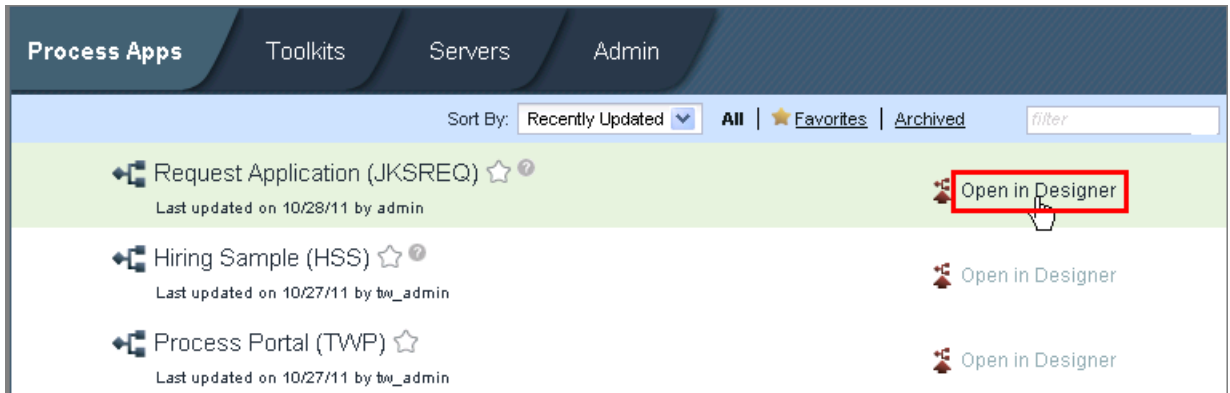Iteration 02.71.twx**. Click **Open**

__ c. Once you return to the **Import Process App**, verify the right address is listed in the test
box and click **Next**

Import Process App

Select a file to import:

C:\Labfiles75\Request Application - Iteration 02.71.twx   Browse...
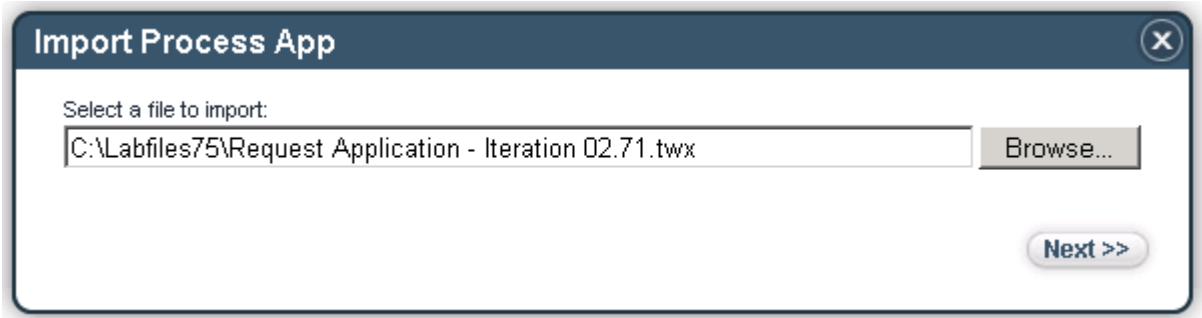
Next >>

__ d. The next dialog confirms your selection to import **Request Application – Iteration 02**.
Click **Import**.

Import Process App

▼ **Will Be Imported**
    Request Application - Iteration 02
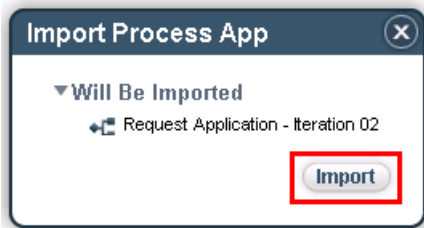
    Import

__ e. Verify you can see **Request Application (JKSREQ)** in the list of available process
applications. Select **Open in Designer**.

Process Apps   Toolkits   Servers   Admin

Sort By: Recently Updated   **All** | ★ Favorites | Archived   *filter*

Request Application (JKSREQ) ☆ ?   Open in Designer
Last updated on 10/28/11 by admin

Hiring Sample (HSS) ☆ ?   Open in Designer
Last updated on 10/27/11 by tw_admin

Process Portal (TWP) ☆   Open in Designer
Last updated on 10/27/11 by tw_admin

NOTE: The previous steps walked you through the steps for importing an existing process
application into the **Process Designer** environment.

____ 5.   Upgrade the dependency version.

Since the original process application was created in WebSphere Lombardi Edition V7.1, the **System Data** version under the **TOOLKITS** category shows 7.1.0 with a warning icon.

You need to bring the version up in order to continue this lab. To update the Toolkit data to the latest version, right click **System Data (7.1.0)** and choose **Upgrade dependency to 7.5.1**.
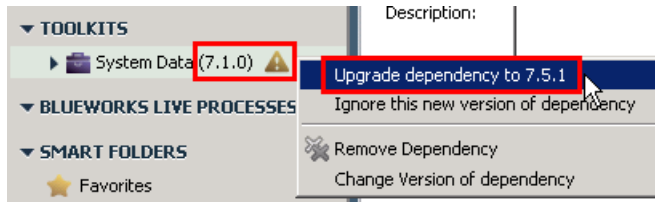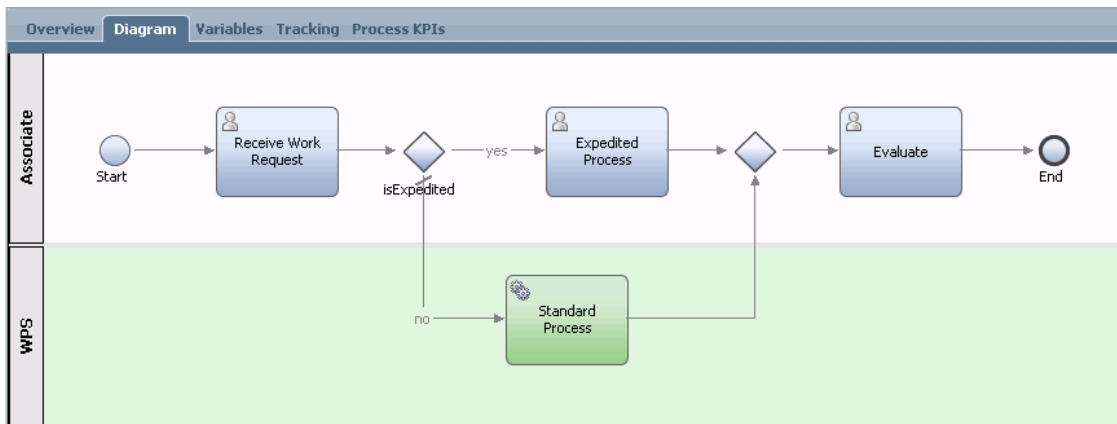

____ 6.   Open **Work Request** business process definition (BPD)
   __ **a.** In the navigation tree of the **Designer** view, click the **Processes** category. In the menu that shows up, double click **Business Process Definitions → Work Request** to open it in the main canvas.
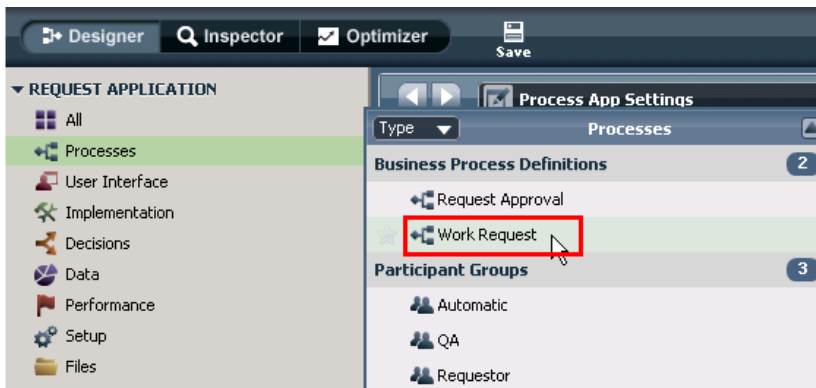
After you double click Work Request, the process flow opens on the main canvas as shown below

_____ 7. Each business process requires variables to capture the business data that is passed from step to step in the process. The **Work Request BPD** has a complex variable called **theWorkRequest.** It is a complex variable of type **WorkRequest**. This complex type holds information about the WorkRequest, such as the number, description, status and priority. Open the variable **theWorkRequest** to review this.

    __ a. Click the **Variables** tab above the main canvas. This will open a dialog for creating and managing variables for the current process.

    __ b. Expand **theWorkRequest** variable to see its constituent parameters and their specific types. This variable has a private scope to this business process.
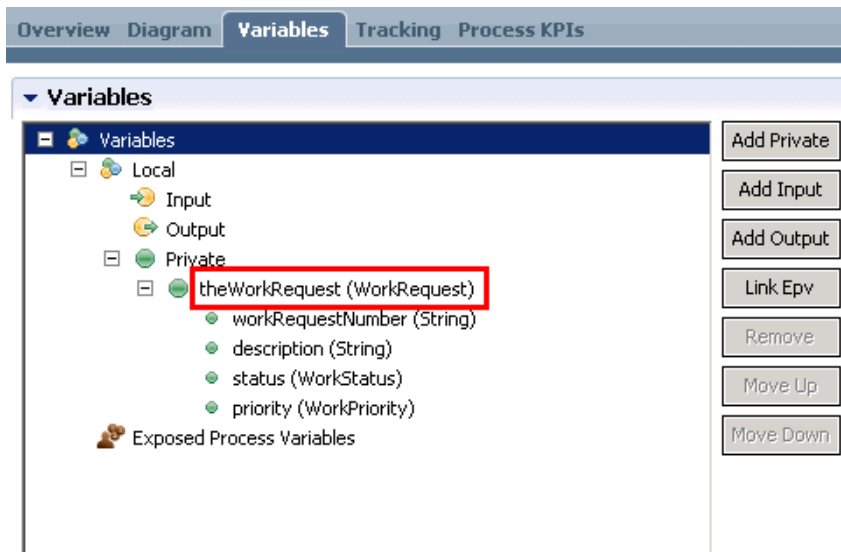
NOTE: While modeling processes in **Process Designer,** you should only model variables that hold business data needed by the process. This keeps the process more manageable. For example, in this lab you do not create a very extensive data schema and only define variables that are needed by the **Work Request** business process.

_____ 8. Return to the main canvas by clicking the **Diagram** tab next to the **Variables** tab.

_____ 9. The activity steps for the business process **Work Request** have already been defined. Take a look at the implementation of each activity, starting with the first activity called **Receive Work Request**
    __ a. Select the **Receive Work Request** activity box on the main canvas and click the **Properties** tab below
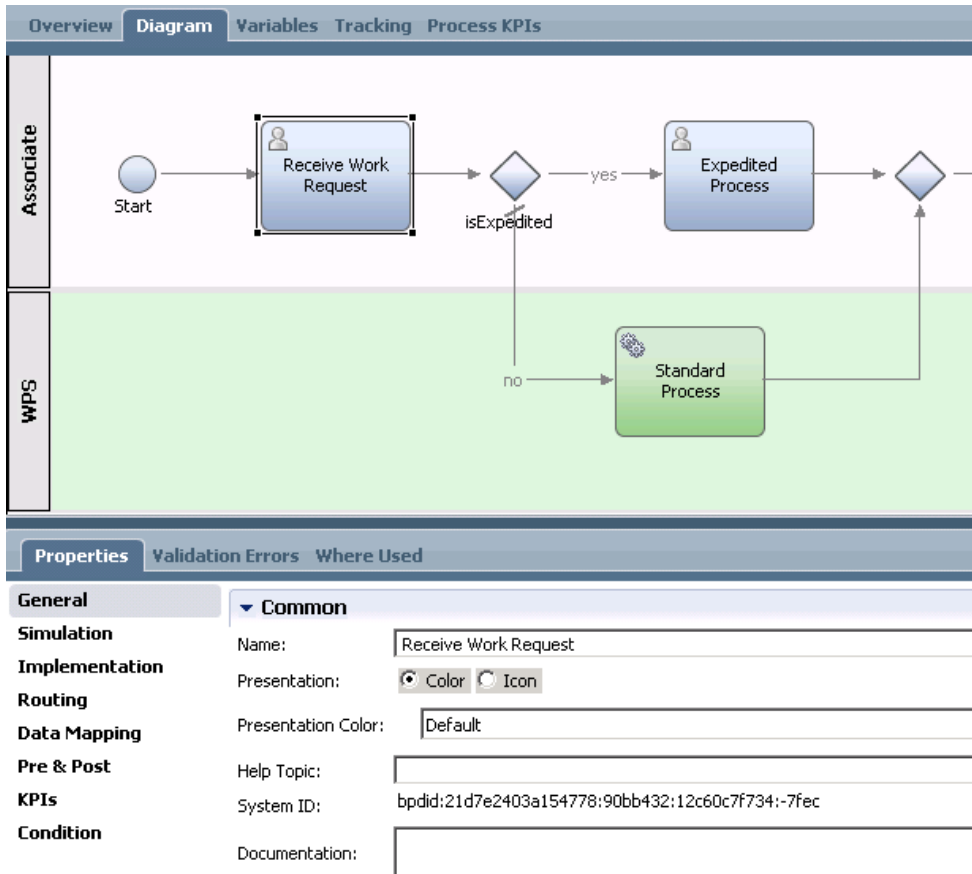
__ b. Click the **Implementation** tab to reveal the associated implementation for the activity. **Receive Work Request** activity has been implemented as a human service called **Submit Work Request**.
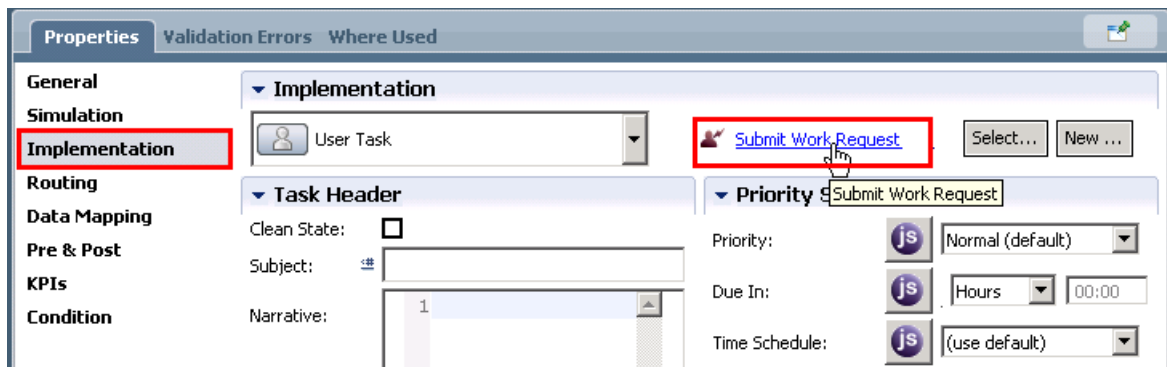
The man icon next to **Submit Work Request** signifies that this is a human service.

