

Course Exercises Guide

Developing workflow solutions using IBM Business Automation Workflow V20.0.0.1

Course code WB835 / ZB835 ERC 1.0



December 2020 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	vi
Exercises description	vii
How to follow the exercise instructions	viii
Exercise structure	viii
User IDs and passwords	x
Exercise 1. Creating your first process application	1-1
Part 1: Start the IBM Workflow Center deployment environment	1-3
Part 2: Log in to the IBM Workflow Center and set preferences	1-7
Part 3: Create and manage a process application	1-14
Part 4: Explore other tools	1-18
Part 5: Export the process application	1-24
Part 6: Import the process application	1-29
Exercise 2. Playback 0: Creating the To-Be process	2-1
Part 1: Create a process	2-4
Part 2: Create process activities	2-8
Part 3: Model the process teams	2-15
Part 4: Assign the activities to the appropriate Team	2-18
Part 5: Create a linked process	2-23
Part 6: Attach the linked process	2-30
Exercise 3. Playback 0: Controlling process flow	3-1
Part 1: Create gateways for parent process	3-3
Part 2: Create gateways for the linked process	3-15
Part 3: Modeling timer intermediate events	3-24
Part 4: Add new process requirements to the process	3-32
Part 5: Validate the process model and take a snapshot	3-40
Exercise 4. Playback 1: Controlling process flow with business data	4-1
Part 1: Implement the intermediate timer event	4-3
Part 2: Create the process flow variables for the Hiring Request Process	4-4
Part 3: Create process flow variables for the Approve Hire Request linked process	4-8
Part 4: Implement gateways	4-12
Part 5: Implement routing for an activity	4-20
Part 6: Review artifact references	4-26
Exercise 5. Playback 1: Business data, services, and coaches	5-1
Part 1: Build business objects and variables	5-3
Part 2: Create a reusable client-side human service and define input controls	5-14
Part 3: Reuse and debug the client-side human service	5-36
Exercise 6. Playback 1: User interface design and implementation	6-1
Part 1: Group controls into tabs on a coach	6-3
Part 2: Change the appearance of the coach by applying a custom theme	6-12
Part 3: Customize controls	6-23
Part 4: Modify the mobile layout	6-29

Exercise 7. Playback 1: Conducting the Playback session	7-1
Part 1: Prepare for the playback	7-3
Part 2: Demonstrate the Review Needed path	7-4
Part 3: Demonstrate the Review Not Needed path	7-14
Part 4: Demonstrate activity tracking	7-19
Part 5: Reset the environment	7-23
Part 6: Create the Hiring Requisition Toolkit and take a snapshot	7-24
Exercise 8. Playback 2: Integrations	8-1
Part 1: Create a decision service	8-3
Part 2: Implement a message start event	8-21
Part 3: Apply asset tags	8-31
Part 4: Create a service to query a database and populate a list	8-34
Part 5: Change an input to a single select on a coach	8-55
Exercise 9. Playback 3: Handling errors and deploying your process application	9-1
Part 1: Implement exception handling in a service	9-2
Part 2: Prepare for final snapshot	9-9
Part 3: Check the history of the process application	9-12
Part 4: Create a snapshot of the process application	9-12
Exercise 10. Implementing a custom routing solution	10-1
Part 1: Add code to determine whether it is the weekend	10-2
Part 2: Add custom routing to the HR Recruitment process	10-4
Part 3: Test the custom routing solution	10-8
Part 4: Clean up the environment	10-11
Exercise 11. Implementing the “four eyes” policy by using a team filter	11-1
Part 1: Modify the process	11-2
Part 2: Create the team filter service	11-7
Part 3: Apply the team filter service to the activity assignment	11-11
Part 4: Test the four eyes policy	11-12
Exercise 12. Building a cancellation pattern	12-1
Part 1: Create the event	12-2
Part 2: Create an undercover agent	12-4
Part 3: Implement the boundary event with the UCA	12-5
Part 4: Create the wrapper service	12-8
Part 5: Create a test harness	12-10
Part 6: Test the cancellation UCA	12-14
Exercise 13. Building web service connections	13-1
Part 1: Create an inbound web service to trigger the UCA	13-2
Part 2: Test the inbound integration by creating an outbound integration	13-4
Part 3: Test the integrations	13-9
Exercise 14. Handling content events in a process	14-1
Part 1: Implement a document start event on a process by using the BPM document store	14-3
Part 2: Configure the Document Viewer to use the Target Object Store (TOS)	14-26
Part 3: Build Case solution	14-32
Part 4: Test your solution	14-45
Part 5: Export the process application	14-52
Appendix A. Solutions	A-1
Part 1: Export current process application	A-2

Part 2: Perform troubleshooting	A-6
Part 3: Import a solution file	A-7

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Bluemix®
SPSS®

Cognitive Era™
Watson Avatar®

Cognos®
Worklight®

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

This course includes the following exercises:

- Exercise 1. Creating your first process application
- Exercise 2. Playback 0: Creating the To-Be process
- Exercise 3. Playback 0: Controlling process flow
- Exercise 4. Playback 1: Controlling process flow with business data
- Exercise 5. Playback 1: Business data, services, and coaches
- Exercise 6. Playback 1: User interface design and implementation
- Exercise 7. Playback 1: Conducting the Playback session
- Exercise 8. Playback 2: Integrations
- Exercise 9. Playback 3: Handling errors and deploying your process application
- Exercise 10. Implementing a custom routing solution
- Exercise 11. Implementing the "four eyes" policy by using a team filter
- Exercise 12. Building a cancellation pattern
- Exercise 13. Building web service connections
- Exercise 14. Handling content events in a process

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

Most exercises include required sections, which should always be completed. It might be necessary to complete these sections before you can start later exercises. Some exercises also include optional sections that you might want to complete when you have sufficient time and want an extra challenge.

This course has been tested in Firefox. Firefox is the default browser. However, Chrome is available. Keep in mind, the behavior may be different. If Firefox prompts to install updates, decline the updates.



Important

The exercises in this course use a set of lab files that might include scripts, applications, files, solution files, project interchange files, and others. The course lab files can be found in the following directory:

C:\labfiles for the Windows platform

The exercises point you to the lab files as you need them.

How to follow the exercise instructions

Exercise structure

Each exercise is divided into sections with a series of steps and substeps. The step represents an action to be completed. If required, the substeps provide guidance on completing the action:

- The numbered steps (1, 2, 3) describe what actions to do.
- The lettered substeps (a, b, c) provide detailed guidance on how to complete the action.

1+1=2 Example

As shown in this example, the numbered step ("3") tells you to change the value in a rule. Substeps "a" and "b" provide details on how to edit.

3. Edit the rule and change the debt-to-income ratio from 0.3 to 0.5.

a. Click **Edit this rule** (the pencil icon) to open the rule editor.



b. In the **if** part of the rule, change **0 .3** to: **0 .5**

if
the yearly repayment of '**the loan**' is more than the yearly income of '**the borrower**' * **0 .5**

If you already know how to edit rules and change values, you can skip the substeps.

Text highlighting in exercises

Different text styles indicate various elements in the exercises.

Words that are highlighted in **bold font** represent GUI items that you interact with, such as:

- Menu items
- Field names
- Icons

Words that are highlighted with a `code font` include the following items:

- Text that you type or enter as a value
- System messages
- Directory paths
- Code

Tracking your progress

As shown in the example step, you can see that an underscore precedes each numbered step and lettered substep.

You are encouraged to use these markers to track your progress. As you complete a step, place an **X** or a check mark on the underscore to indicate that it is completed. By tracking your progress in this manner, you can stay focused when you experience interruptions during a lengthy exercise.

Required exercise sections

Most exercises include required sections that should always be completed. It might be necessary to complete these sections before you can start subsequent exercises.

Dependencies between exercises are listed in the exercise introduction.

Optional exercise sections

Some exercises might also include optional sections that you can complete when you have sufficient time and want an extra challenge.



Stop

Course updates and errata



A Course Corrections document might be available for this course.

If you are taking the class with an instructor, the instructor can provide this document to you.

If you are taking the course in a self-paced environment, the course corrections document is provided with the other manuals.

To check whether a Course Corrections document exists for this course:

1. Go to the following URL: <https://ibm-learning-skills-dev.github.io/>
2. On the web page, click the **IBM Cloud courses** category.
3. Find your product in one of the categories and click the link to open the category.
4. Find your course in the list and click the link to see whether an errata document exists with updated instructions.
5. To save the file to your computer, click the document link and follow the dialog box prompts.

User IDs and passwords

Table 1. User accounts and passwords

Type	Username	Password
Windows Server 2016	administrator	passw0rd
IBM Business Automation Workflow	author1	passw0rd
WebSphere Application Server cell administrator	administrator	passw0rd

Exercise 1. Creating your first process application

Estimated time

01:00

Overview

This exercise introduces Business Automation Workflow and the creation of your first process application. You are also introduced to various tools that can be used for troubleshooting. After creating your process application, you learn how to export and import snapshots.

Objectives

After completing this exercise, you should be able to:

- Start IBM Business Automation Workflow
- Create a process application in Workflow Center
- Use the Process Admin console and other tools to aid in troubleshooting
- Export and import process applications

Introduction

IBM Business Automation Workflow is a platform to create workflow applications to improve productivity. Workflow applications coordinate work between tasks that are performed by humans and automated tasks to improve daily business operations.

IBM Business Automation Workflow is part of the IBM Digital Business Automation platform, which includes various product offerings that you can use to digitize all aspects of business operations, while providing real-time insight into your business.

Requirements

None

Index

[Part 1: Start the IBM Workflow Center deployment environment](#)

[Part 2: Log in to the IBM Workflow Center and set preferences](#)

[Part 3: Create and manage a process application](#)

[Part 4: Explore other tools](#)

[Part 5: Export the process application](#)

[Part 6: Import the process application](#)

Exercise instructions

Part 1: Start the IBM Workflow Center deployment environment

Before you can start the IBM Workflow Center deployment environment, three server configurations must be started. After logging on to the lab environment, start the Deployment Manager profile, the Node Agent profile, and the Deployment Environment.



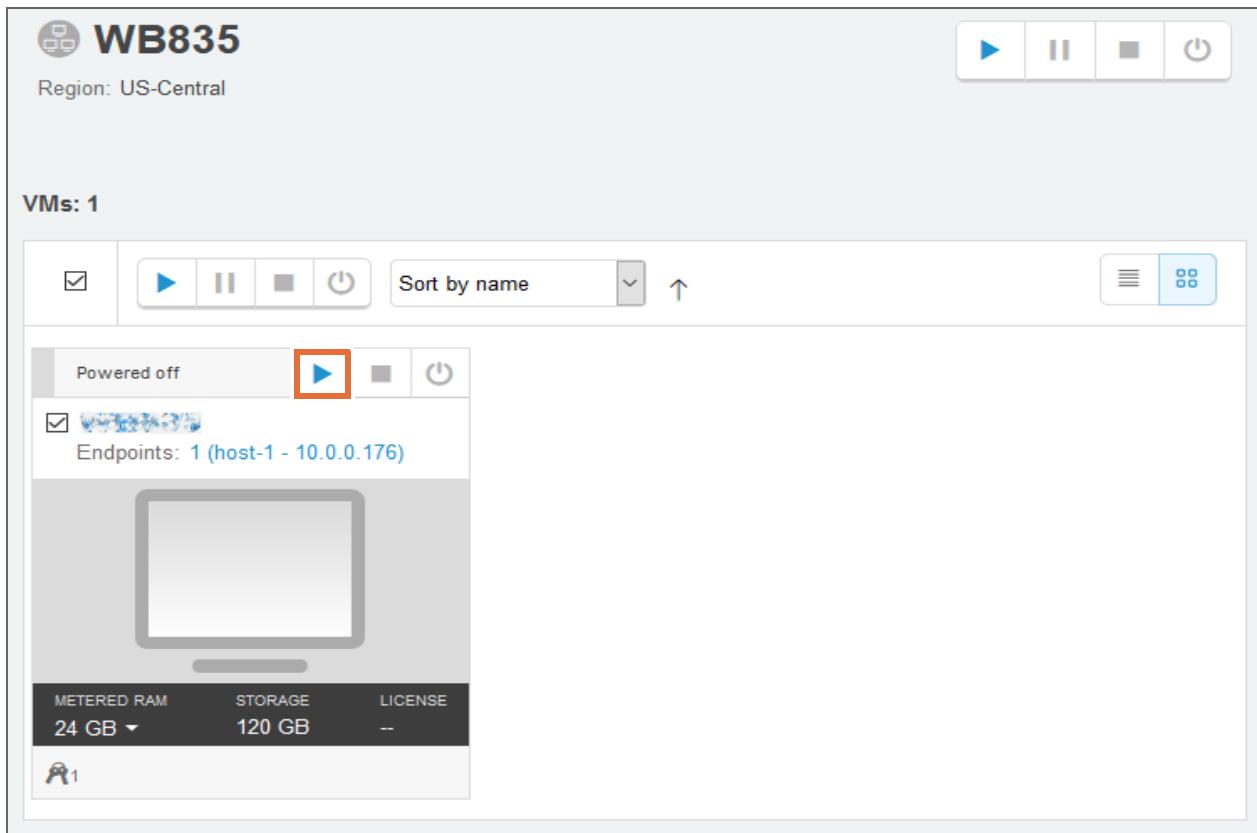
Important

All three server configurations must be started in order, starting with the Deployment Manager profile, then the Node Agent profile, and followed last by the Deployment Environment. To accomplish this task, the product provides a Quick Start routine to run the server start in that order.

- 1. Start the lab environment.

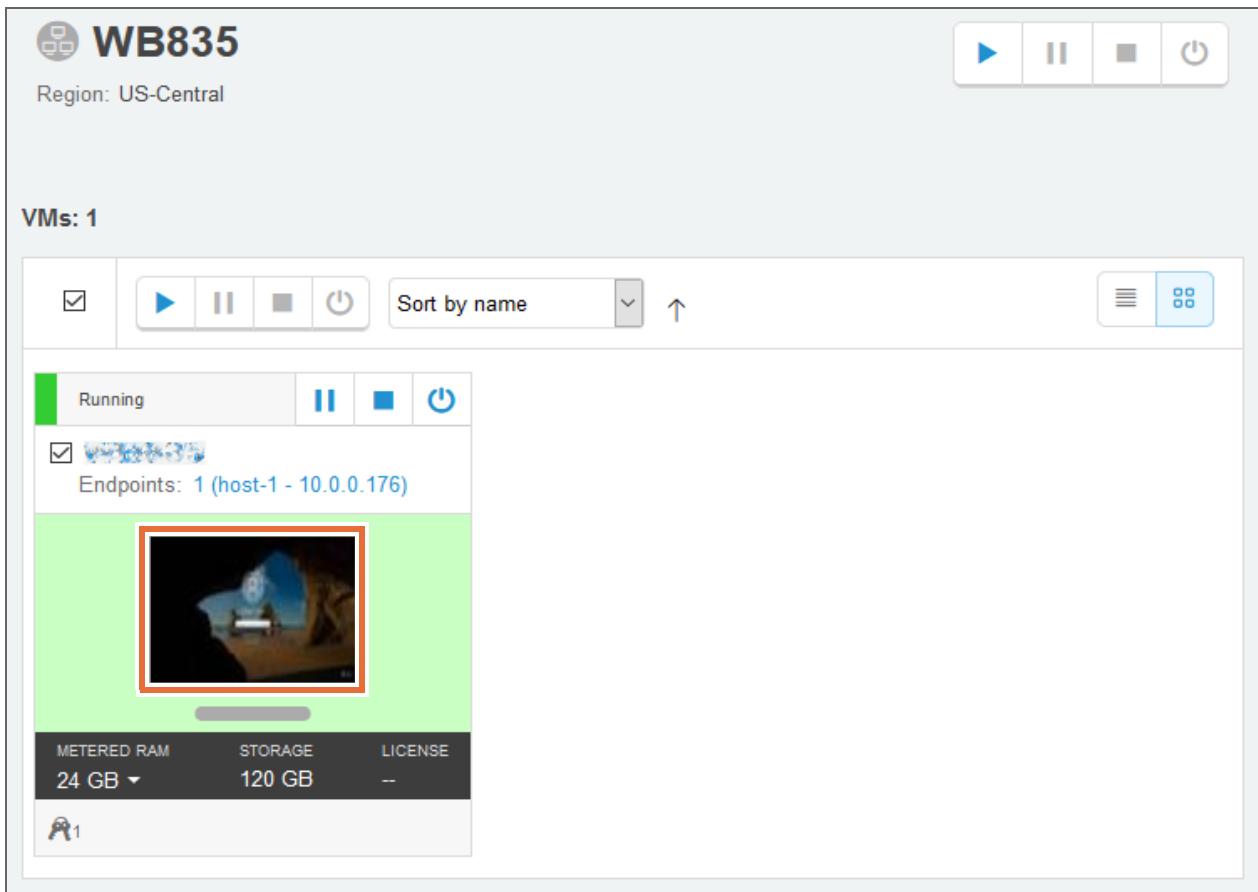
When you receive your lab environment, the environment is in a Stopped state.

- a. To start the lab environment, click **Run this VM**.



It may take several minutes for the lab environment to start.

- __ b. When the environment is in a started state, it turns green and the status changes to Running. Click the computer icon to enter the environment.



The image in the computer might be different.

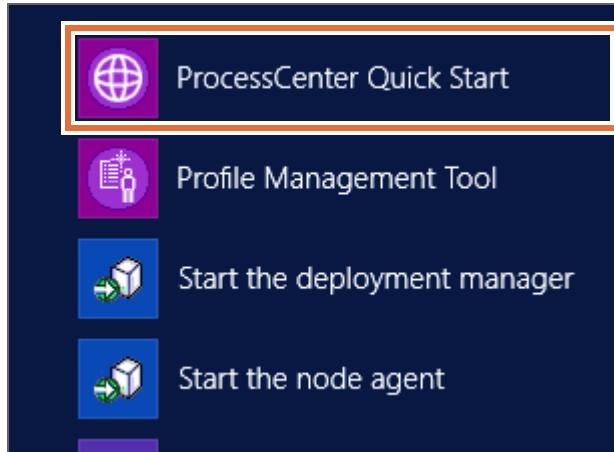
- __ 2. Start the **Workflow Center** server.
- __ a. Log in to the Windows environment using the following credentials:
Username: administrator **Password:** passw0rd
- __ b. Click the **Windows Icon**.



- ___ c. In the Start menu, scroll down to the **IBM** group and click the **Down Arrow** link next to it.



- ___ d. Click **ProcessCenter Quick Start**.



- ___ e. In the **IBM Business Automation Workflow Quick Start** window that opens, click the **Start the Workflow Center Deployment Environment** link.



It takes a while for the deployment environment to get started. A command prompt window runs through the start of the Deployment Manager profile, Node Agent profile, and Deployment Environment. Allow the entire start process to complete. The Quick

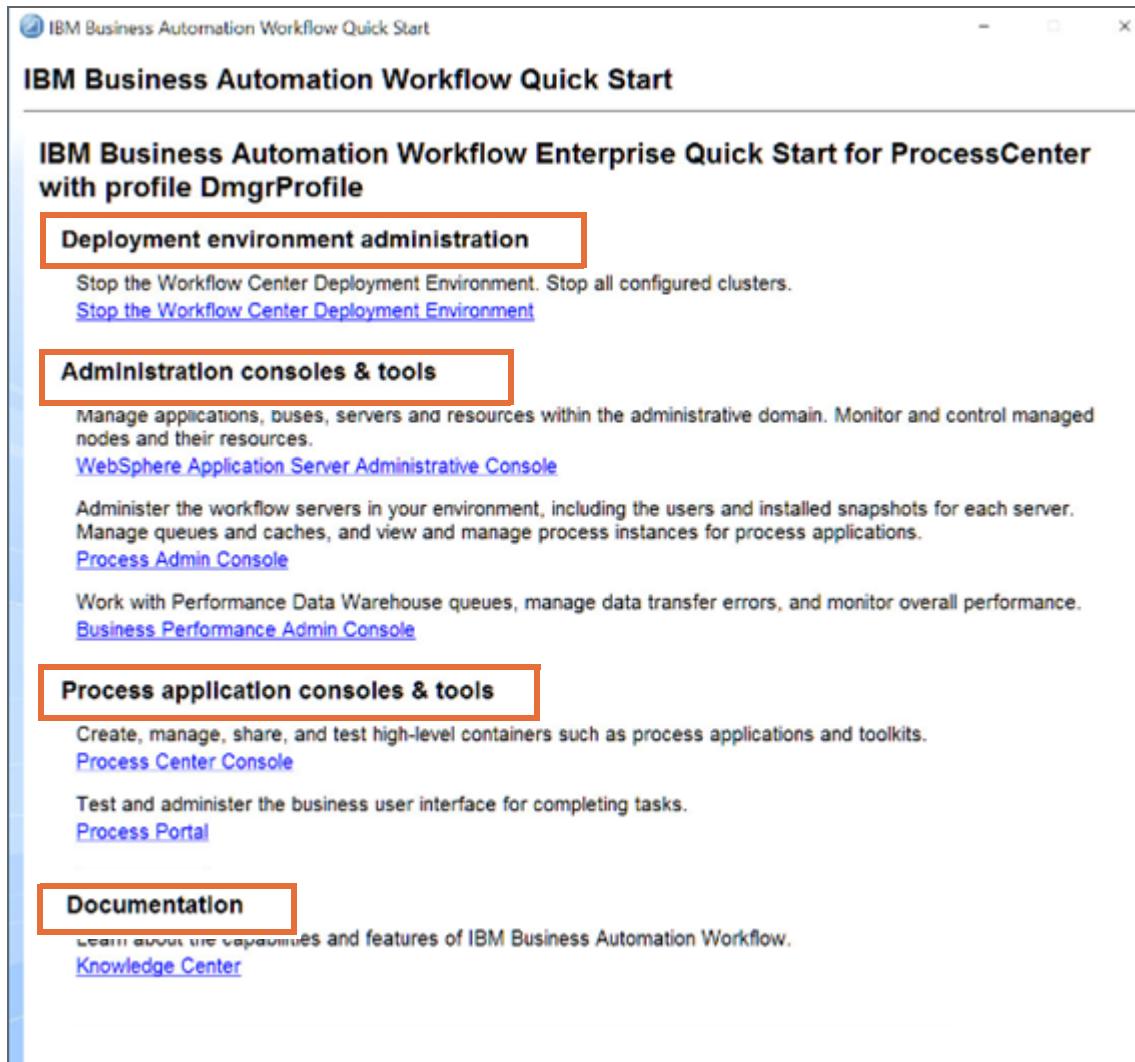
Start window displays that the environment is starting. It can take about 10 – 20 minutes, so make sure that you provide ample time for this initial start.



- ___ f. Switch to the newly open command window. When the environment starts, the **Press any key to continue** message is displayed. When prompted, press any key. You might have to press a key multiple times, at which point the command window closes.

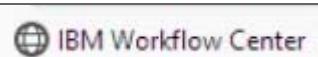
```
C:\Windows\system32\cmd.exe
CWUP0001I: Running configuration action detectNewProducts.ant
CWUP0001I: Running configuration action ejb-deploy-clear-cache.ant
CWUP0001I: Running configuration action clearOSGiCache.ant
CWUP0001I: Running configuration action runiscdeploy.ant
CWUP0001I: Running configuration action clearClassCache.ant
ADMU0116I: Tool information is being logged in file
          C:\IBMWWorkflow\v18.0\profiles\dmgrProfile\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the DmgrProfile profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 812
Starting node Node1.
CWUP0001I: Running configuration action detectNewProducts.ant
CWUP0001I: Running configuration action ejb-deploy-clear-cache.ant
CWUP0001I: Running configuration action clearOSGiCache.ant
CWUP0001I: Running configuration action runiscdeploy.ant
CWUP0001I: Running configuration action clearClassCache.ant
ADMU0116I: Tool information is being logged in file
          C:\IBMWWorkflow\v18.0\profiles\Node1Profile\logs\nodeagent\startServer.log
ADMU0128I: Starting tool with the Node1Profile profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 6148
Starting cluster SingleCluster.
When the BPMConfig command is used to start a deployment environment, it invokes the processes that are used to start the associated clusters. If the command is successful in invoking the processes, it returns a message to report that the command completed successfully. However, to determine whether the cluster members were all started successfully, you need to check the log files of the cluster members. The log files are located in <profile_root>/logs.
The 'bpmpconfig.bat -start -profile DmgrProfile -de ProcessCenter' command completed successfully.
Press any key to continue . . .
```

- ___ g. The Quick Start window refreshes and updates the display. It might take up to 10 – 20 minutes for the servers to fully engage depending on the hardware configuration. Wait for the four sections to be populated in the **IBM Business Automation Workflow Quick Start** window (Deployment environment administration, Administration consoles and tools, Process application consoles and tools, and Documentation).



Part 2: Log in to the IBM Workflow Center and set preferences

- ___ 1. The Workflow Center includes a repository for all processes, services, and other assets created in IBM Process Designer. The Workflow Center provides the tools that you need to maintain the repository. You now connect to the **IBM Workflow Center**.
 - ___ a. Minimize both the **IBM Business Automation Workflow Quick Start** window and the **ProcessCenter Quick Start** command window.
 - ___ b. Open Firefox and click the link on the Bookmark toolbar for **IBM Workflow Center**