NOTE: Submit Work Request has already been implemented for this lab as a human service with a custom user interface to take the input of the work request from the user. Click it to review this implementation.

_____ **10.** Review the implementation of the **Submit Work Request** human service.
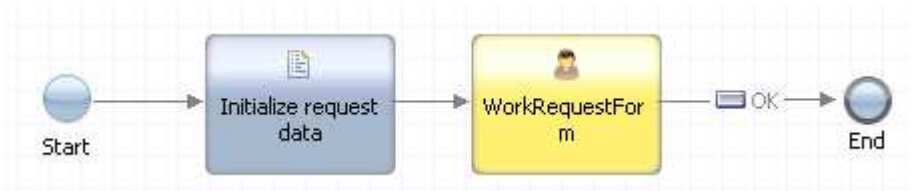
__ **a.** Click **Submit Work Request** link to open its implementation in the editor.

__ **b.** Select the **Initialize request data** activity and then the **Implementation** tab in the properties below.

__ **c.** All complex business objects must be instantiated before they can be used. This includes the nested complex business objects also. In this **Server Script** the complex business objects are instantiated and then default values are assigned to all the variables so that when the coach is invoked, the fields are pre-populated.

```
▼ Script
  1 tw.local.theWorkRequest = new tw.object.WorkRequest();
  2
  3 tw.local.theWorkRequest.workRequestNumber = "WRxxx";
  4 tw.local.theWorkRequest.description = "Describe the nature of the request here";
  5 tw.local.theWorkRequest.status = "new";
  6 tw.local.theWorkRequest.priority = "standard";
```

__ **d.** To return to the main diagram you can close this diagram or keep it open and navigate back to using the arrows or the drop down box.
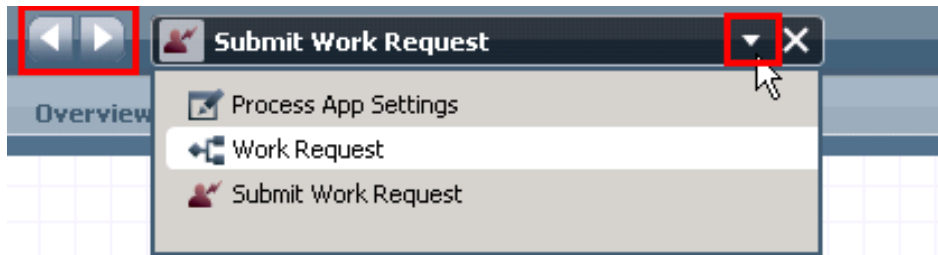
_____ **11.** Customize the **Swim Lanes**.

__ **a.** By default there are always two swim lanes created, **Participant** and **System**. The swim lanes represent the user roles that are responsible for the activities contained in them. You can create additional lanes as needed, change the names and the background color.

__ **b.** The **System** lane is a special swim lane where activities that are implemented as automatic services are placed. It is the responsibility of 'the system' to run and manage them. Select the lane and take a look at the **Behavior** section in the properties.

__ **c.** The original developer changed the name of the default participant lane to **Associate**, and changed the color of the system lane to green and named it to **WPS**. The name for the Associate lane is still appropriate but the name of the system lane needs to be changed to **Adv Process Server**, to indicate that the activities therein will be run and managed by the advanced process server component.

__ **d.** To change the name of the WPS system lane to **Adv Process Server**, select the lane, the background or the name, and then in the **Common** section of the properties below, change the name. Alternatively using direct editing, click the name on swim lane header and change it right at the in-place name editor.
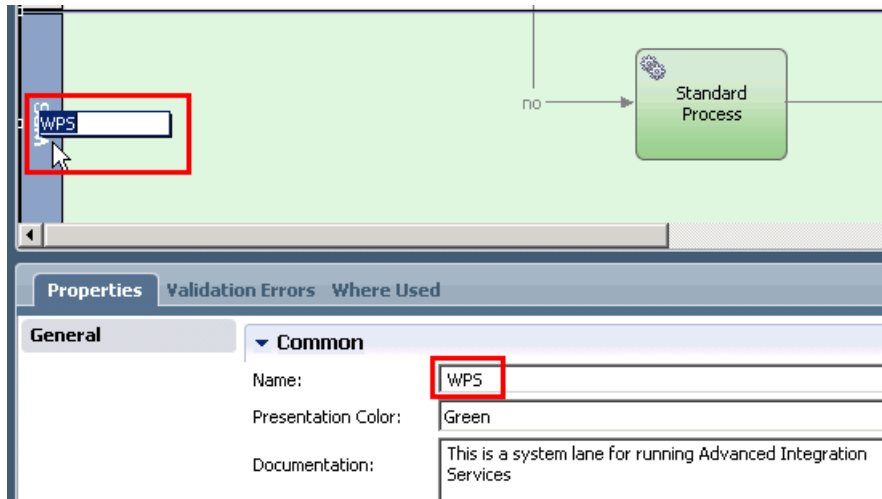
____ **12.** Next define an Advanced Integration Service to be associated the **Standard Process** activity.

__ **a.** From the navigation tree of the **Designer** view, click the "+" sign on the right side of **Implementation** category. In the menu that shows up, click **Advanced Integration Service**.
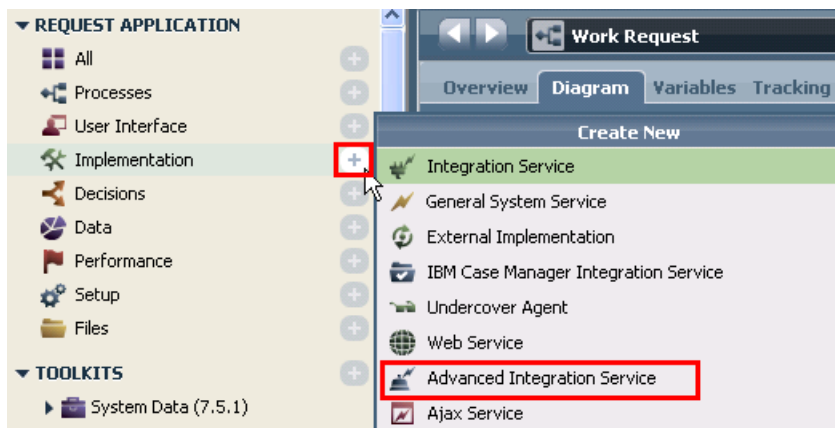
__ **b.** In the **New Advanced Integration Service** dialog, specify the name 'Standard Process AIS', and click **Finish**.
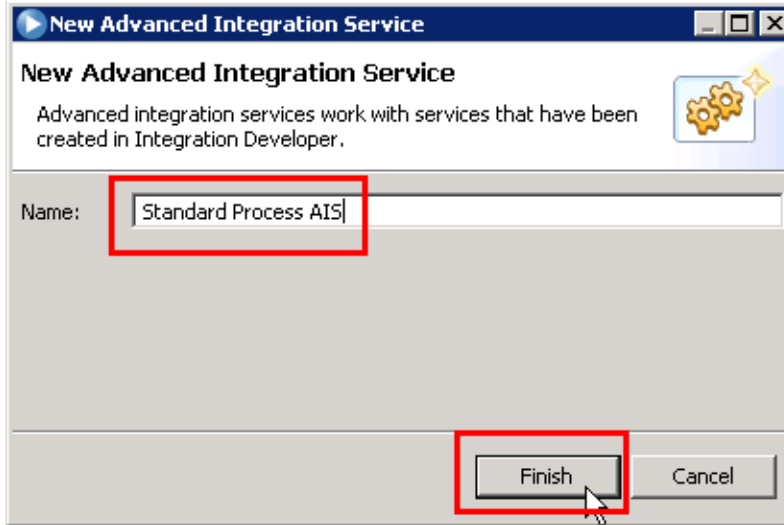
__ **c.** On **Standard Process AIS** service **Overview** page, notice the Advance Integration Service properties are blank at this time. These will be filled once this service has been implemented in **Integration Designer** and published.
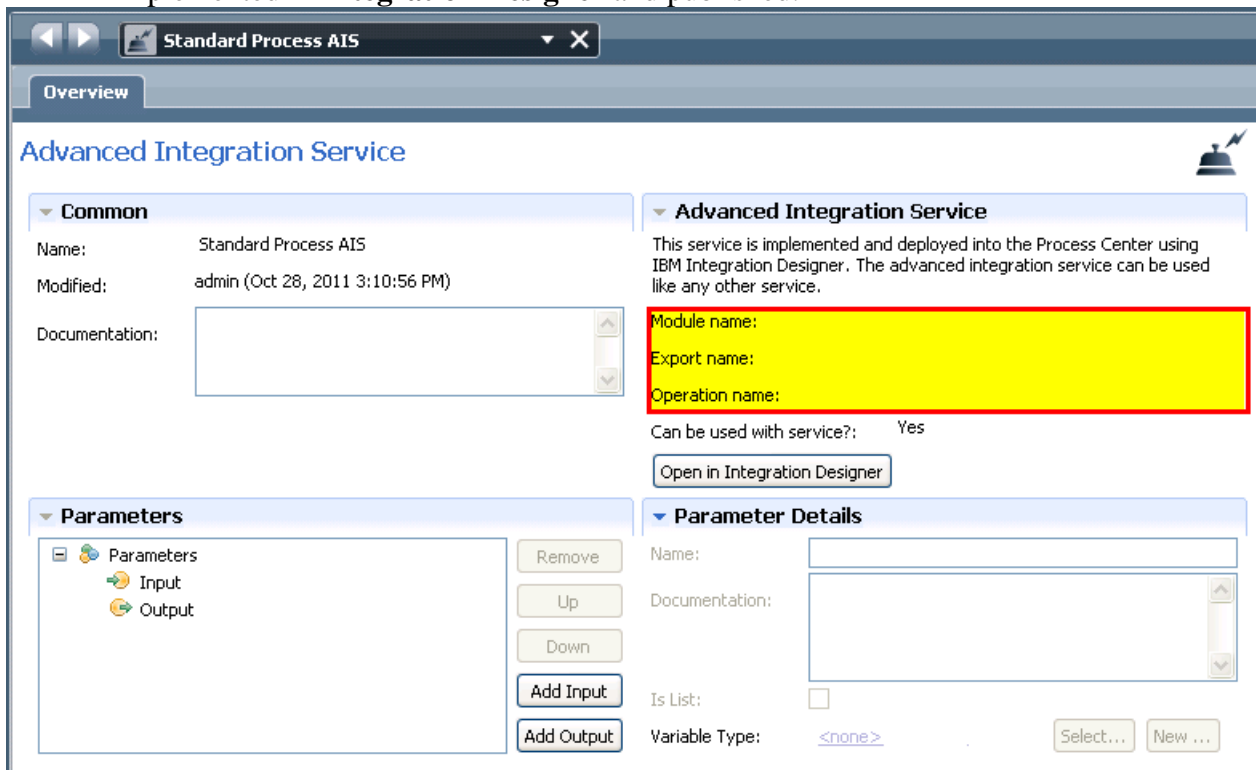
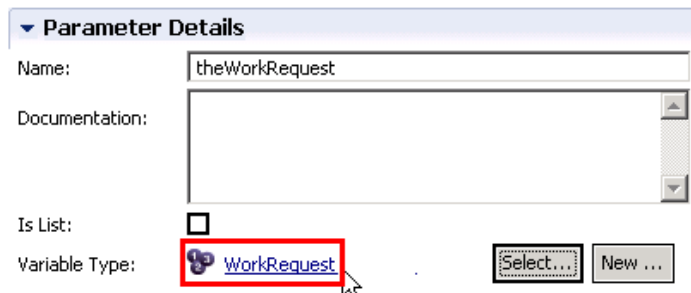__ **d.** In the **Parameter**, click **Add Input**.