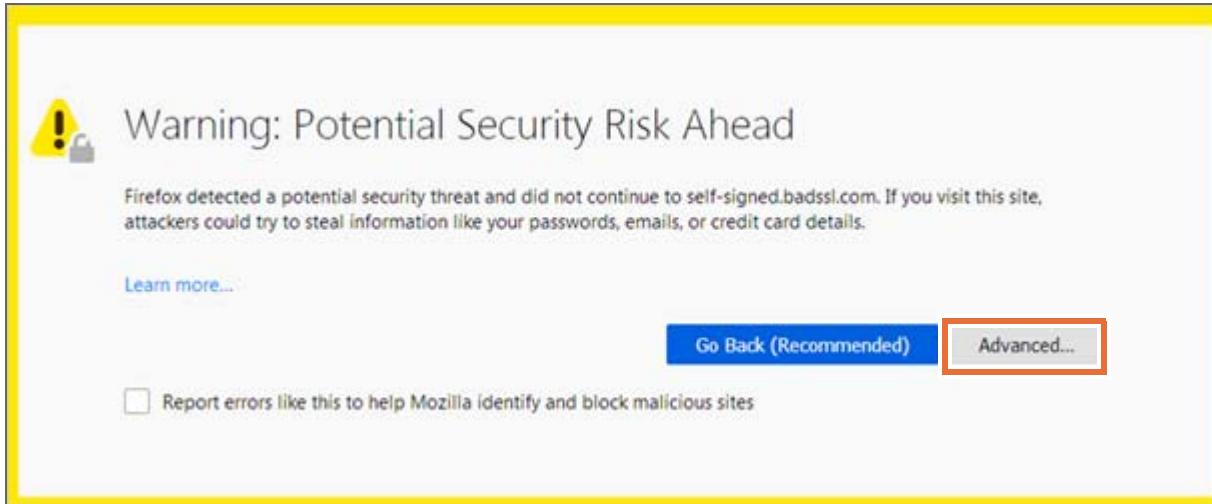




Information

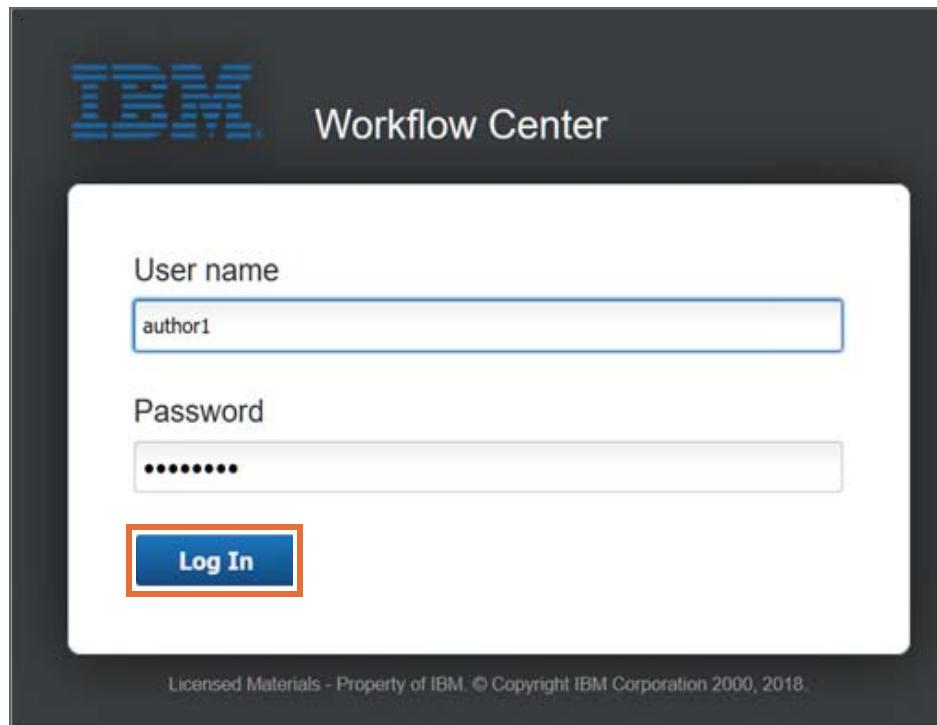
The new terminology for Process Center is Workflow Center. While the new term is used throughout the product, there are times when the product still makes a reference to Process Center.

- ___ c. If a Firefox security warning message is displayed, it is safe to ignore that. Click **Advanced** to continue. Depending on the security message, you might have to click a few times.



A loading screen and then a login screen are displayed for Workflow Center.

- ___ d. Enter `author1` in the **User name** field and `passw0rd` in the **Password** field.
___ e. Click **Log In**.





Information

The **Workflow Center** repository URL is: <https://ws2016x64:9443/WorkflowCenter/>

The **Classic Workflow Center** repository URL is: <https://ws2016x64:9443/ProcessCenter/>

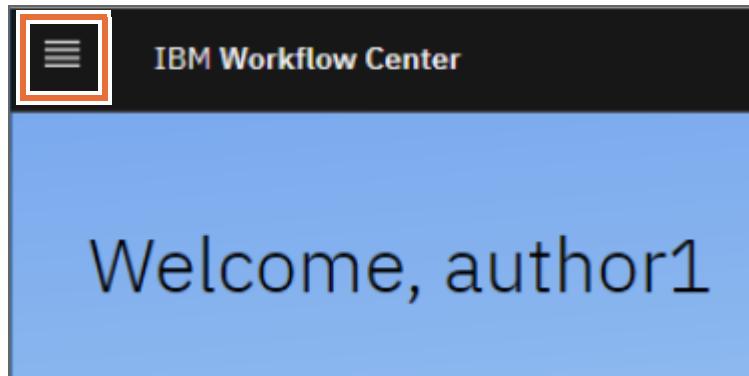
You are now logged in to the Workflow Center as author1.

The screenshot shows the IBM Workflow Center interface. At the top, there is a navigation bar with icons for settings, search, help, notifications, and user profile (author1). Below the header, a large blue banner says "Welcome, author1". To the left of the banner, there is a text block: "Create and manage your projects and solutions from one convenient location." To the right of the banner is a central illustration featuring a white robot head, a blue smartphone-like device with a pie chart and line graph, a white coffee maker, and two blue circular arrows. A magnifying glass is focused on a document icon. Below the banner, the text "Start building" is displayed. There are two main sections: "Case solutions" (represented by a folder icon) and "Process apps" (represented by a network icon). Both sections have "View all" and "Create +" buttons at the bottom. The "Case solutions" section also includes a brief description: "Enable content-based activities and unify information, processes, and roles around a persisted case object model."

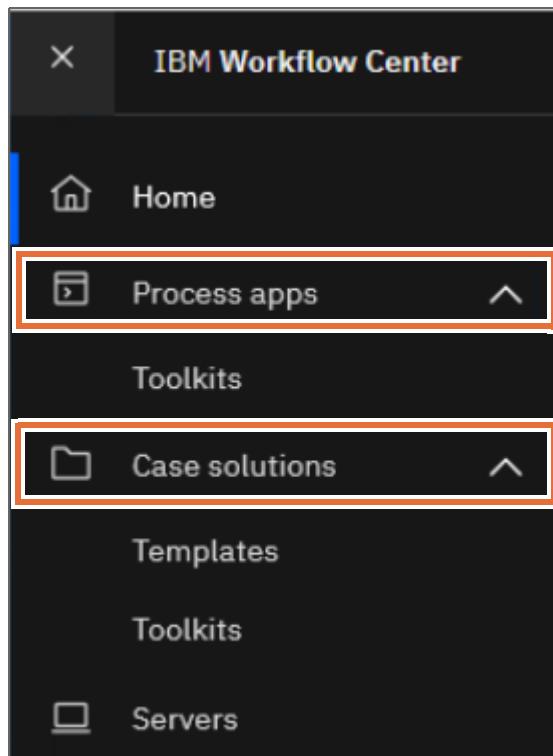
___ 2. Review and set preferences.

When initially logging in to Workflow Center, it defaults to displaying the Basic configuration.

- ___ a. To display the functions associated with the Basic configuration, click the navigation icon in the upper left.



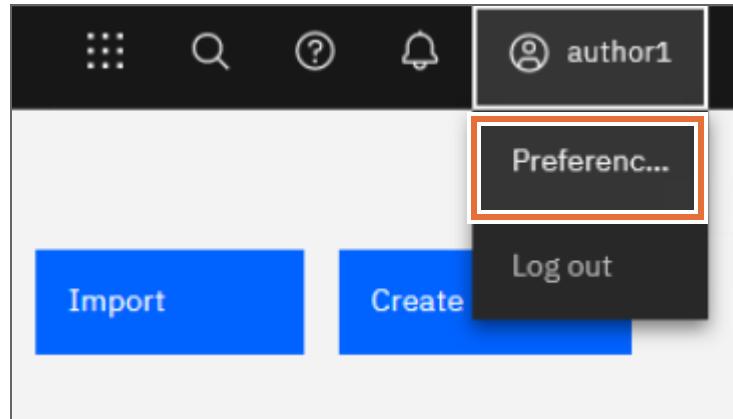
- ___ b. Expand the **Process apps** and **Case solutions** sections.



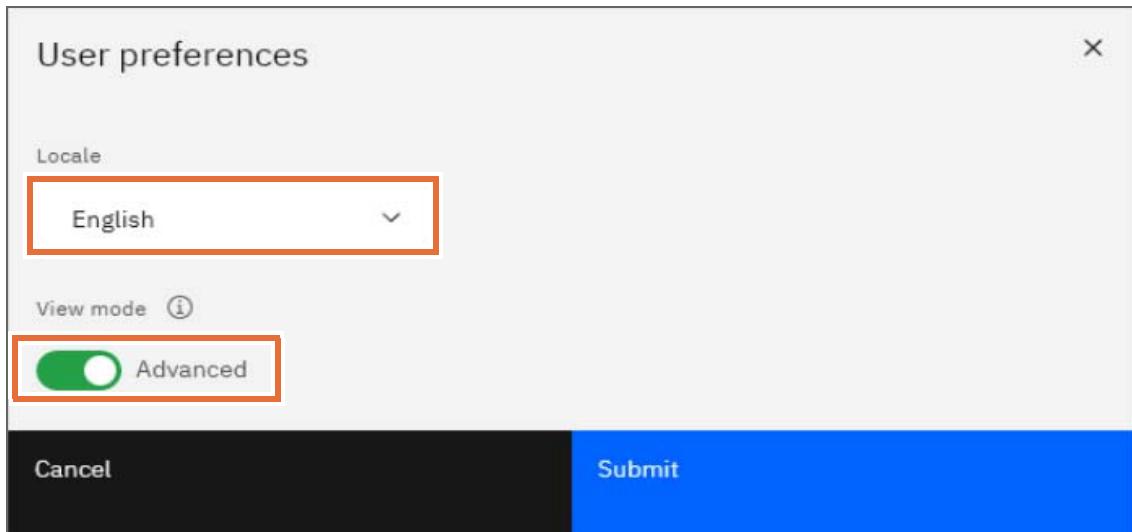
Notice that you can access Process apps, Toolkits, Case solutions and Case Templates. You can also create servers by using the Server tab.

- ___ c. Click the X in the upper left corner to close the navigation bar.

- ___ d. Open the Preferences settings by clicking **author1 > Preferences**

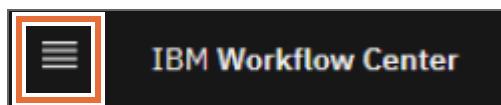


- ___ e. In the User Preferences dialog box, change the **Locale** to your language and the **View mode** to **Advanced**.

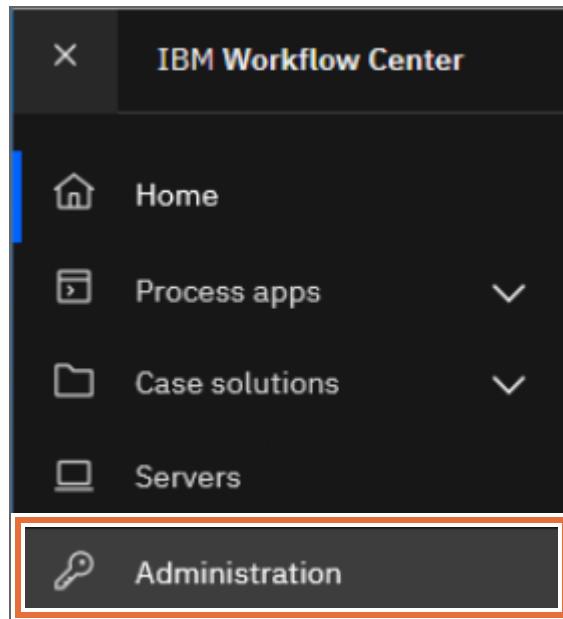


- ___ f. Click **Submit**

- ___ g. Click the navigation icon in the upper left.



__ h. Select Administration



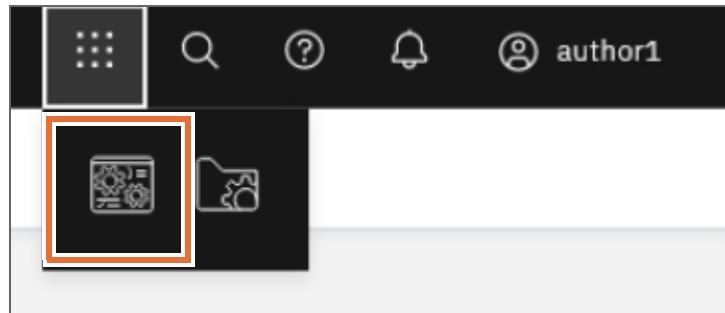
After setting preferences to Advanced, you can now see the Administration option.

You are presented with the Permissions page displaying the default groups for Workflow Center. You can access this page because author1 is a member of the tw_authors group. From this page, you can add users and groups.

Permissions		Resource registry			
Manage access to your repository ⓘ Change access to a project area in the Case administration client . <input type="button" value="Add users"/> <input type="button" value="Add groups"/>					
User or group		Read	Write	Admin	Remove
	tw_admins (tw_admins)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value=""/>
	tw_authors (tw_authors)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value=""/>

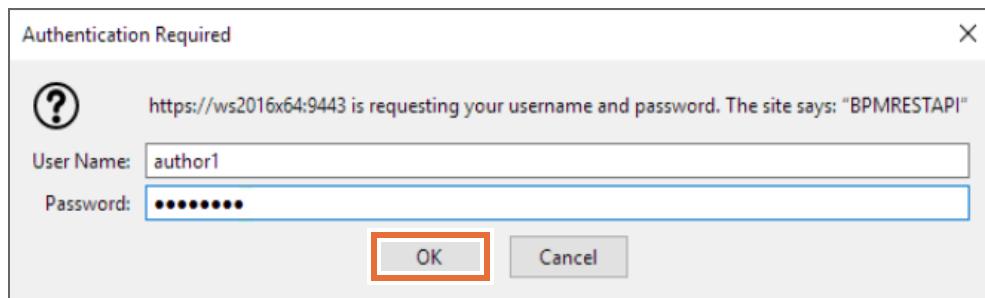
- __ i. Click the grid in the menu bar

In case you need to access the Classic Workflow Center, you can click the **Classic Workflow Center** icon



Troubleshooting

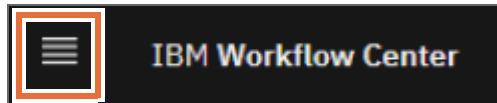
If you leave your workstation for a period of time, BAW logs you out and provides a dialog box to log back in. If this happens, enter your credentials and press **OK**.



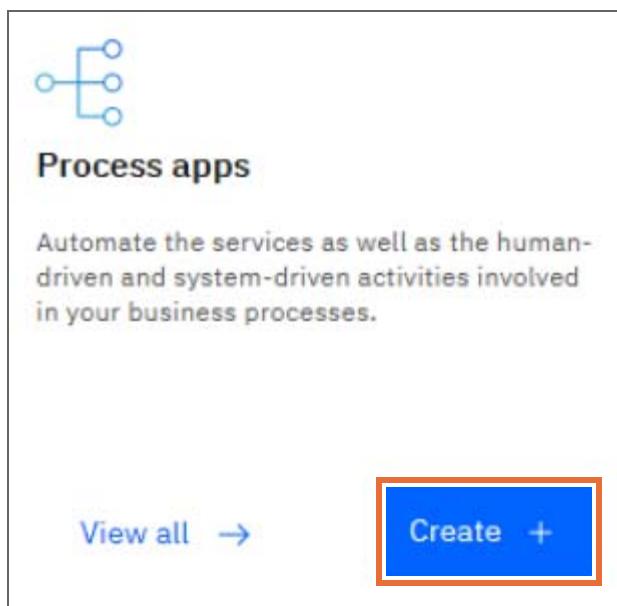
Part 3: Create and manage a process application

In this course, you build a Hiring Request Process application. You create that process application in this section. In subsequent lab exercises, you build all the required artifacts, specifying more details as you progress.

- ___ 1. Create a process application.
 - ___ a. Click the navigation icon in the upper left.



- ___ b. Select **Home**.
- ___ c. Click **Create** in the Process apps tile.



Hint

While completing the form, be sure not to click outside of it. Once the focus is taken away, the form is withdrawn. If this happens, click **Create** again to initiate the dialog.

- ___ d. When the **Create a process app** dialog box is displayed, enter the following values:
 - **Name:** HR Recruitment Process
 - **Acronym:** HRR
 - De-select (clear) the **Open in Process Designer** option
 - **Target Environment:** Traditional

- **Description:** Contains HR processes that are used for recruitment.

Create a process app X

Name

Acronym ⓘ

Open in Designer

Target environment

Description
Highlight your text to see formatting options.

Cancel Create

— e. Click **Create**.



Troubleshooting

If you do not see the Acronym text box, be sure you correctly set the View Mode for the author1 username to **Advanced**.

- __ f. Verify that the process application is created and is now in your list of process apps.

The screenshot shows the IBM Workflow Center interface with the title "Process apps (8)" at the top. Below the title, there is a search bar and two dropdown menus: "Recently updated" and "All". There are also "Import" and "Create" buttons. A message indicates "Displaying 8/8 projects". Three tiles are listed: "HR Recruitment Process" (highlighted with an orange border), "Performance", and "Process Portal". Each tile has a star icon, a more options icon, and a "Details →" button. The "HR Recruitment Process" tile contains the subtext "Contains HR processes that are used for recruitment." and a small icon of a database.

Notice the **Traditional** icon is displayed denoting a Traditional (on-premises) implementation.

- __ 2. Manage the HR Recruitment Process process application.
__ a. To view the details of the **HR Recruitment Process** process application, click **Details** on the tile.

- ___ b. By default, the **Overview** tab is displayed. Other tabs that are displayed include Snapshots, Permissions, History, and Comments. You can create snapshots in Workflow Center or in IBM Process Designer. From this page, you can also change permissions, review the history of the application, and provide comments.

The screenshot shows the IBM Workflow Center interface. At the top, there's a navigation bar with icons for search, help, and notifications, and a user account labeled "author1". Below the bar, the title "HR Recruitment Process" is displayed, along with a "Back to process apps /" link and a "Open in Designer" button. A horizontal navigation bar below the title has tabs for "Overview", "Snapshots", "Permissions", "History", and "Comments", with "Overview" being the active tab. The main content area shows a summary card for the process. On the left, there's a small diagram icon. To its right, the acronym "HRR" is listed under "Acronym". Below that, "Target environment" is shown as "Traditional". On the right side of the card, there are "Created" and "Last updated" details: "author1" on 9/17/2020 at 2:59:10 PM, and "author1" on 9/21/2020 at 6:02:27 PM. There are also a star icon and a more options menu. Below the card, there's a checkbox for "Let users create branches." and a "Description" section containing the text "Contains HR processes that are used for recruitment." with an edit icon.

- ___ c. Click the **Comments** tab and enter a comment.
___ d. Click **Add comment**.

The screenshot shows a "New Comment" dialog box. At the top, it says "New Comment" and "Highlight your text to see formatting options.". Below that is a large text area containing the text "New process application created today.". At the bottom of the dialog, there are three buttons: "Clear", "Add comment", and a plus sign (+) button. The "Add comment" button is highlighted with a red box.

- The Comments entry is added to the left with a date and time stamp.
- ___ e. Review the **History** and **Permissions** tab to become familiar with the information presented. These sections are specific to the process application.

In the next section, you are introduced to tools that are helpful in troubleshooting development issues.

In the final section, you export and import the process application. At the beginning of each exercise, you have the option to import an existing solution file or to continue working with your solution. If you run into issues in your process application, you have the option of importing a working solution to keep progressing in the class.

Part 4: Explore other tools

In addition to working with the Workflow Center console, there are several other tools that one works with when developing applications in IBM Business Automation Workflow. You explore some of those tools in this section. You can use these tools throughout the course to aid in troubleshooting.

- ___ 1. Explore the Administration console. The Administrative console is used to administer applications, services, and other resources at a cell, node, server, or cluster scope. You can use the console with stand-alone servers and with deployment managers that manage all servers in a cell in a networked environment.
- ___ a. To access the administration console, go back to the minimized Quick Start window and click the **WebSphere Application Server Administrative Console** link.



The screenshot shows a window titled "IBM Business Automation Workflow Quick Start". Below the title, it says "IBM Business Automation Workflow Enterprise Quick Start for ProcessCenter with profile DmgrProfile". There are two main sections: "Deployment environment administration" and "Administration consoles & tools". Under "Deployment environment administration", there is a link "Stop the Workflow Center Deployment Environment. Stop all configured clusters." and "Stop the Workflow Center Deployment Environment". Under "Administration consoles & tools", there is a link "WebSphere Application Server Administrative Console" which is highlighted with a red rectangle. Below that, there is another link "Administer the workflow servers in your environment, including the users and installed snapshots".

- ___ b. If prompted, log in by entering author1 in the **User ID** field and passw0rd in the **Password** field. Click **Log In**.



Troubleshooting

If you receive a message stating another user is currently logged in with the same user ID, leave the default selection (to log out the other user) and click **OK**.

- ___ c. Expand the Servers panel and click **Deployment Environments**. On this screen, you can start, stop, and create new deployment environments. You can also verify the status of a deployment environment.

Select	Status	Deployment Environment Name	Features	Pattern	Description
<input type="checkbox"/>		ProcessCenter	IBM Business Automation Workflow Advanced Workflow Center	Single Cluster	

- ___ d. Expand the **Resources** panel, then expand the **JDBC** section and click **Data sources**.
 ___ e. Scroll down to the bottom and click **Training Course DB**. This is the JDBC driver that is used later in the course to access a DB2 database.

<input type="checkbox"/>	Training Course DB	jdbc/TrainingDB	Cluster=SingleCluster	DB2 Using IBM JCC Driver
--------------------------	------------------------------------	-----------------	-----------------------	--------------------------

- ___ f. Scroll down to the bottom of the page and the data source properties are displayed.

Common and required data source properties	
Name	Value
* Driver type	<input type="button" value="4"/>
* Database name	Training
* Server name	ws2016x64
* Port number	50000

- ___ g. While logged in, feel free to explore the administrative console further. Optionally, click **Logout** to log out of the console.



- ___ 2. Explore the Process Admin console. The Process Admin Console is used to administer the process servers in your environment, including the users and installed snapshots for each server. In addition, it provides tools to help you manage queues and caches.
- ___ a. To access the administration console, go back to the Quick Start window and click the **Process Admin Console** link.

IBM Business Automation Workflow Quick Start

IBM Business Automation Workflow Enterprise Quick ProcessCenter with profile DmgrProfile

Deployment environment administration

Stop the Workflow Center Deployment Environment. Stop all configured clusters
[Stop the Workflow Center Deployment Environment](#)

Administration consoles & tools

Manage applications, buses, servers and resources within the administrative domain. Control managed nodes and their resources.
[WebSphere Application Server Administrative Console](#)

Administer the workflow servers in your environment, including the users and instances for each server. Manage queues and caches, and view and manage process instances.
[Process Admin Console](#)

Work with Performance Data Warehouse queues, manage data transfer errors and overall performance.

- ___ b. If you logged out of the Administrative console, then you are prompted to log in to the Process Admin Console. Enter author1 in the **User Name** field and passw0rd in the **Password** field. Click **Log In**.

- ___ c. When the Process Admin console is displayed, note the various links in the left pane.

The screenshot shows the IBM Process Admin Console interface. The left sidebar contains a tree view with the following items:

- ▶ Workflow Admin
- ▶ User Management
- ▶ Monitoring
- ▶ Event Manager
- ▶ Admin Tools
- ▶ Alert Definitions
- ▶ Performance

The item "User Management" is highlighted with a red box. The main content area on the right displays the "Process Status Summary" with four status indicators: Active (green), Completed (blue), Failed (red), and Suspended (orange). Below this, there is a section titled "Process Applications" listing several options: Heritage Process Portal (deprecated) (TWP), Process Portal (SYSRP), Performance (SYSPERFDB), and Hiring Sample (HSS).

- ___ d. Expand **User Management** and then click **User Management**.

The screenshot shows the same IBM Process Admin Console interface, but now the "User Management" link under the expanded "User Management" category in the left sidebar is highlighted with a blue box. The other items in the sidebar remain the same as in the previous screenshot.

- ___ e. In the **Retrieve Profile** field, enter * and click **Retrieve** to view a list of users.

- ___ f. Click **Group Management** in the left menu.
 ___ g. In the **Select Group to Modify** field, enter %%
 ___ h. Examine the groups listed. Do not make any changes as it impacts the exercises.

- ___ i. As time allows, continue to explore the Process Admin Console.
 3. Explore the Process Inspector.
 Use the Process Inspector to view the task, activity, and data details of your active process instances.
 ___ a. Click the **Process Inspector** link in the navigation bar at the top.

The Process Inspector opens in a new tab. If you run into issues in your development, you can use the Process Inspector to view the status of all running process applications.

___ b. Click **Search**

The **Filter process instances by** section is updated and all process applications in the repository are listed on the right with their status. The Search button changes to Refresh.

___ c. Click the **Standard Employee Requisition for Tom Miller** instance.

___ d. On the right, expand **Tasks and Data**.

Showing 2 of 2 Instances
Sort by: Date of last action

Advanced Employee Requisition NG (List) for Tom Miller (14) Advanced HR Open New Position
Last modified Feb 13, 2019 Due Feb 15, 2019

Standard Employee Requisition for Tom Miller (Standard HR Open New Position) Standard HR Open New Position
Last modified Feb 13, 2019 Due Feb 13, 2019

Instance ID: 11
Status: Completed
Start time: Feb 13, 2019 5:15 PM (1 year ago)
Last action: Feb 13, 2019 11:53 PM (1 year ago)
Due date: Feb 13, 2019 9:15 PM (1 year ago)

Actions
Refresh
Edit

Tasks (2) (Active | Completed | All)
 REVIEW NEW POSITION REQUEST [2]
The task was closed by author1 (author1).
 SUBMIT POSITION REQUEST [1]
The task was closed by author1 (author1).

Data
 requisition(Requisition)
 currentPosition(Position)
 candidates(Candidate)(List) Empty

As you develop your solution, you can use the Process Inspector to inspect the status of active, completed, suspended, terminated, and failed process applications. You can also view the task last executed and the data that is associated with the task.

If you expect to see a task or process display in Process Portal and it does not, you can use the Process Inspector to determine what state the process application is in.

Part 5: Export the process application

Now that this exercise is complete, you are ready to export the process application. This export saves your application as a `.twx` file in a folder for future use. If you wish to keep your solution at the end of the course, you can export it following this procedure and upload it to a web-based file sharing program such as Box.