__ **e.** In the **Parameter Details** section, change the name from 'Untitled1' to 'theWorkRequest'. Then click **Select…**.
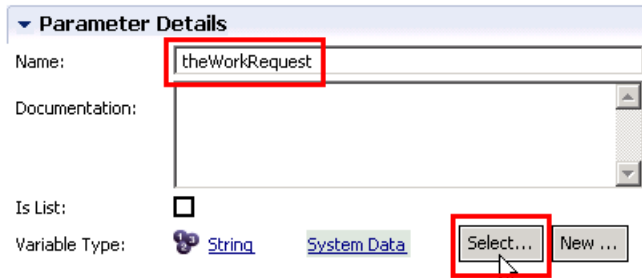
__ **f.** Select **WorkRequest** from the variable type list.
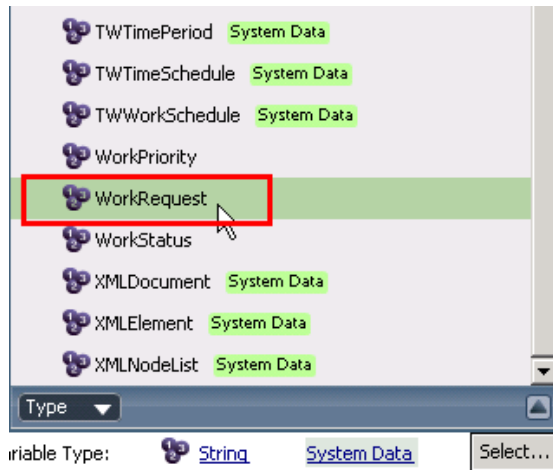
__ **g.** Notice that the **Variable Type** is now changed from **String (System Data)** to **WorkRequest**.

__ **h.** Click **Add Output**, and again type in **theWorkRequest** for the name, click **Select…** and click **WorkRequest** again.

__ **i.** Now the **Parameter** section should have both input and output with **theWorkRequest** variables.
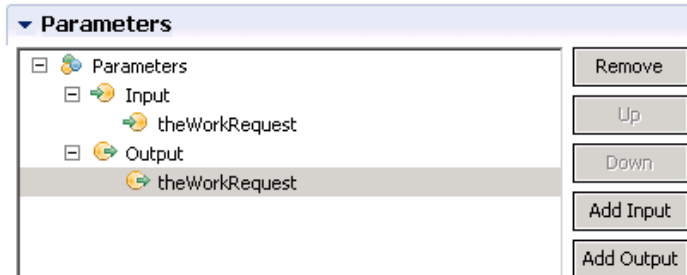
__ **j.** Save the **Standard Process AIS** service implementation.

__ k. Click the **X** next to **Standard Process AIS** to close this overview dialog and return to Work Request process flow.
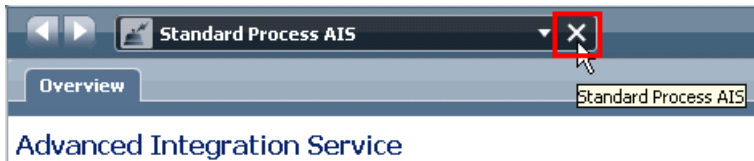
____ **13.** Now you will associate the **Standard Process AIS** service with the **Standard Process** activity.

__ **a.** Select the **Standard Process** activity box and open the **Properties → Implementation** tab. The activity by default is implemented as a **Default Human Service**. Click **Select…**.
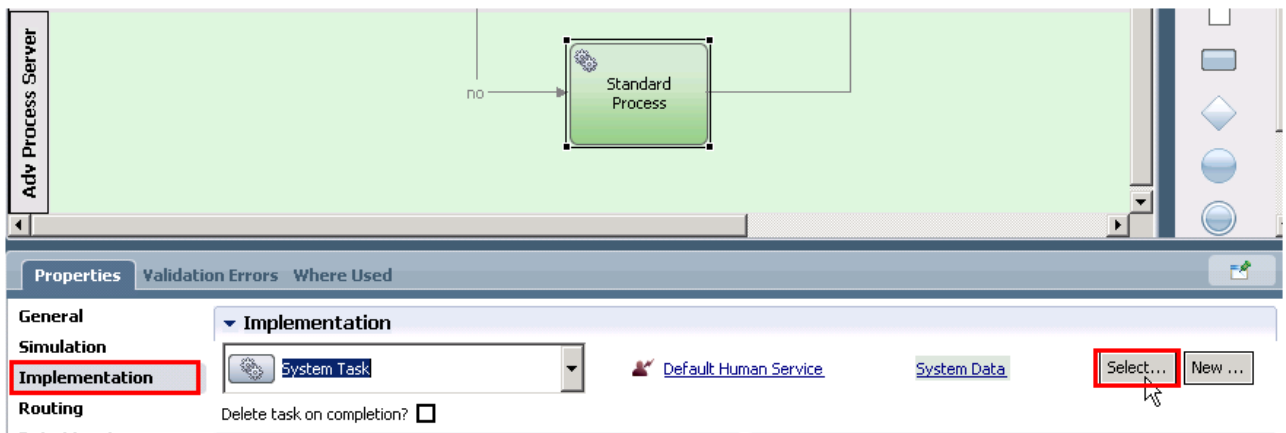
__ b. From the service list, click **Standard Process AIS** service at the top.
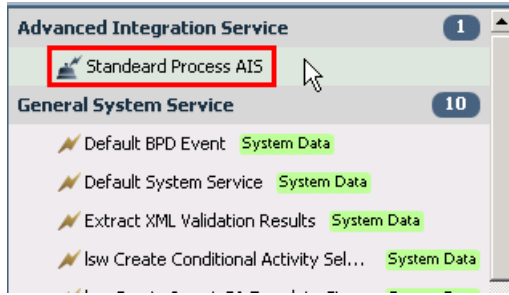

__ c. Navigate to **Properties → Data Mapping**. Click the double arrow icon for both **Input Mapping** and **Output Mapping** to make the assignments.
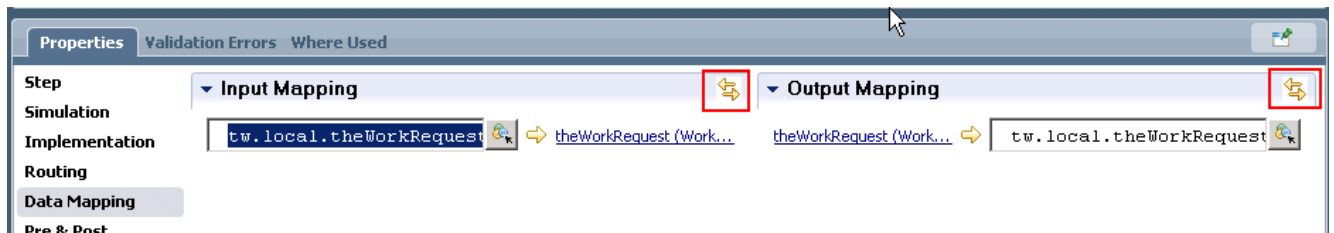

NOTES:

1. The double arrow icon auto-maps input/output properties with variables from service

2. The mapping can be assigned manually also through the select variable button

---

____ 14. If there is an ***** next to **Work Request**, it signifies that there are pending changes to be saved. Click **Save** to save all open editors.
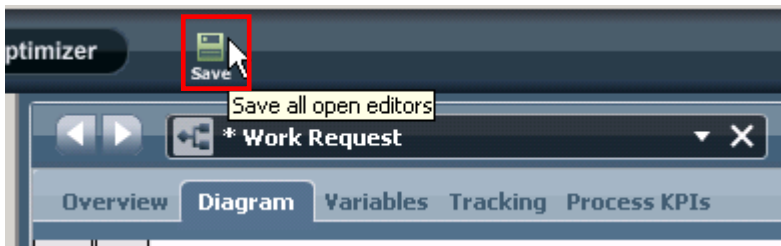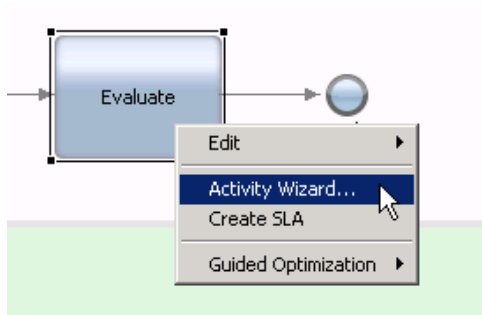

____ **15.** Next implement the **Evaluate** activity in the **Associate** swim lane.

__ **a.** Select the **Evaluate** activity box and open the **Properties → Implementation** tab**.** The activity by default is implemented as a **Default Human Service**.
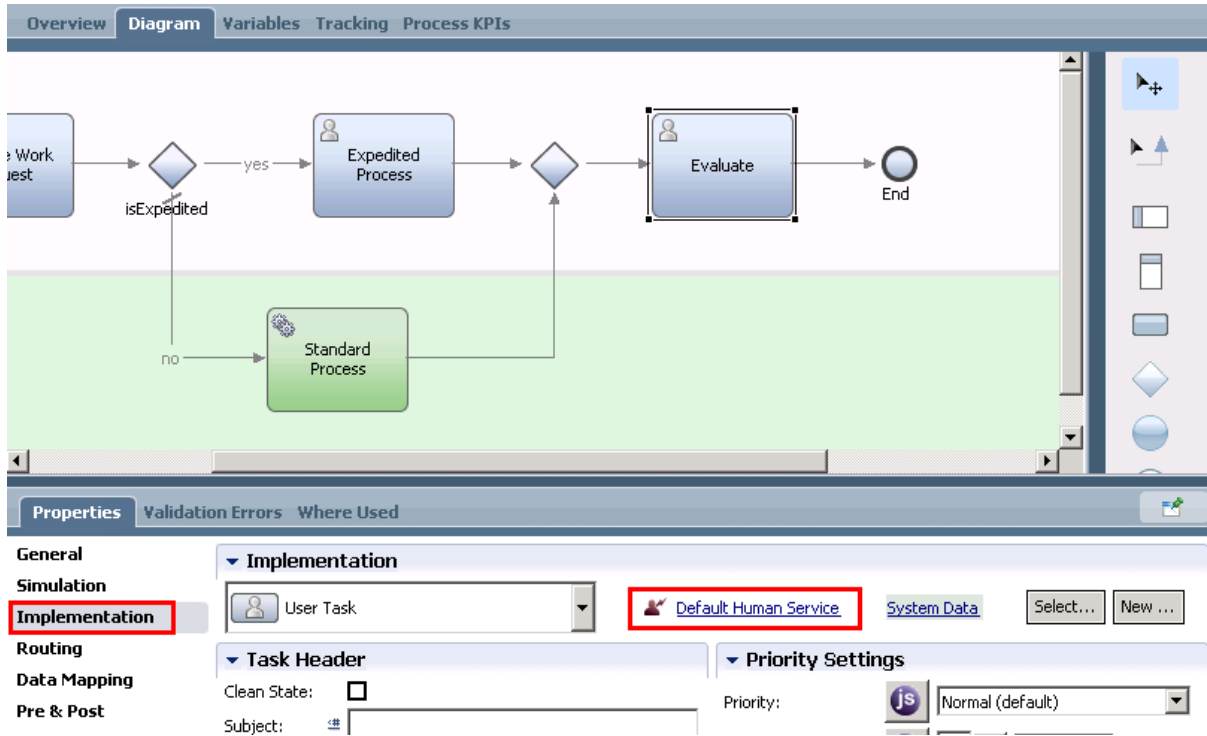
__ b. Right click **Evaluate** activity box and select **Activity Wizard**