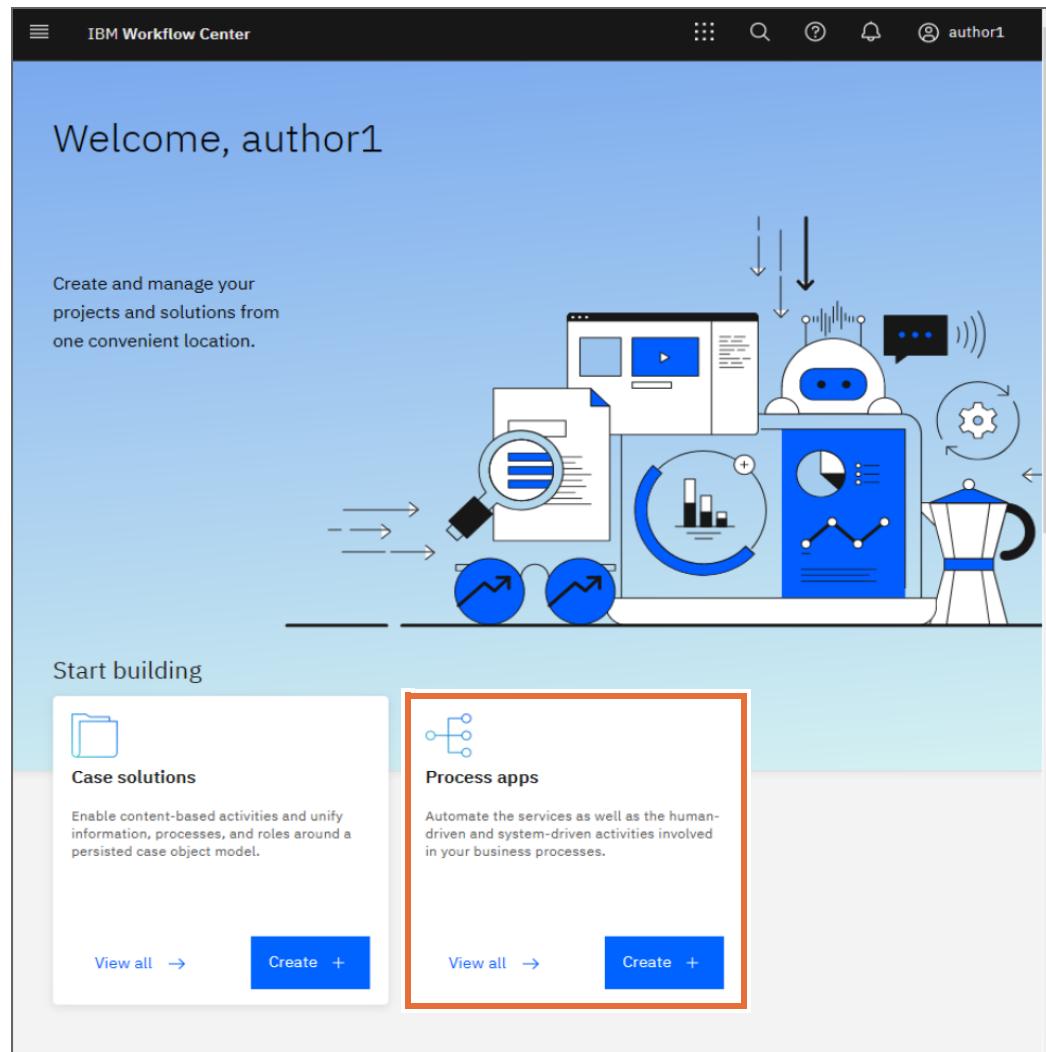
Each exercise in this course builds on the prior one. The progression resembles a real-world implementation of the IBM playback methodology.

There are working solutions for each exercise listed in [Appendix A. Solutions](#). If you run into issues in an exercise, you can export your solution and import the working solution to perform troubleshooting. If you are unable to determine the issue, you can import the exercise solution for the current exercise to continue working. This process helps you to continue working with exercises without any interruption from an unresolved error in your previous exercise.

1. Create a snapshot

Only process applications with snapshots can be exported.

a. Return to the Workflow Center and click the Process apps tile.



- __ b. Click **Details** on the HR Recruitment Process tile.

IBM Workflow Center

Process apps (8)

Recently updated ↓ All

Import Create +

Displaying 8/8 projects

HR Recruitment Process Contains HR processes that are used for recruitment. Details →	Performance Details →	Process Portal Process Portal Details →
--	---	--

- __ c. Select the **Snapshots** tab.
- __ d. Open the list of Options by clicking the three dots on right side of the **Current** tip.
The tip is the current working version of a project.
- __ e. Select **Create a snapshot**

Back to process apps /

HR Recruitment Process

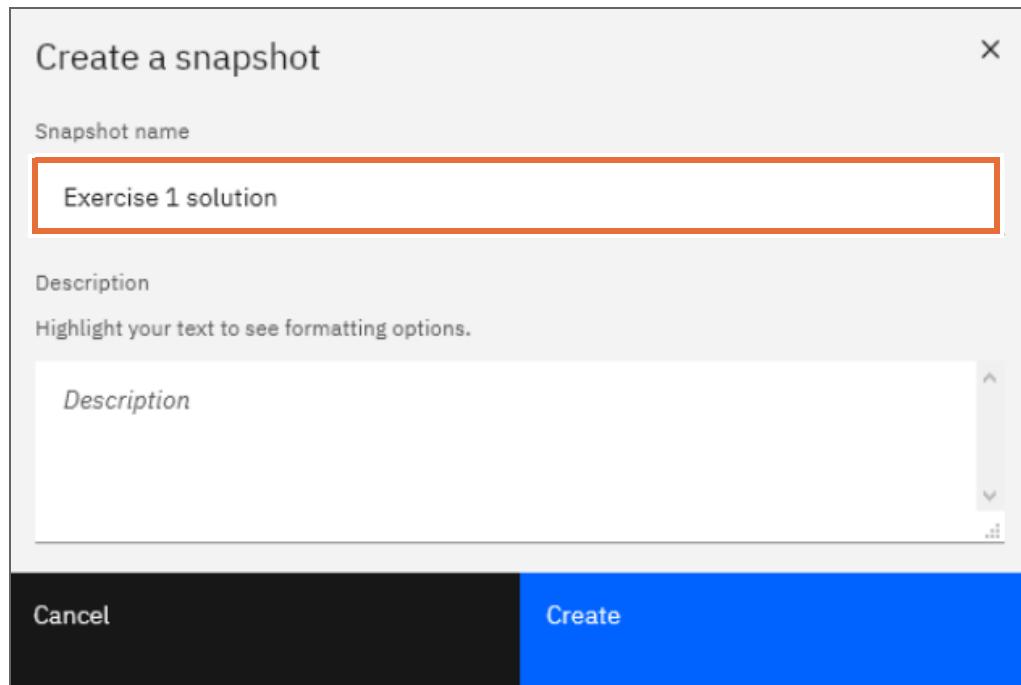
Open in Designer

Name	Status	Created	Last updated	Target environment
→ Current	author1	9/22/2020, 12:51:50 PM	author1 9/22/2020, 12:51:50 PM	Traditional

[Create a snapshot](#)

[Open in Designer](#)

- ___ f. Enter `Exercise 1 solution` for the name.



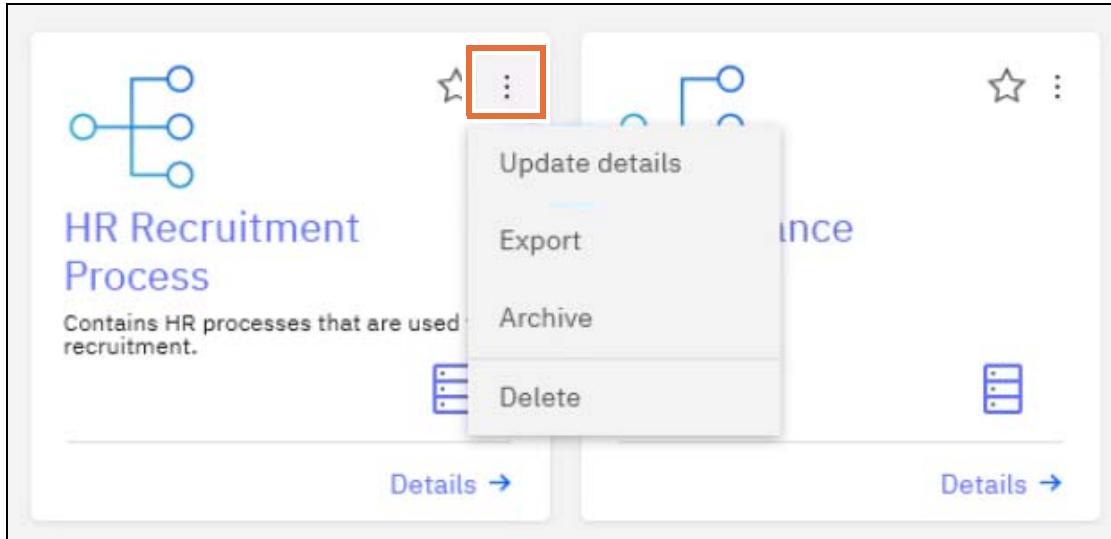
- ___ g. Click **Create**.

- ___ 2. Export the application. Exporting the application also exports the toolkit that it depends on. You can only export snapshots. Since there are no associated toolkits yet, only the process application is exported.
- ___ a. Open the list of options for the snapshot and select **Export**

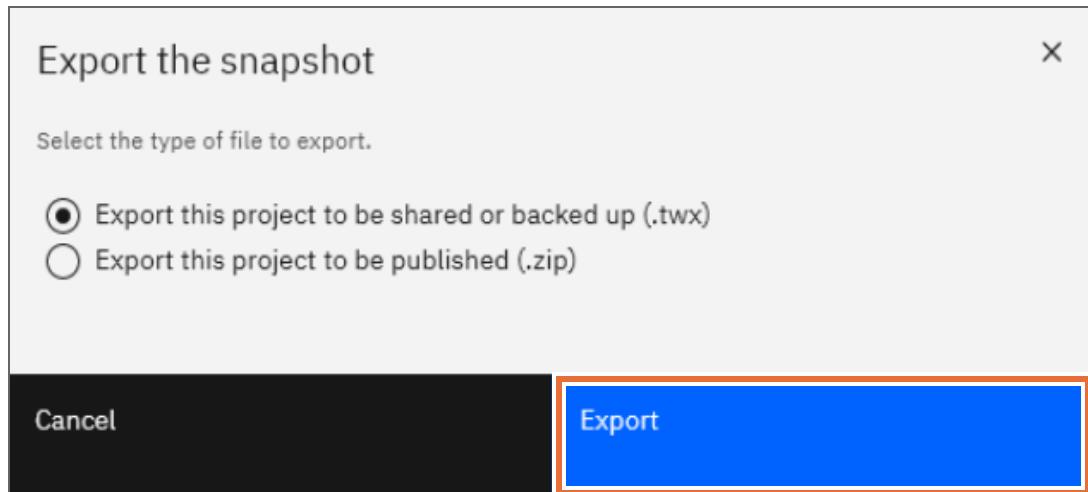
Name	Status	Created	Last updated	Target environment
→ Current	author1 9/22/2020, 12:51:50 PM	author1 9/22/2020, 12:51:50 PM		Traditional
→ Exercise 1 solution (E1S)	New	author1 9/22/2020, 3:35:42 PM	author1 9/22/2020, 3:35:42 PM	Traditional

**Hint**

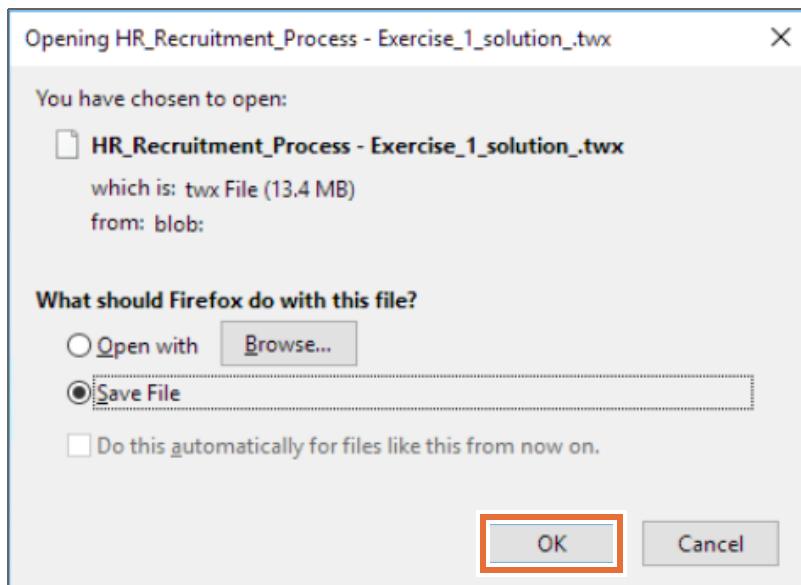
You can also export, archive, and delete process applications directly from the **Process apps** page by accessing the list of options on the tile.



- ___ b. In the **Export the snapshot** window, click **Export**. It may take a minute or two to generate the **.twx** file.



- ___ c. Click **OK** to save the file. The application is saved to the default folder, which is the Downloads directory.



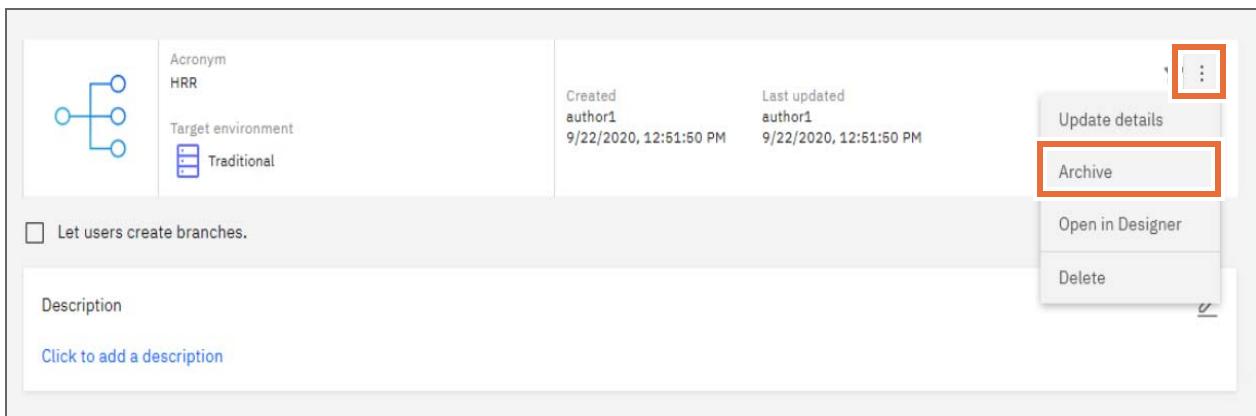
- ___ d. Verify that the application is listed in the downloads directory.
- ___ 3. Archive the process application and delete it. Although you don't have to archive an application before deleting it, it is a good practice. Archiving an application also archives the toolkit that it depends on. Since there are no associated toolkits yet, only the process application is archived.