__ c. In the **Activity Wizard** leave the default selections - **Activity Type** as **User Task** and new service **Name** as **Evaluate.**
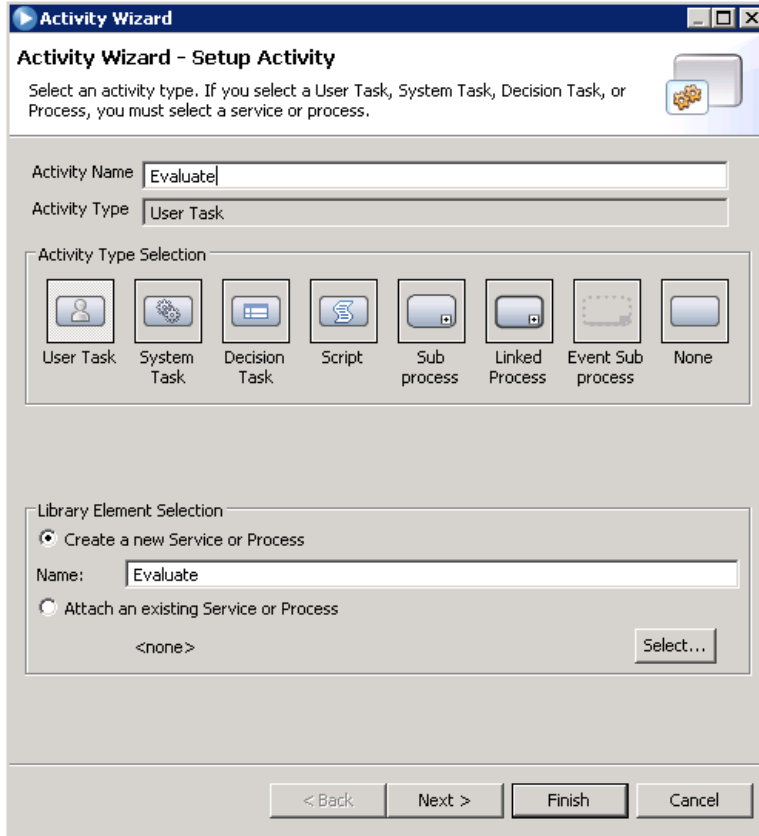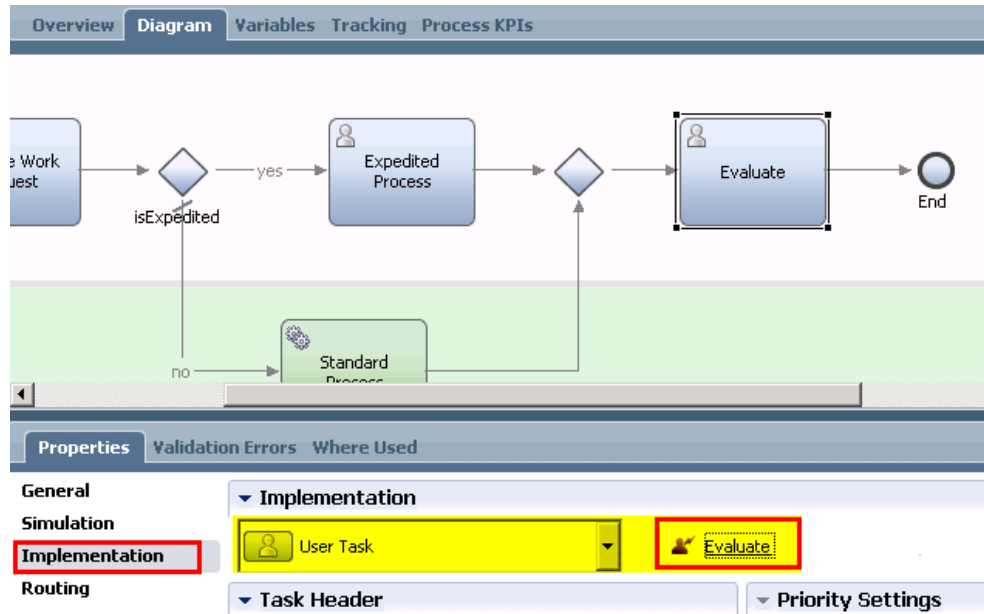
__ d. Click **Next**

__ e. In the **Activity Wizard – Parameters** dialog, notice that **theWorkRequest** variable will
automatically be defined as the input and output process variables for **Evaluate** service.
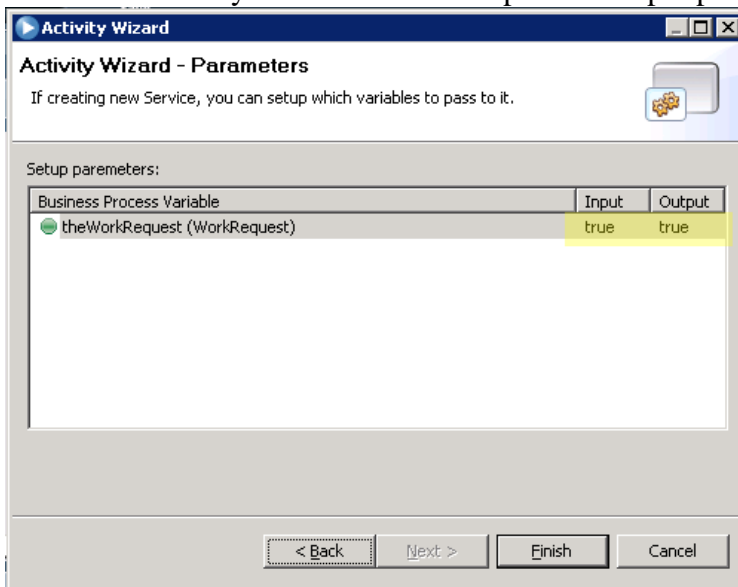
__ f. Click **Finish**

__ g. In the **Properties → Implementation** dialog for **Evaluate** activity, the implementation service is changed to **Evaluate** service.


__ h. In the **Properties → Implementation** click the **Evaluate** service link. This will open the configuration dialog to define and configure **Evaluate** service.

__ i. Click the **Coaches** tab to review the auto generated user interface base on the settings in the activity wizard.


NOTE: The Coach Editor may also be opened by double clicking on the Coach activity on the diagram canvas.
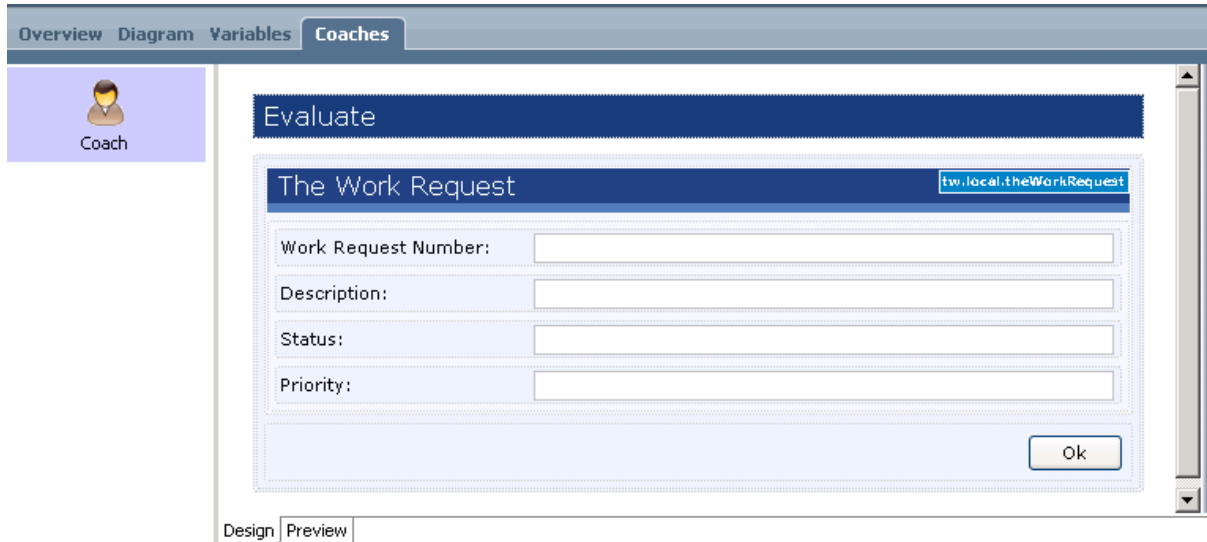
Notice that this coach page has been setup based on **theWorkRequest** variable set in the activity wizard.

NOTE: In this example the default coach is all that is needed. The palette on the right provides many tools to help customize and configure the interface if needed.

____ 16. Close **Evaluate** service editor by clicking on **X** next it name and return to **Work Request** process flow diagram.
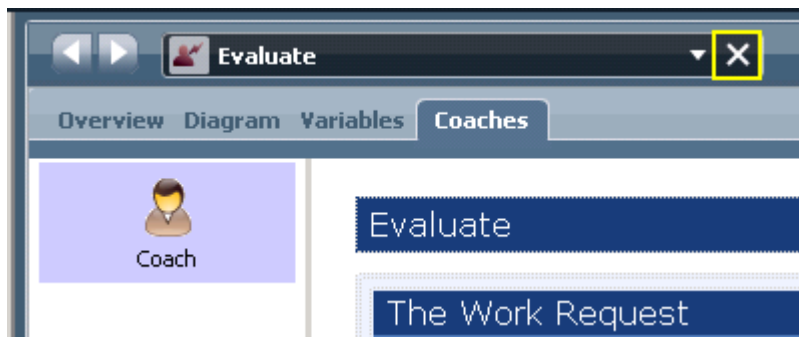
____ 17. The Work Request process flow is now completely implemented. Save all your changes by entering **Ctrl + S.**

At this point you are ready to implement the Advance Integration Service in the IBM Integration Designer in Part 2

## Part 2: Integration Designer - Implementing the Advanced Integration Service

In this section, your role will be changing from the Process Developer role to the Integration Developer role. You will use Integration Designer to implement the Advance Integration Service (Standard Process AIS) created in Part 1 as a Java component. You will then publish the implemented service back to the Process Center repository to test in Part 3.

NOTE: Leave Process Designer open while you work here in Integration Designer for speed of finishing this lab (if current machine or image's memory allows).

_____ 1.   Start Integration Designer

    __ a. Start Integration Designer in a new workspace by clicking on the Integration Designer from the Windows *Start* menu.
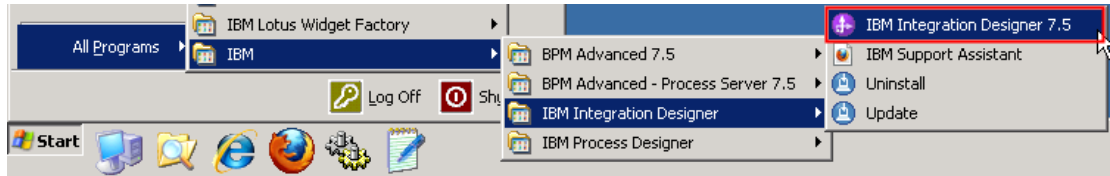
    __ b. In the Workspace Launcher dialog, enter **C:\AppDev\WorkSpaces\Tutorials-RequestApp** for the workspace location.
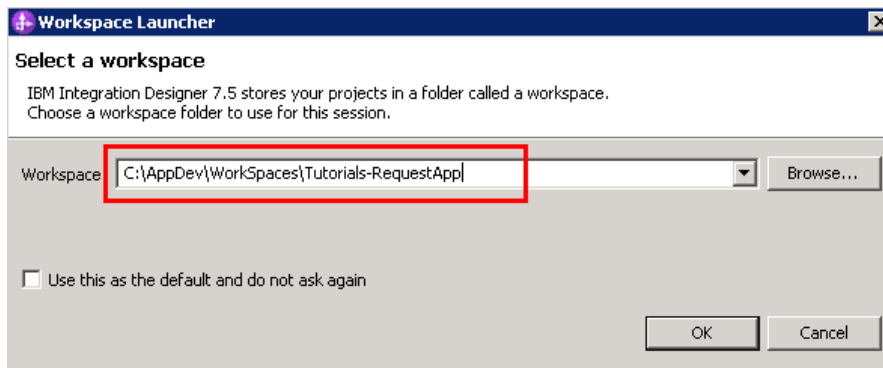
Upon opening a new workspace, Integration Designer will initially open to the default Business Integration perspective but quickly switch to the Process Center perspective in order to login to the Process Center. Logging into the Process Center first is important because you will need to connect to the Process Center in order to open and work on the Advanced Integration Service you created in Process Designer.
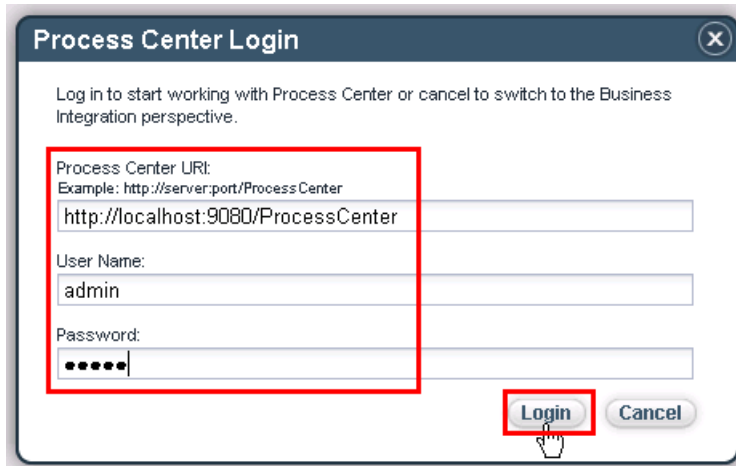
NOTE: The Process Center perspective in the Integration Designer is the 'integration developers' interface to the Process Center. When a process application is opened in the Integration Designer workspace, it is represented from the implementation point of view. This is different than the process application representation which is shown when the process application is opened from the Process Designer.