Important

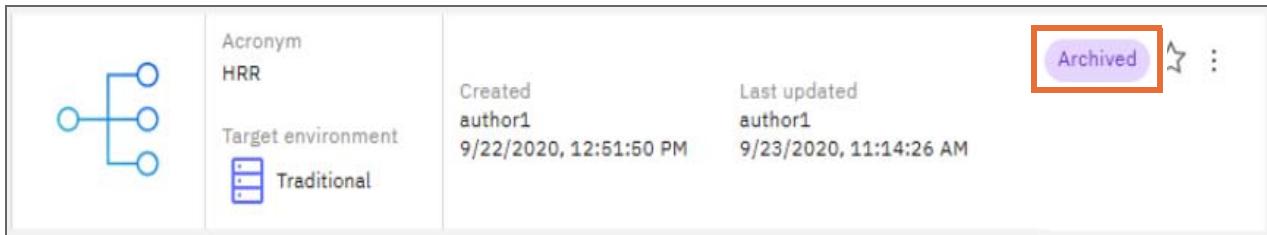
Don't archive a toolkit that other projects depend on nor use archived toolkits when creating toolkit dependencies in Process Designer. Archived items are not part of the active library.

- ___ a. Return to the HR Recruitment Process application in the Workflow Center and select the **Overview** tab.
- ___ b. Open the list of options for the application and select **Archive**.

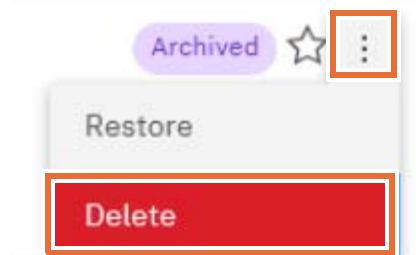


- ___ c. In the window that opens, click **Archive** to confirm the action. Clicking **Archive** does not delete the application; it removes it from the available **Process Apps** and makes it ready for deletion.

An **Archived** tag appears in the application Overview tile.



- ___ d. Open the list of options for the application and select **Delete**.



- ___ e. In the window that opens, click **Delete** to confirm the deletion. The process application is now deleted.

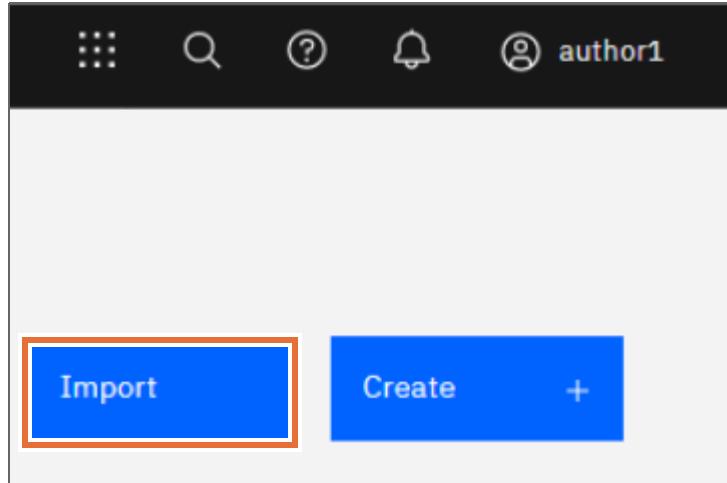
You are returned to the list of Process apps.

- ___ f. Verify that **HR Recruitment Process** is no longer listed.

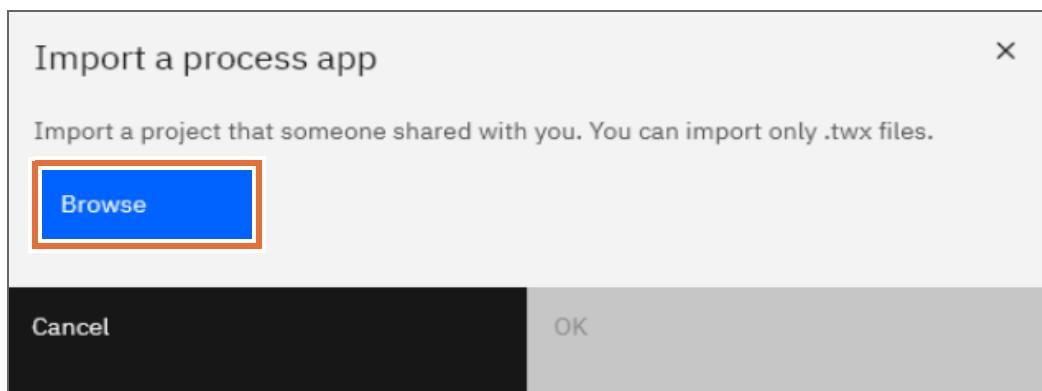
Part 6: Import the process application

In this section, you import the process application solution that you exported in the previous section. If you perform this import procedure in the beginning of an exercise by using the solution for the prior exercise, this ensures that you have a good copy of the process application at the start of the exercise. If you run into issues in your exercise, you can import the working solution. This process helps you to continue working with exercises without any interruption from an unresolved error in your previous exercise.

- __ 1. Import the process application.
- __ a. On the list of Process apps, click **Import**.

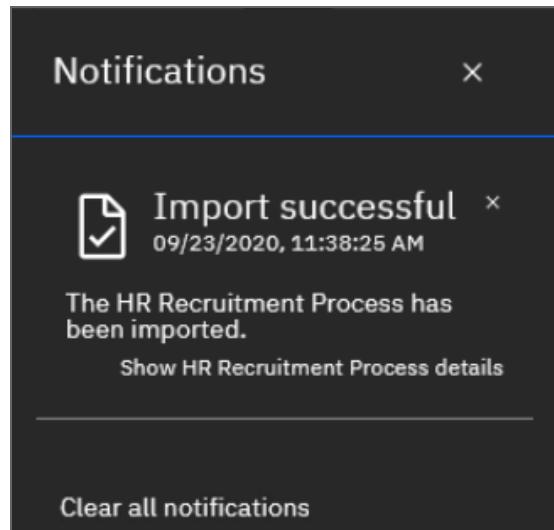


- __ b. Click **Browse**.



- __ c. Navigate to the **Downloads** folder and select the file:
HR_Recruitment_Process - Exercise_1_solution.twx
- __ d. Click **Open**.
- __ e. Click **OK**.

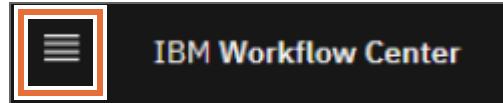
- ___ f. The Notifications window displays the message



- ___ g. Close the **Notifications** window by clicking the **X** in the upper right corner.
___ h. Verify that the **HR Recruitment Process** application is listed.
___ 2. Let users update the toolkit.

This step is only pertinent after Exercise 7 when you start using toolkits. When importing a toolkit, it does not default to let others use it.

- ___ a. Click the navigation icon in the upper left.



- ___ b. Select **Toolkits**.

- __ c. Click **Details** on the Hiring Requisition Toolkit.

Toolkits (10)

Recently updated ↓ All

Import Create +

Displaying 10/10 projects

Hiring Requisition Toolkit This toolkit contains all common artifacts that are associated with the HR Recruitment Processes. Details →	UI Toolkit Details →	SAP Guided Workflow (deprecated) Toolkit of SAP Guided Workflow Details →
---	--	--

- __ d. Verify the box is checked next to **Let users update the toolkit**.

Back to toolkits /

Hiring Requisition Toolkit

Open in Designer ↗

Overview	Snapshots	Permissions	History	Comments
 Acronym HRT Target environment 	Created author1 11/3/2020, 10:47:36 AM	Last updated author1 11/4/2020, 4:34:55 PM	<input checked="" type="checkbox"/> Let users update the toolkit.	

You have completed this exercise.

You are now ready to start building your process application. Each exercise shows step-by-step instructions that you can follow to complete the tasks. In IBM Business Automation Workflow, you find many different ways to complete modeling tasks. The step-by-step instructions show one way to accomplish these tasks in the exercises.

End of exercise

Exercise 2. Playback 0: Creating the To-Be process

Estimated time

02:00

Overview

This exercise covers how to create a process by using IBM Process Designer. The Hiring Requisition process owner provided detailed information about the process and its current state to the BPM analyst, who in turn documented the information. This step completed the process discovery and initial analysis, and now the process model can be created. To accomplish the task of creating the initial process model, you model it with a pool, lanes, and flow objects such as activities, and events, and nested processes. You take the information that is provided for the use case and translate that into a process. Your first task is to create a process and name it according to naming conventions. You then add the activities in the appropriate lanes and use sequence flow to connect the activities. Be sure to model the happy path (critical path) first. You also complete decomposition on your process and create a linked process where you see the opportunity.

Objectives

After completing this exercise, you should be able to:

- Translate business process workflow steps that are documented in the process discovery and analysis into process model tasks
- Create the foundation for a process by adding the appropriate lanes to the default pool
- Model the expected process flow for the initial process model
- Decompose business process workflow steps that are documented in the process discovery and analysis into process model tasks
- Create a linked process

Introduction

In the previous exercise, you created a process application. In this exercise, you model the process to be used for the application.

In this exercise, you create a to-be process model. To accomplish the task of creating the initial process model, you define a pool, lanes, and flow objects such as activities, events, and a linked process.

In this exercise, you add activities to the appropriate lanes and use sequence flow to connect the activities. As mentioned in the unit, you model the happy path first. Gateways and various flows are presented in the next exercise.

Finally, you create a subprocess and link it to the main process.

This exercise is long, so a break is included in the middle of the instructions.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Create a process](#)

[Part 2: Create process activities](#)

[Part 3: Model the process teams](#)

[Part 4: Assign the activities to the appropriate Team](#)

[Part 5: Create a linked process](#)

[Part 6: Attach the linked process](#)

Exercise instructions

Hiring Requisition process

The company expanded from the simple (as-is) hiring process to a more detailed (to-be) hiring process. The detailed process now includes more information. The process that you are going to examine and model is called the Hiring Requisition Process. The company also wants to reduce rework by introducing a more structured process for the Hiring Requisition process. This process covers a new job position through submission, approval, and completion so applicants can apply for the job position.

Core requirements

- 1.1: A Hiring Manager submits a hiring requisition to the HR Department. The request contains the following information:
 - Customer details:
 - Requisition number
 - Date of request
 - Requester
 - Date position available
 - Job title
 - Job description
 - Job level
 - Number of people managed
 - Division
 - Department
 - Salary to offer
 - Bonus amount
 - Hiring Manager comments
 - New position
- 2.1: If the answer to “New position” is Yes, the request is forwarded to a General Manager. After the General Manager receives the request, the General Manager indicates approval or disapproval.
- 2.2: If the request is not approved, the General Manager specifies a reason and the request is closed. If the request is approved, a salary compliance check is conducted.
- 2.3: The Hiring Manager is notified of the General Manager’s decision after the General Manager approval step.
- 2.4: After the requisition is submitted, an automated system checks for salary compliance. If the request meets salary compliance, the hiring request is automatically posted to the HR Positions database and made available for dissemination.

- 2.5: When a request violates the established salary guidelines of the company, the HR Administrator can approve or reject the requested salary override.
- 2.6: If the salary override is approved, the request is posted to the HR Positions database and made available for dissemination.
- 2.7: If the HR Administrator rejects the requested salary, the HR Administrator must provide comments for the violation, add a proposed salary, and send the request back to the Hiring Manager who originated the request.
- 2.8: When the Hiring Manager gets the request back because of a rejection, the Hiring Manager can negotiate an adjusted salary or can cancel the request. If the negotiation is successful, the request is resubmitted back to the same HR Administrator.
- 2.9: All hiring requests must be added to the HR Positions database regardless of the disposition at the end of the process during a finalization activity.
- 2.10: The HR Administrator has 4 hours to complete the review. If the review is not completed within 4 hours, an email is sent to the HR Administrator. The email notifies the HR Administrator of the missed deadline.

The process owner provides detailed information about the process and its current state to the IBM Process Analyst, who in turn documents the information and analyzes the process for improvement. The process discovery and initial analysis are already completed for you to proceed with the creation of the to-be process model.

Part 1: Create a process

To add the process model to the HR Recruitment Process application you created in the last exercise, return to the Process Designer window.

A process defines the activities that are needed to resolve a specific business problem. A process also defines who works on these activities and the steps they take to resolve the problem. At run time, a business user works with an instance of a process.

- ___ 1. Create a process.
 - ___ a. From the Workflow Center Home page, click **Process Apps**.

__ b. Click **HR Recruitment Process**

The screenshot shows the 'Process apps (8)' section of the IBM Workflow Center. There are three items listed:

- HR Recruitment Process**: This item is highlighted with a red box. It has a blue icon with three nodes connected by lines. Below the icon, the text 'Used for recruitment.' is visible.
- Performance**: This item has a blue icon with three nodes connected by lines. Below the icon, the text 'Performance' is visible.
- Process Portal**: This item has a blue icon with three nodes connected by lines. Below the icon, the text 'Process Portal' is visible.

At the top right, there are buttons for 'Import', 'Create', and a '+' sign. Below the search bar, it says 'Displaying 8/8 projects'.

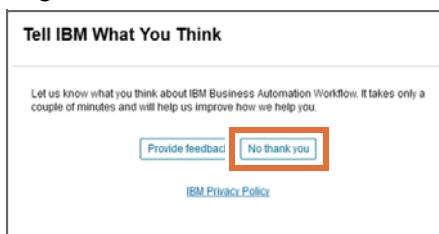
The process application is opened in Process Designer.

__ c. From the process library on the left, click the **(+)** plus sign next to Processes and select **Process** from the list.

The screenshot shows the 'Process App Settings' page for 'HR Recruitment Processes'. The 'Processes' section is selected, indicated by a blue background. A blue box highlights the '+ New' button. Another blue box highlights the 'Process' option in the list below, which is also highlighted with a red box.

i **Information**

If you see a pop up window asking for feedback, click **No thank you**.



**Note**

When developing web-based user interfaces, it is important to keep in mind the different ways in which different browsers render JavaScript. The screen that you see might vary slightly from the screen captures depending on how the JavaScript is being rendered in the browser. For instance, in the Chrome browser, it places a box around the highlighted tab.



Although the development environment might vary slightly, the most important thing to keep in mind is how the user interface is displayed in the browser to be used with the solution.

- ___ d. Enter Hiring Request Process in the **Name** field. Click **FINISH**.

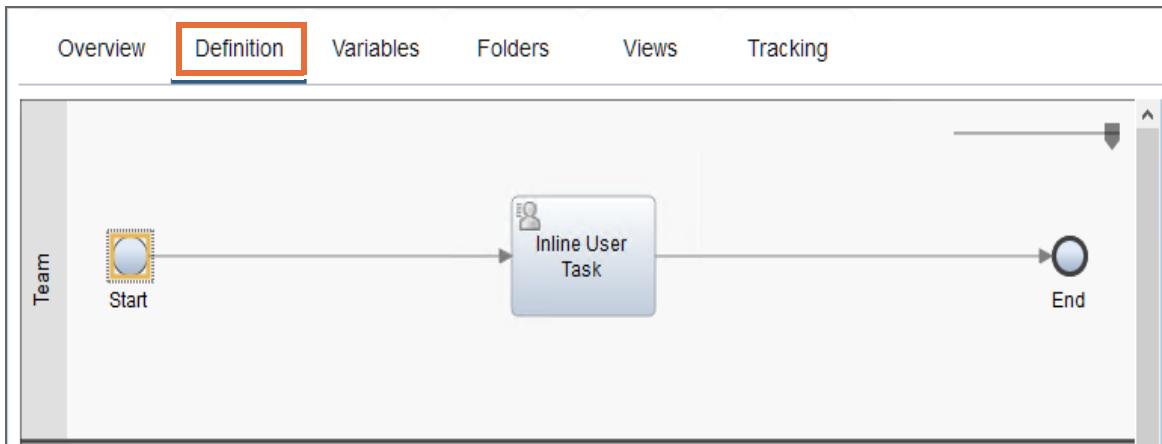
New Process

A process captures a set of activities and the data and content to support the activities. These activities can be part of a structured flow, ad-hoc activities that are not part of a structured flow, or a combination of the two.

Name:

FINISH **CANCEL**

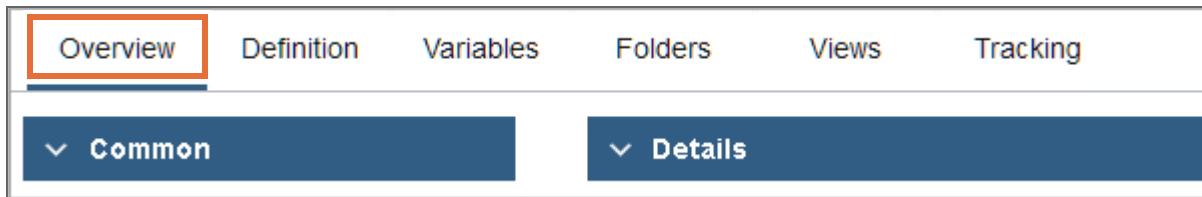
The process is created, and the initial model is shown under the **Definition** tab. By default you have a process with a team lane, a system lane, a start event, an inline user task, and an end event.



- __ 2. Expose the process to **All Users** so that all the users can start the process.

For now, you assign all users to the Hiring Request process. In a later unit and lab exercise in this course, you learn how to define and assign a team and users to your process.

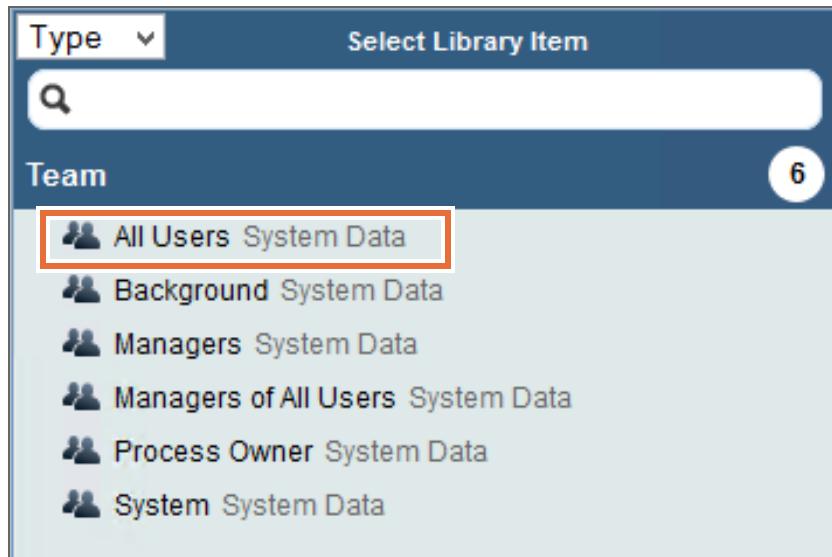
- __ a. Click the **Overview** tab.



- __ b. Under the Exposing section on the right, click **Select** for the **Expose to start** option.



- __ c. Select **All Users**.



- __ d. Click **Finish Editing** to save your changes.



Hint

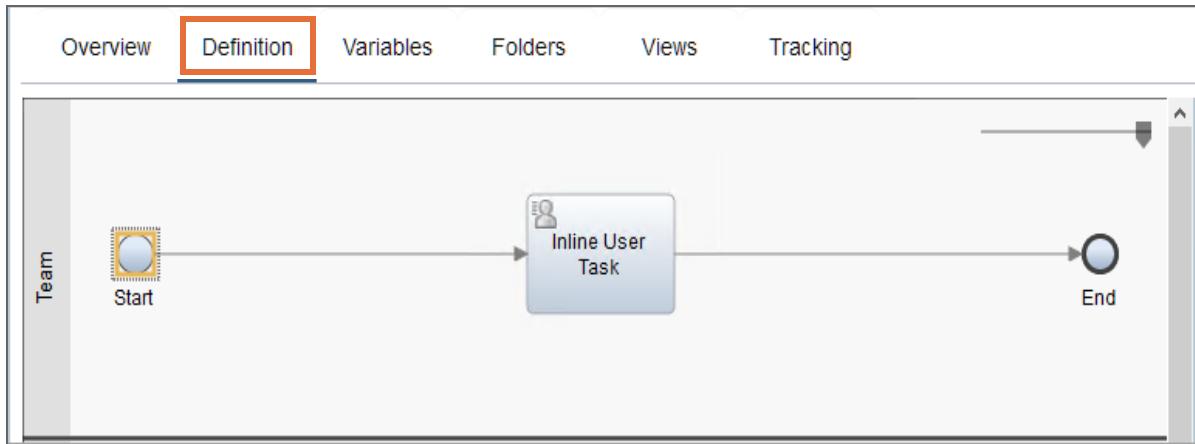
A number appears next to the check mark corresponding to the number of artifacts with unsaved changes. You can also use **Ctrl+S** to save your work.

Part 2: Create process activities

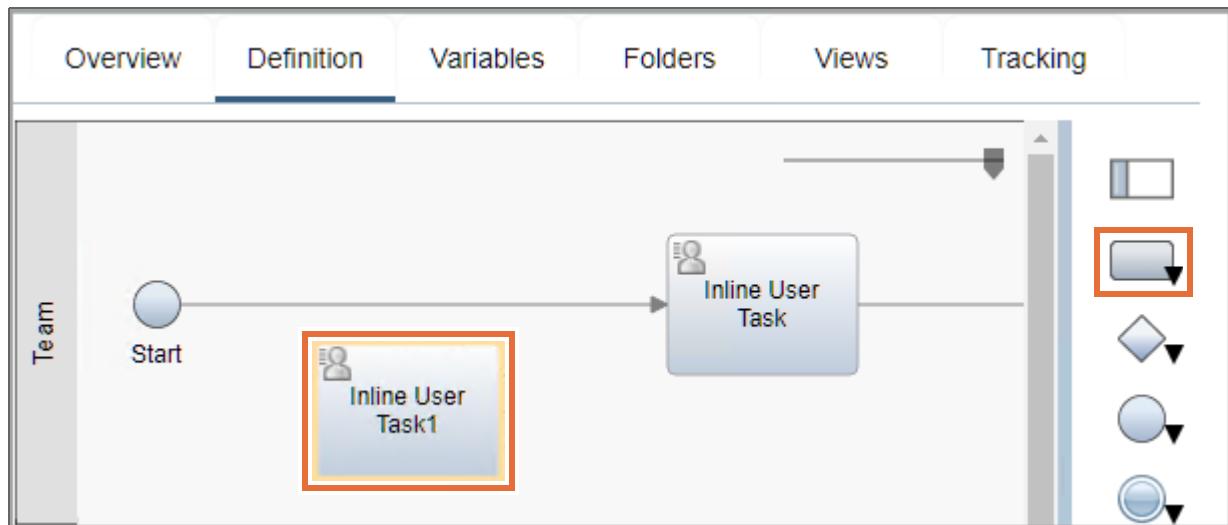
To add activities and events to the process model, use the process requirements that are provided at the beginning of this exercise. Read the main process requirements and determine the activities from the requirements. Because the requirements were written down, it might be easier to write down the activities on paper before proceeding. Remember, activities use a verb-noun naming convention. If you read the requirements carefully, notice that the main process is described from item 2.1 through item 2.4. In the process requirements, four activities are in the main process: Submit Hiring Request, Approve New Hire Request, Approve Hire Request, and Complete Hire Request.

- ___ 1. Create an activity that is named **Submit Hiring Request**.

- ___ a. Click the **Definition** tab to reopen the model.



- ___ b. While holding down the left mouse button, drag an **Activity** from the palette on the right side of the canvas into the Team lane to the right of the start event. When you drag an object to the canvas, the cursor turns into a green plus when you can drop it.
 - ___ c. Release the left mouse button to place the activity on the canvas to the left of Inline User Task.

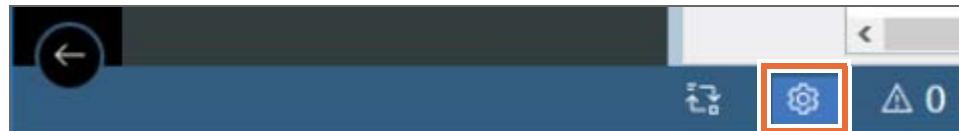


**Hint**

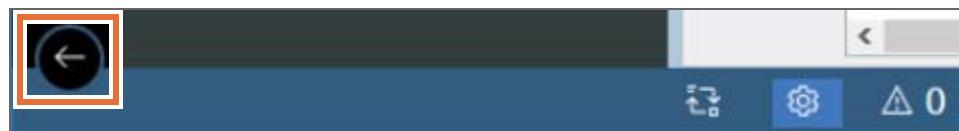
As you model more artifacts, you can increase the size of the canvas by dragging the double lines on the line separating the model definition from the properties section.



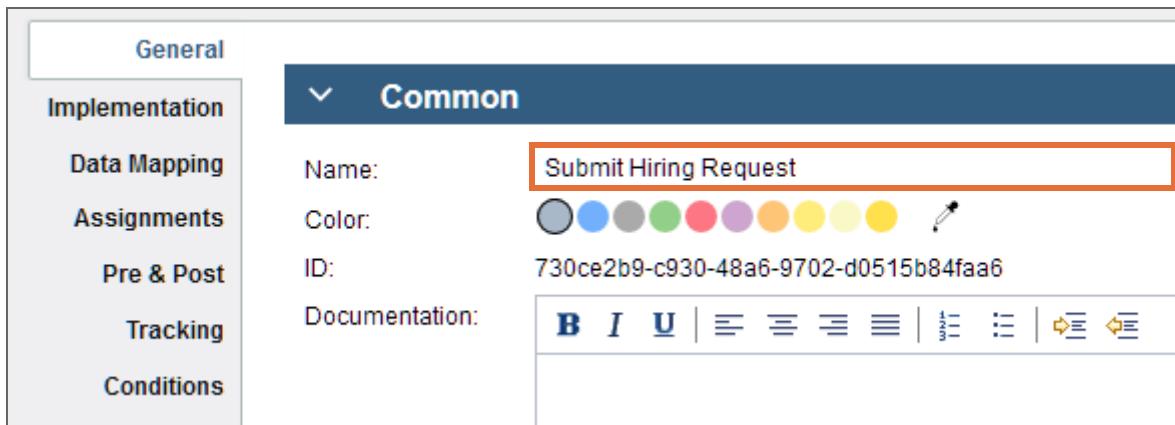
Alternatively, you can toggle the Properties section to display by selecting the Properties icon in the bottom status bar of Process Designer.