__ c. **Type the following credentials** into the Process Center Connection Details prompt and click **Login**:
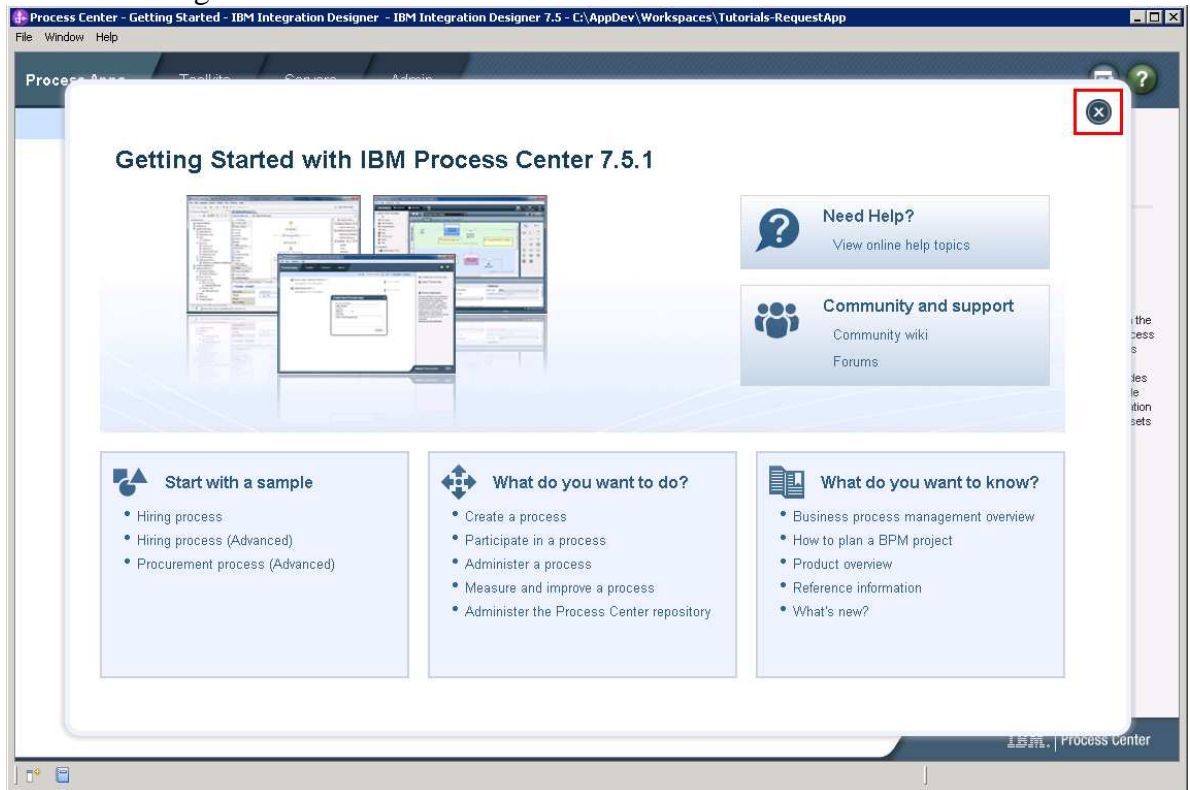
Process Center URL: http://localhost:9080/ProcessCenter
User Name: <your user id>   (default admin user is *admin*)
Password: <your password>   (default admin password is *admin*)

__ d. **Click the X on the welcome dialog** in the Process Center perspective to close the welcome dialog.

_____ 2. **Click the "Open in workspace" link in the same row as the Request Application (JKSREQ) process application.** This action will import the unimplemented service and any necessary schema needed to implement the Advanced Integration Service.
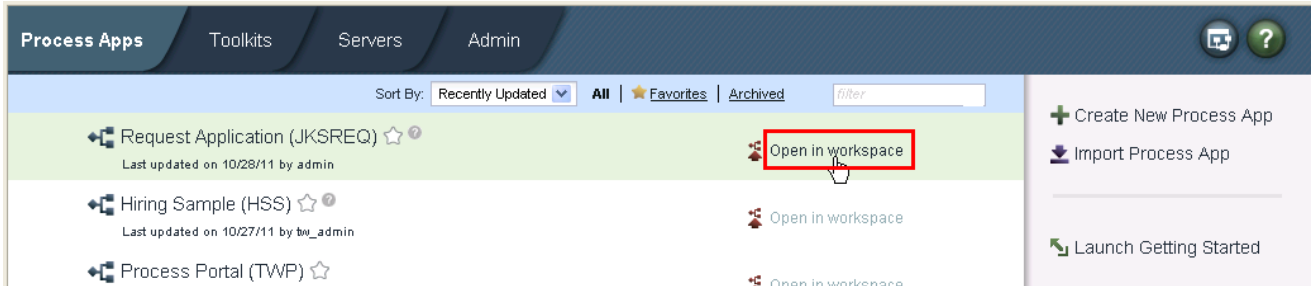

_____ 3. You will be prompted to choose which Process Applications or Toolkits associated with the Request Application process app you want to open in the workspace.
Keep defaults and click OK.
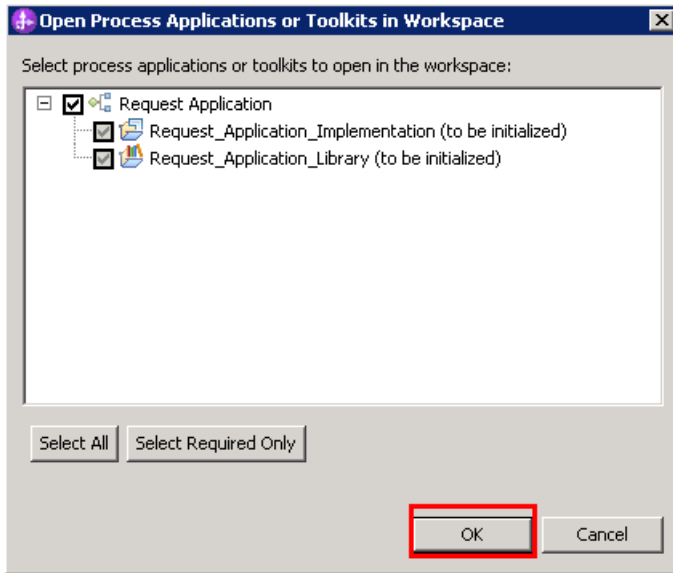
When you click the OK, the process application that contains your unimplemented "Standard Process AIS" is opened in the Business Integration perspective. The process application is the top-level scope/folder created to hold SCA modules. In this case, there are two - the implementation module (where the work will be done) and the library module (where business objects and interfaces artifacts used by the implementation will be stored).
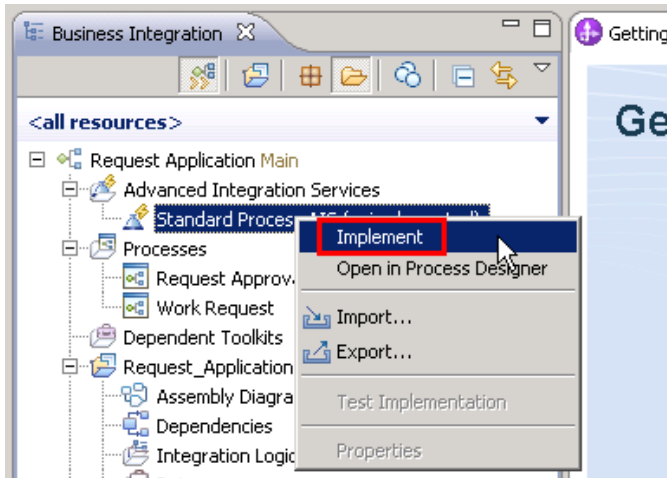
_____ 4. **Click the + signs on all artifacts in the business integration view** to fully expand what was imported from the Process Center repository into Integration Designer, including the implementation and library modules to support the process application.

Notice the artifacts in the modules. It is the responsibility of the integration developer role to implement the Advanced Integration Service using these artifacts given by the process developer.
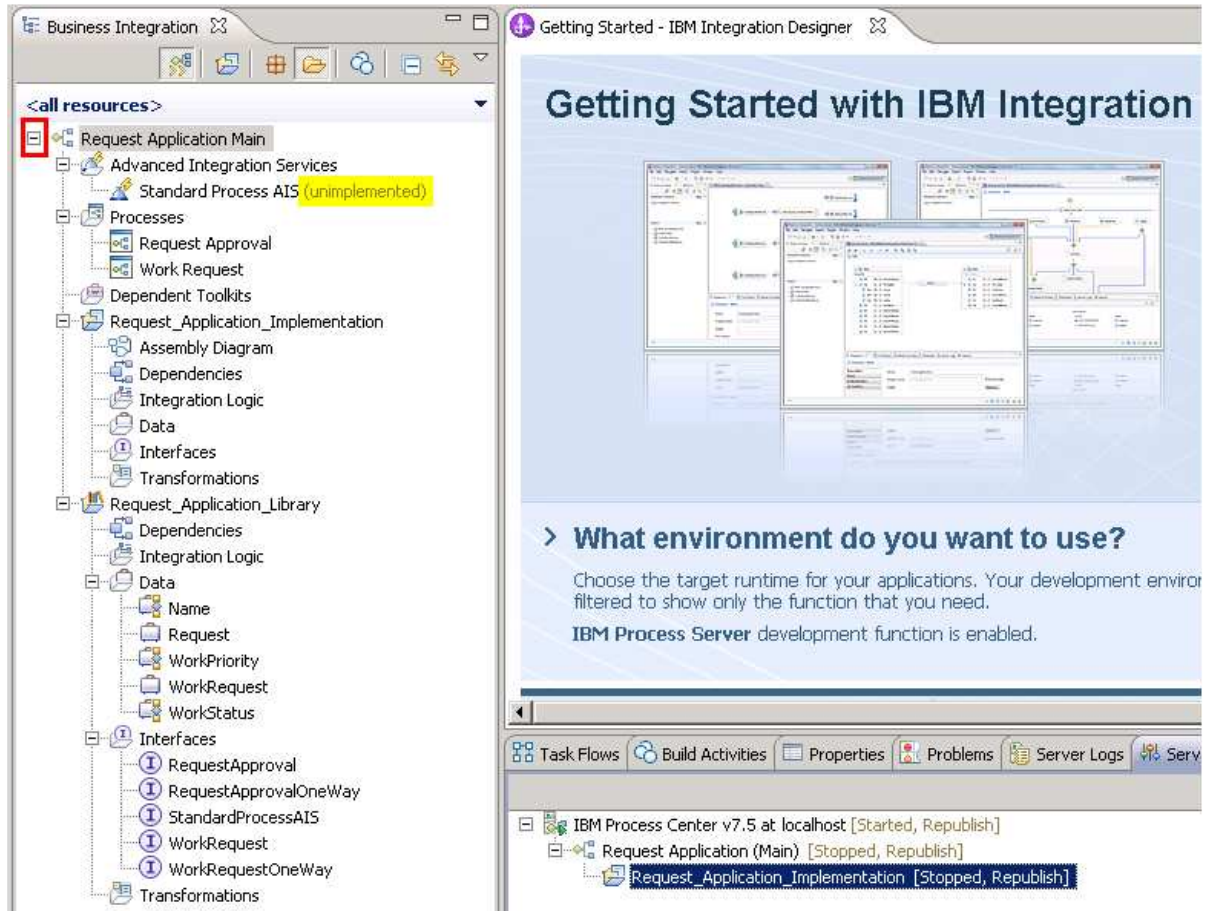

____ 5. Right click **Standard Process AIS** under Advanced Integration Services in the business integration view.

From the pop-up menu, **select "Implement".**

____ 6. The Implement wizard asks you which of the four types you want to use for this Advanced Integration Service. You have the choice of a microflow, a long-running BPEL, a Java component, or an empty implementation.

Select "Java component" in order to generate a Java implementation skeleton for the Advanced Integration Service. The implementation skeleton will be based on the inputs and outputs the business process developer specified when creating the *Advanced Integration Service* in the Process Designer. Click **Finish.**
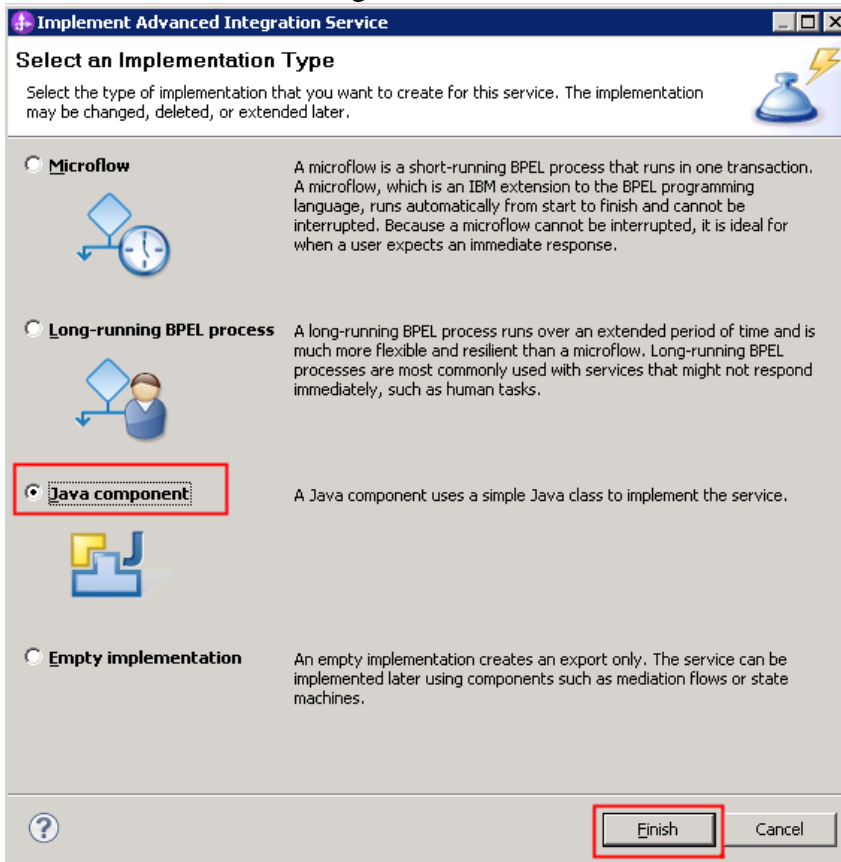
____ 7. **Click the + signs on all artifacts in the business integration view** to fully expand what was automatically generated for you.

Notice the "unimplemented" text next to Standard Process AIS disappeared after generation to signify that the Advanced Integration Service wizard has been run. Notice the Java integration logic in the implementation module.
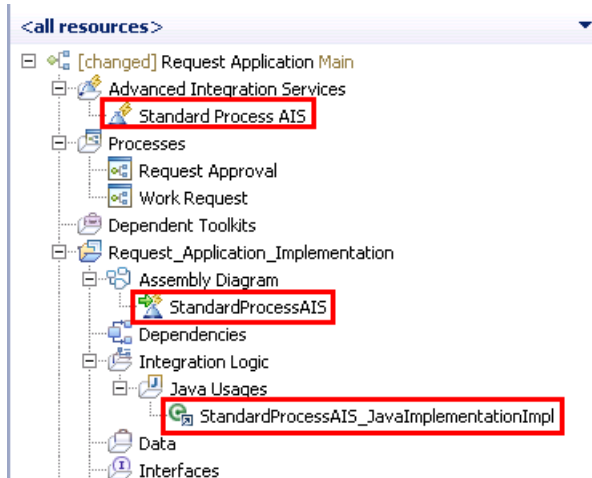
Notice that the Java editor automatically opens on the Java implementation skeleton.
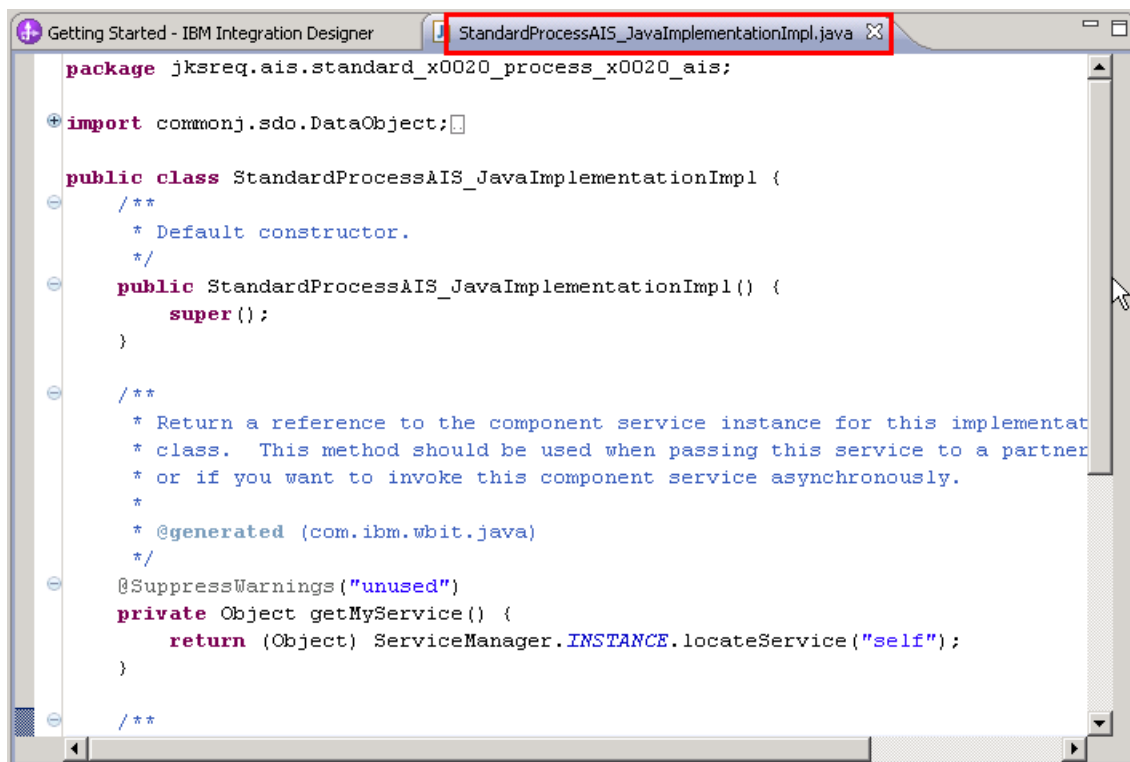
____ 8. **Scroll down to the bottom** of the skeleton Java code that was generated for you, till you see method, **DataObject invoke(DataObject theWorkRequest)**.

```
  @SuppressWarnings("unused")
  private Object getMyService() {
      return (Object) ServiceManager.INSTANCE.locateService("self");
  }

  /**
   * Method generated to support implementation of operation "invoke" defined f
   * named "StandeardProcessAIS".
   *
   * The presence of commonj.sdo.DataObject as the return type and/or as a para
   * type conveys that it is a complex type. Please refer to the WSDL Definitio
   * on the type of input, output and fault(s).
   */
  public DataObject invoke(DataObject theWorkRequest) {
      // To create a DataObject, use the creation methods on the BOFactory:
      // com.ibm.websphere.bo.BOFactory boFactory = (com.ibm.websphere.bo.BOFac
      //
      // To get or set attributes for a DataObject such as theWorkRequest, use
      // To set a string attribute in theWorkRequest, use theWorkRequest.setStr
      // To get a string attribute in theWorkRequest, use theWorkRequest.getStr
      // To set a dataObject attribute in theWorkRequest, use theWorkRequest.se
      // To get a dataObject attribute in theWorkRequest, use theWorkRequest.ge
      return null;
  }
```

__ a. Replace the return null; with the following code. (Copy and paste).

Alternatively, you can open the **Snippets and things.rtf** file under downloaded lab artifacts directory, and copy the snippet under Top-Down Scenario.

```
System.out.println("***** Current Status: " + theWorkRequest.getString("status"));
theWorkRequest.setString("status", "complete");
System.out.println("***** New Status    : " + theWorkRequest.getString("status"));
return theWorkRequest;
```
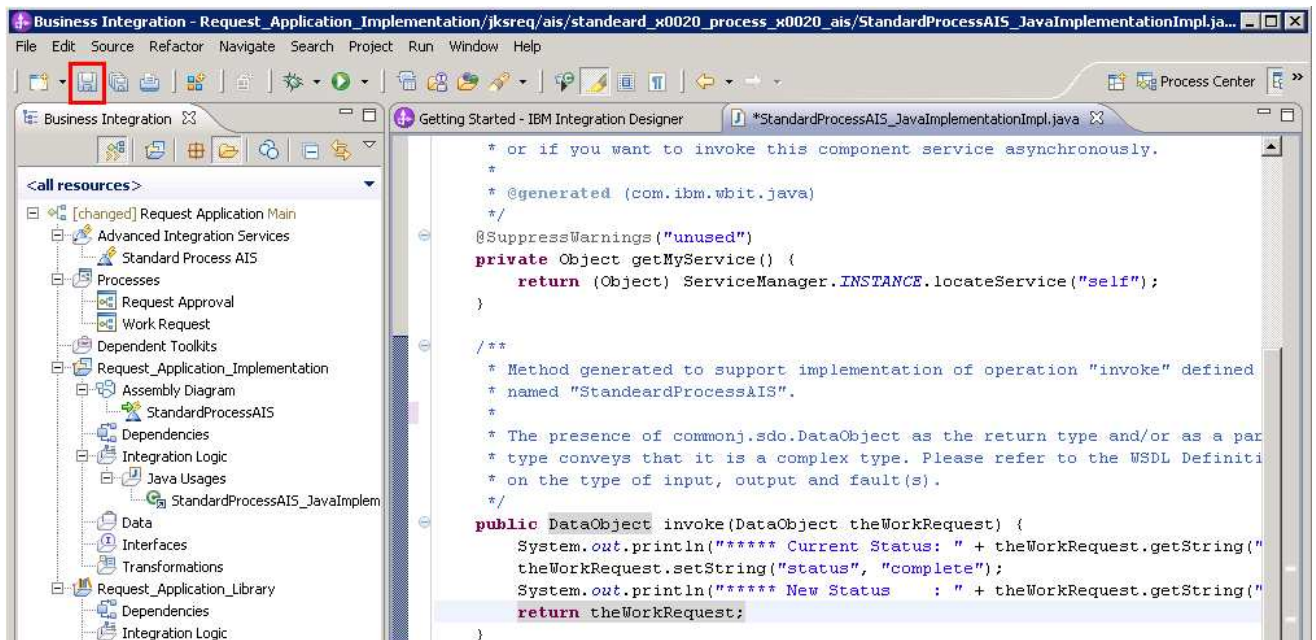
What this code does is print the current status to the standard out log, changes the status to "complete", prints the new status to the standard out log, and returns the data object.

____ 9. **Save the Java implementation file** with Ctrl-S or by clicking **save** (icon highlighted in this screen capture). It is always a good practice to close the window when done.

Notice * symbol in the StandardProcessAIS_JavaImplementation tab disappeared after saving.

____ 10. Publish implemented Advanced Integration Service back to the Process Center repository.

Right-click the top-level process application "Request Application Main" and select Refresh and Publish.
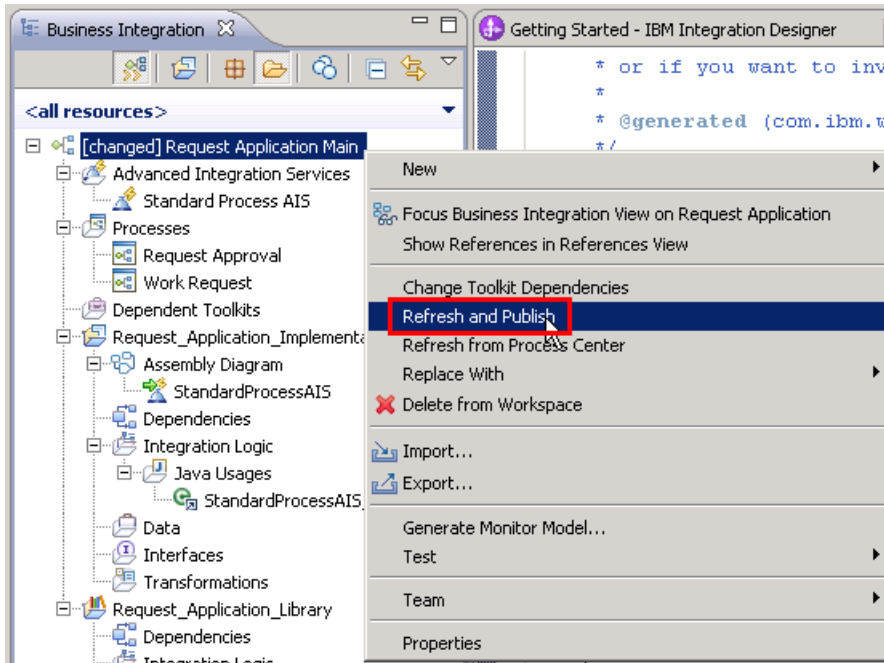
What this action does is publish the changes you just made as the Integration Developer role back into the Process Center repository. If other changes had been made to this process application while you were editing in Integration Designer, you would have received a prompt to compare and merge your changes and whatever else was changed between the time you imported, made your changes, and published.

____ 11. You have completed your part as the integration developer. The Request Application has been published to the Process Center repository with your additions. It is now up to the process developer to test and verify your work.

If you like you can safely delete the Request Application Main from you workspace. The latest version can always be retrieved from the Process Center repository.

At this point you are ready to test the business process in the Process Designer's Inspector for Part 3.

Part 3: Process Designer - Test the Work Request business process

In this section, your role will be changing from the Integration Developer role to the Process Developer. You will use **Process Designer** to step through the Work Request business process and test the newly created **Standard Process AIS** *implementation* in the context of the business process flow.

____ 1. Switch back to **Process Designer** from **Part 1** and open the **Work Request** BPD.
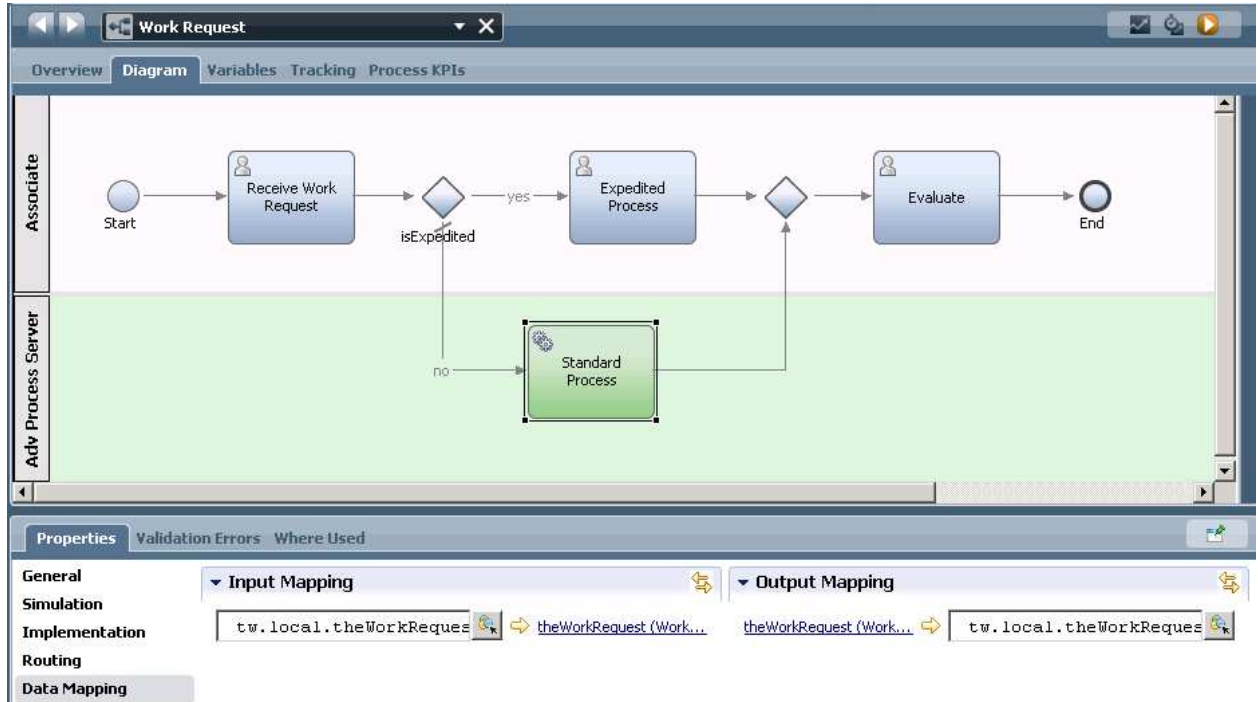
____ 2. You have finished the implementation of the Standard Process AIS service. Take a look at the implementation information.

     __ a. Select **Standard Process** activity in the diagram and choose the **Implementation** section under **Properties** tab.

   __ b. Click the **Standard Process AIS** link. The AIS overview page opens.
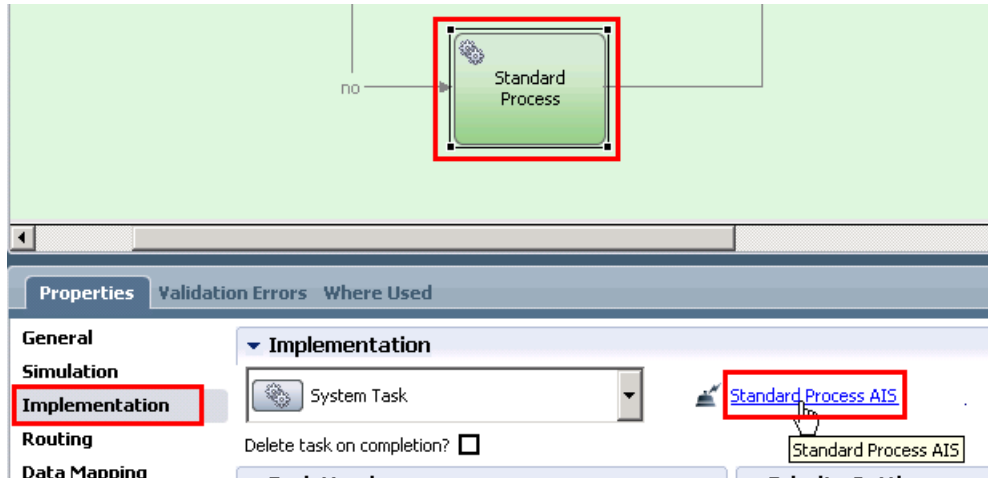
__ c. You can see the AIS implementation information you did in Integration Designer are displayed.
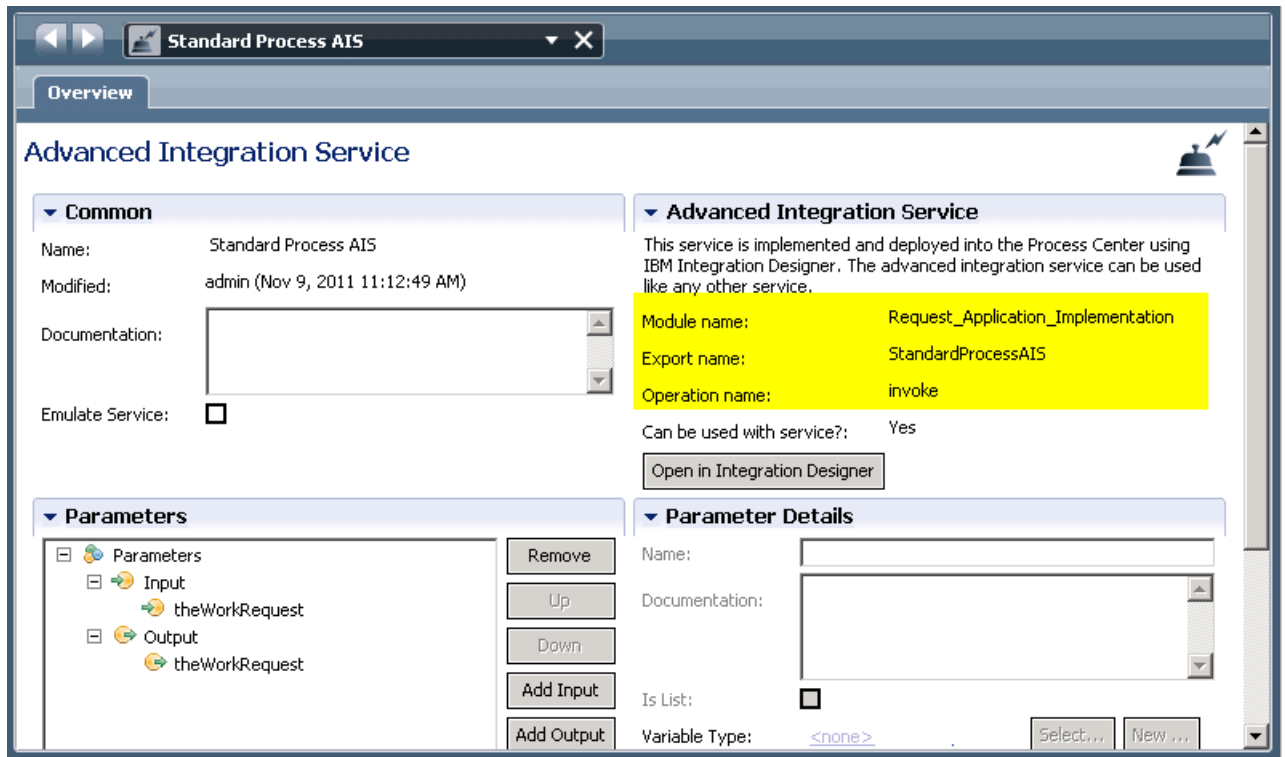
__ d. Click the **X** next to **Standard Process AIS** to close this overview dialog and return to Work Request process flow.

____ 3. Execute and run **Work Request** business process definition in the **Inspector** view of **Process Designer**
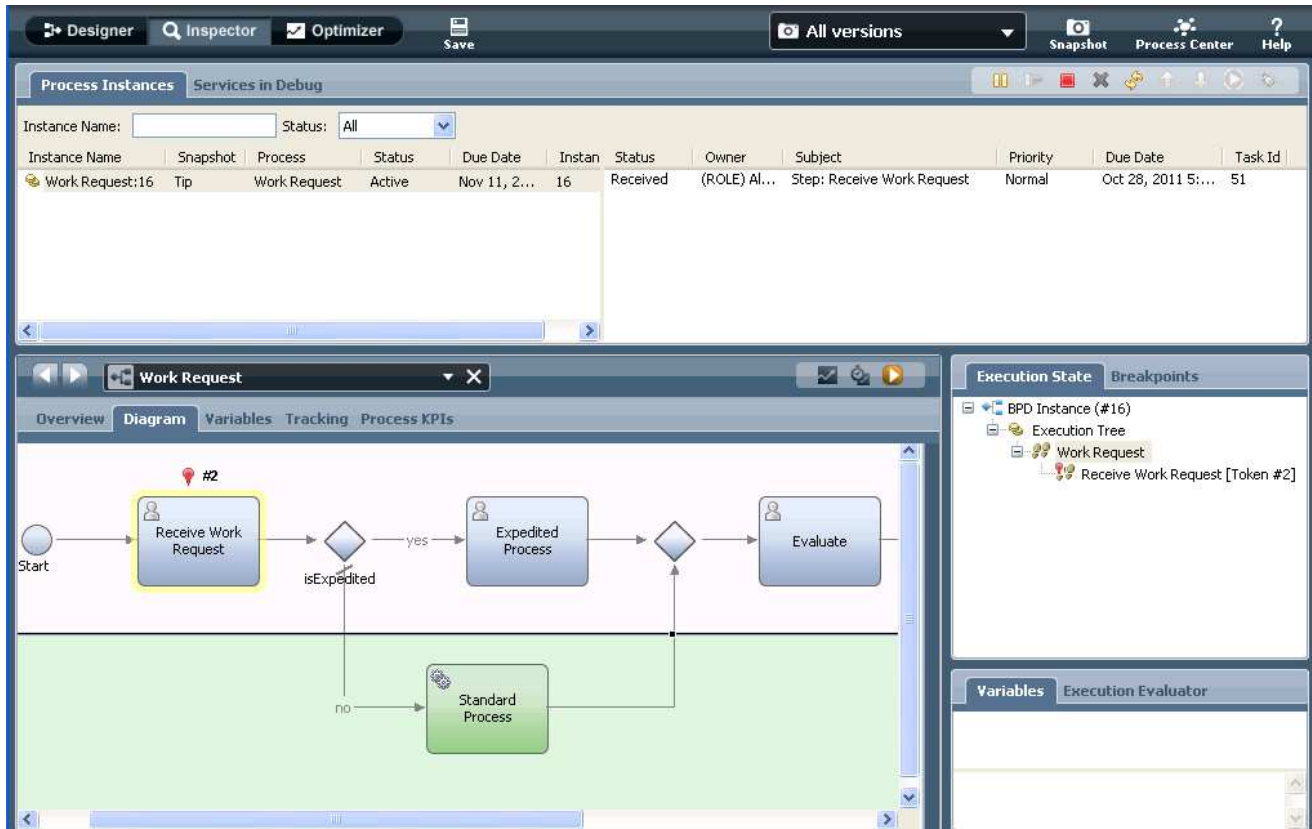
__ a. In top right corner click **Run Process**
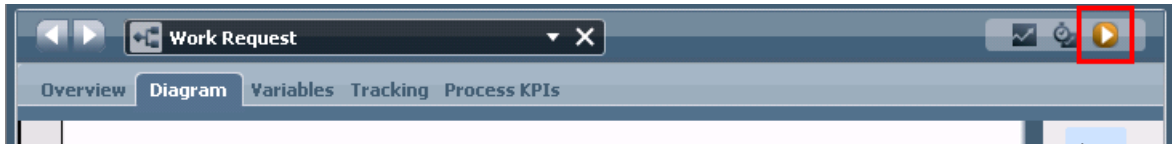
__ b. If you get a **Switch View?** pop-up dialog, select **Yes.** You can also check **Remember decision?** if you want to switch to Inspector view for all BPD tests in the future.
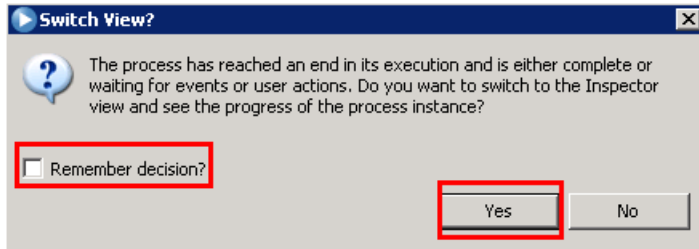
After you click **Yes** the **Inspector** view will open

__ c. Select the **Work Request** process instance displayed under the **Process Instances** tab (for example, Work Request:16 in the screen capture below). The current task for the selected process instance is shown on the right side. You will also see a red token indicating which activity the current instance is on in the lower **Diagram** view.
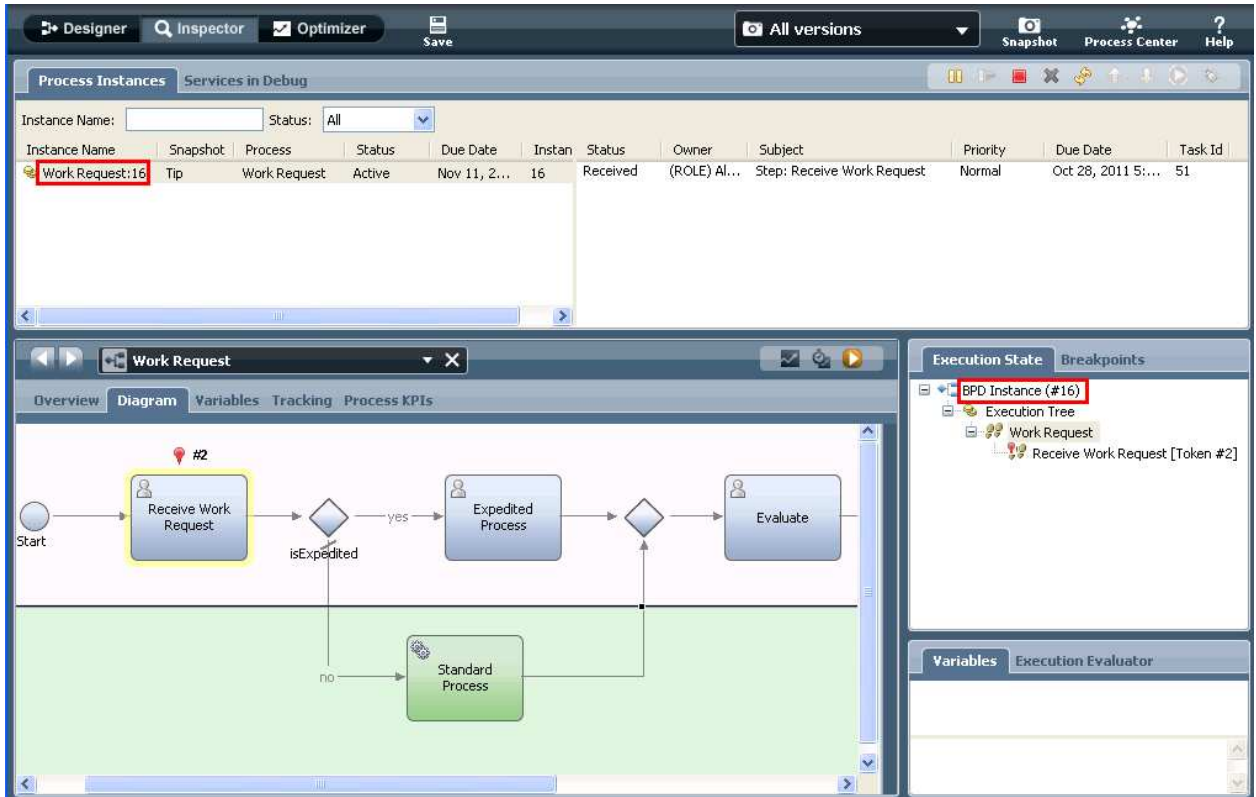
The current task for the selected process instance in the image is **Receive Work Request** activity. This is also as indicated by the red token and the **Execution State** tab on the right.

__ d. Under the task table select the **Receive Work Reques**t task and click **Run the selected task** icon.
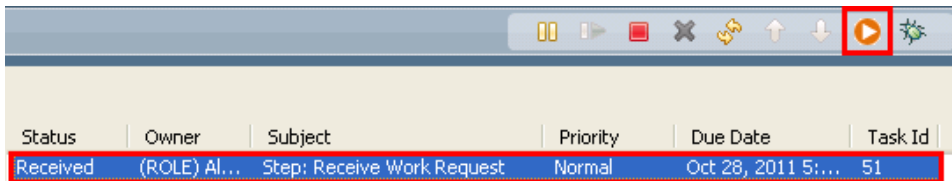
__ e. Select the user that ID should run this task. For the purposes of this lab, select **admin** (or whatever role is appropriate for your installation) and **select OK.**
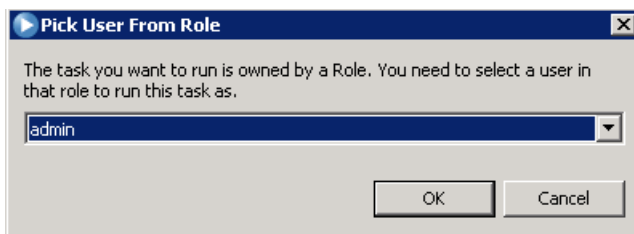
__ f. Once you click **OK**, your default local system browser will open to the coach within the **Receive Work Request → Submit Work Request** service

Notice that the fields are pre filled with default values as initialized in the **Submit Work Request** service. Also note the current value of **Status** is **New Request.**

__ g. Select **OK**.

__ h. This completes the **Submit Work Request** service. Close the browser window and return to **Process Designer**

__ i. Click the **Refresh** icon to see the next step in the **Work Request** process.

Notice after you click Refresh the status of **Receive Work Request** changes to **Closed**. Also the status of **Standard Process** automatically becomes closed since there is no human intervention needed for this advanced integration service.


Now move to the next step called **Step: Evaluate**

__ j. Select **Step: Evaluate** and click the **Run the selected task** icon . Again in the **Pick User From Role** dialog select user **admin** and select **OK.**

__ k. Verify that the process variable **Status** field has been updated by the advance integration service called **Standard Process AIS** to **complete.** Select **OK**
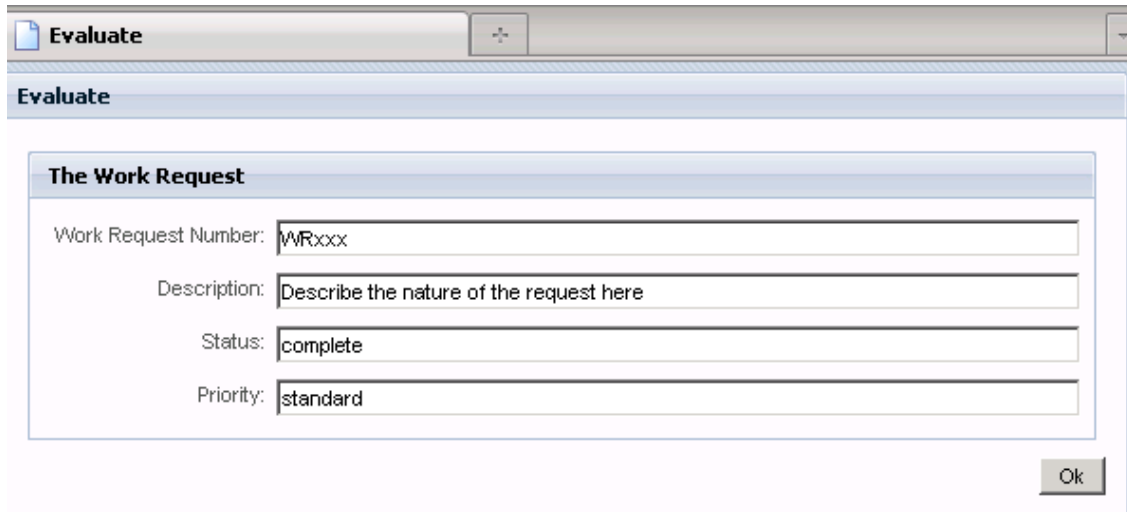
__ l. This completes the **Evaluate** service. Close the browser window and return to **Process Designer**

__ m. Above the task list select the **Refresh** icon to see the complete execution of the **Work Request** process
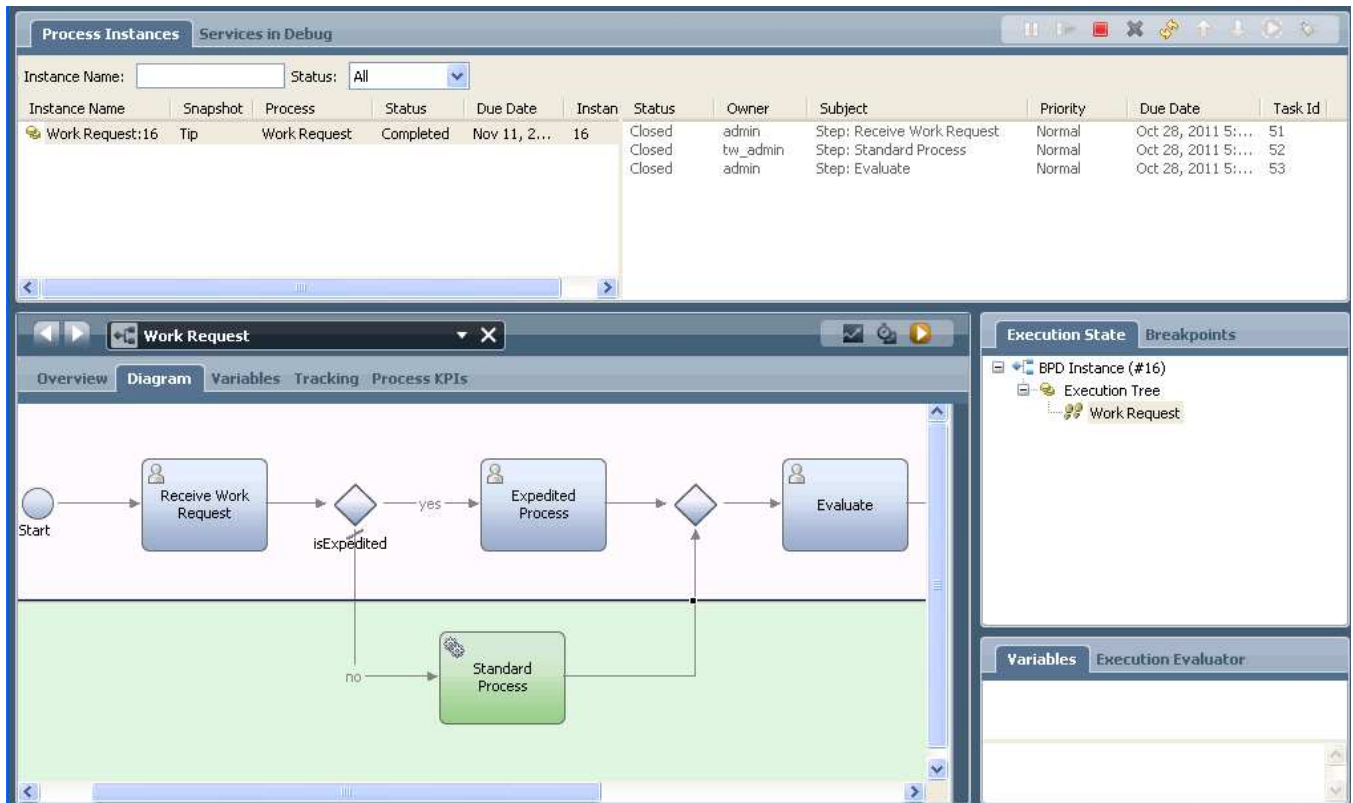
You have successfully defined, implemented, and run a top down implementation of a business process integrated with an Advance Integration Service.

What you did in this exercise

In this lab you did the following:
- Imported an existing business process definition into the Process Center
- Defined an Advanced Integration Service in the Process Designer
- Configured the Integration Designer to connect to the Process Center
- Implemented the Advanced Integration Service using the Integration Designer
- Published and synchronized changes between the Integration Designer and the Process Center
- Tested a business process definition that uses an advanced integration service in Process Designer.