To open and close the Process Library, click the arrow at the bottom left.



- ___ d. While **Inline User Task1** is selected, rename it to Submit Hiring Request under the **General > Common** properties section.

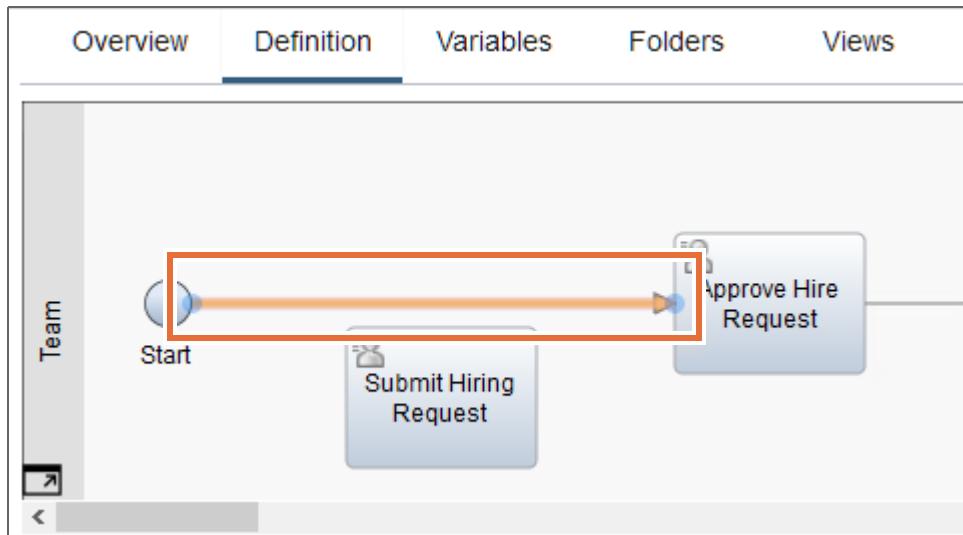


- ___ 2. Create an activity that is named **Approve Hire Request**.
 - ___ a. Click the **Inline User Task** to the right of the Submit Hiring Request activity.

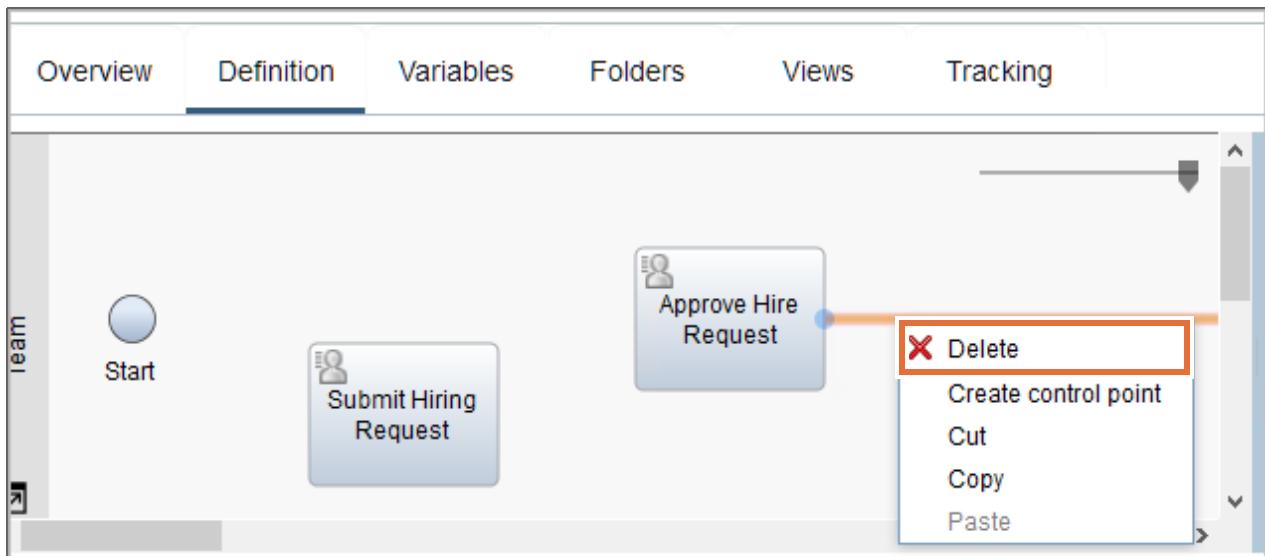
- ___ b. While highlighted, click the text on the activity to rename it as **Approve Hire Request** and press the **Enter** key to apply the name change.



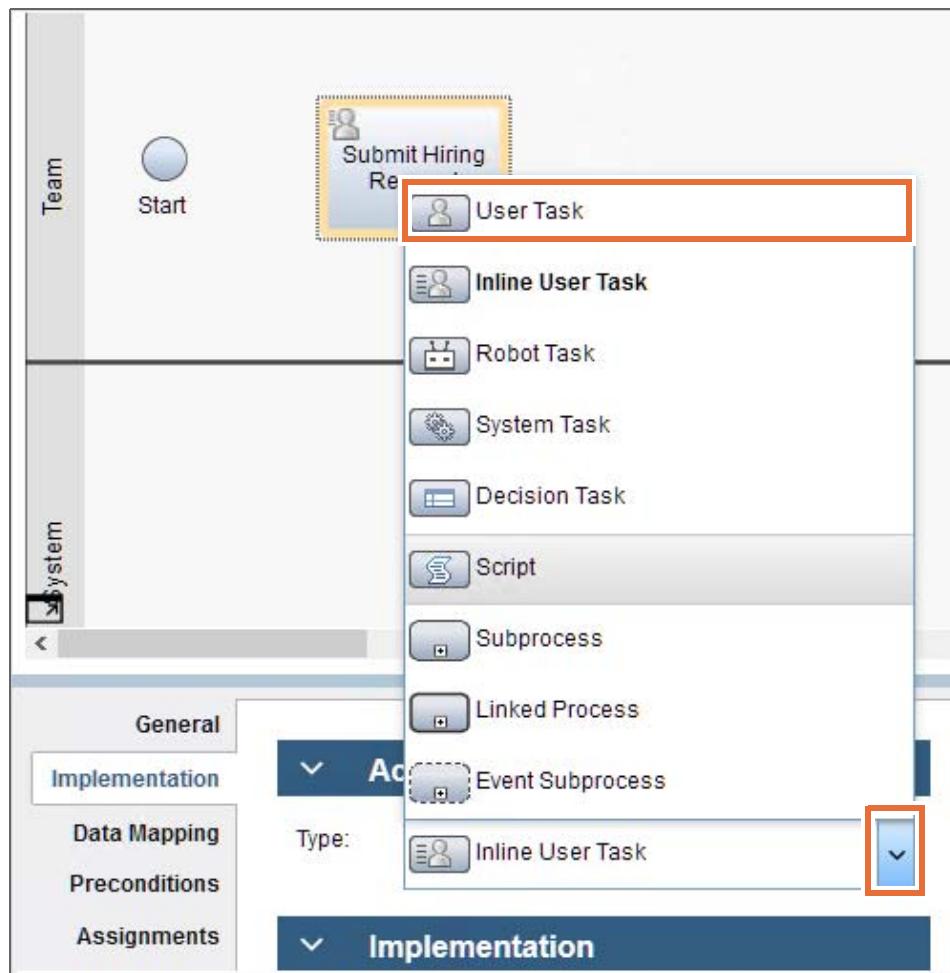
- ___ c. Click the **flow line** between the Start event and the Approve Hire Request activity. Press the **Delete** key on your keyboard to delete the flow line.



- ___ d. Click the **flow line** between the Approve Hire Request and the End Event. This time, right-click the flow line and click **Delete**.



- ___ e. Later in the course, you build custom user interfaces for these tasks. Click the **Submit Hiring Request** activity. In the properties panel at the bottom, go to the **Implementation > Activity Type** menu, expand **Type**, and change the setting to **User Task**.



- ___ f. Under the **Implementation** section, click the **Select** button and select **Default UI Human Service**.
- ___ g. Verify that the **Default UI Human Service** implements your User Task.

The screenshot shows the 'Activity Type' configuration screen. On the left, there's a sidebar with tabs: General, Implementation (which is selected), Data Mapping, Preconditions, Assignments, Pre & Post, and Tracking. The main area has a header 'Activity Type' with a dropdown arrow. Below it, under 'Type:', there's a user icon followed by the text 'User Task'. In the 'Implementation' section, there's a dropdown labeled 'Implementation:' with an orange border around the option 'Default UI Human Service'. To the right of the dropdown are buttons for 'JI Toolkit', 'Select...', and 'New...'. A small blue circular icon with an 'i' is located to the left of the 'Information' heading.

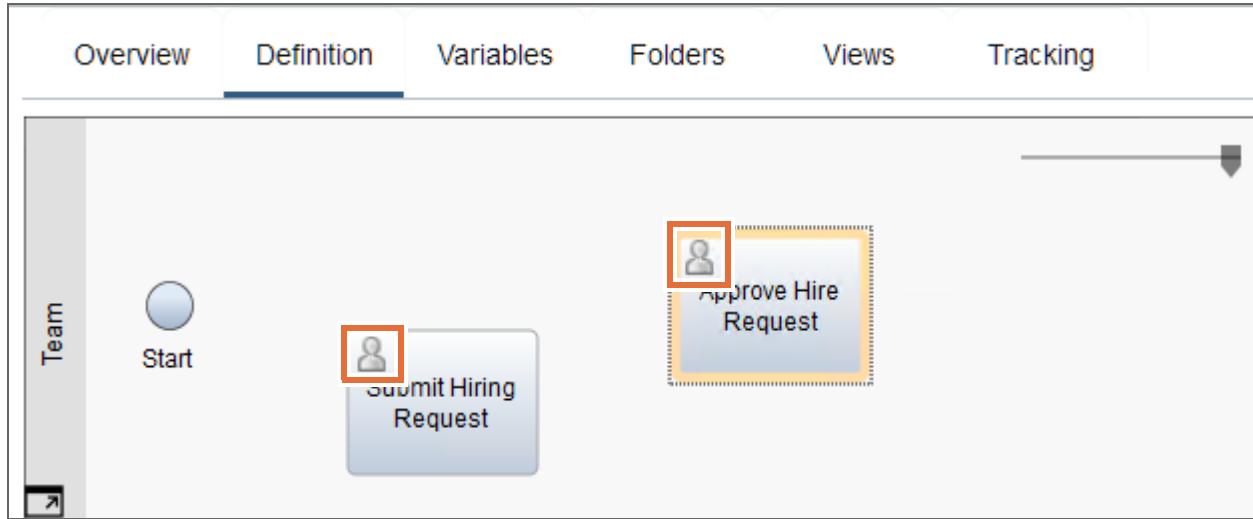
Information

Alternatively, you could continue to use the inline user tasks. An inline user task is an extension of a regular user task and automatically exposes the data that is mapped to the user task. Inline user tasks can be used to run the process immediately after an IBM Blaworks Live import.

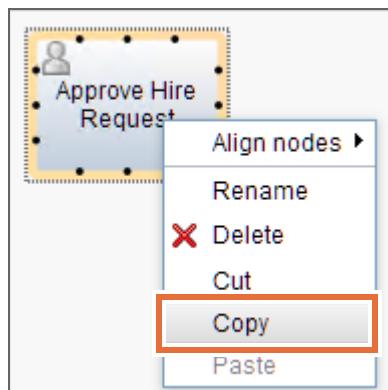
You also have the option to extend the inline user task by creating a stand-alone implementation. For the purposes of this training, you use the default UI human service until you create the user interfaces in a later exercise.

The screenshot shows the 'Activity Type' configuration screen for an inline user task. The sidebar and 'Activity Type' header are identical to the previous screenshot. In the 'Implementation' section, the 'Implementation:' dropdown shows 'Inline User Task' with an orange border around it. Below the dropdown, a note states: 'An inline user task is an extension of a regular user task. Its interface is automatically defined by the data that is exposed to it and an implementation is not required.' A link 'Learn More' is provided. At the bottom of the 'Implementation' section, there are buttons for 'View template:', 'New', 'Select...', and 'X'. Below these buttons is a button labeled 'Create Stand-alone Implementation...' with an orange border around it.

- __ h. Click the **Approve Hire Request** activity. In the properties panel below, go to the **Implementation > Activity Type** menu, change this activity to **User Task**. Both activities now have the same icon.



- __ 3. Create an activity that is named **Approve New Hire Request**.
- __ a. Right-click the **Approve Hire Request** task and select **Copy**.

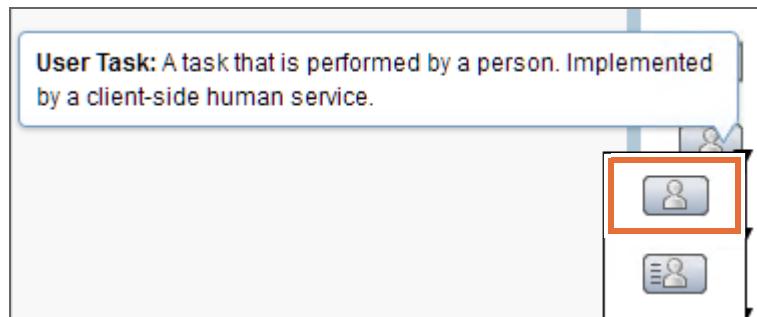


- __ b. Right-click on the canvas and select **Paste**.

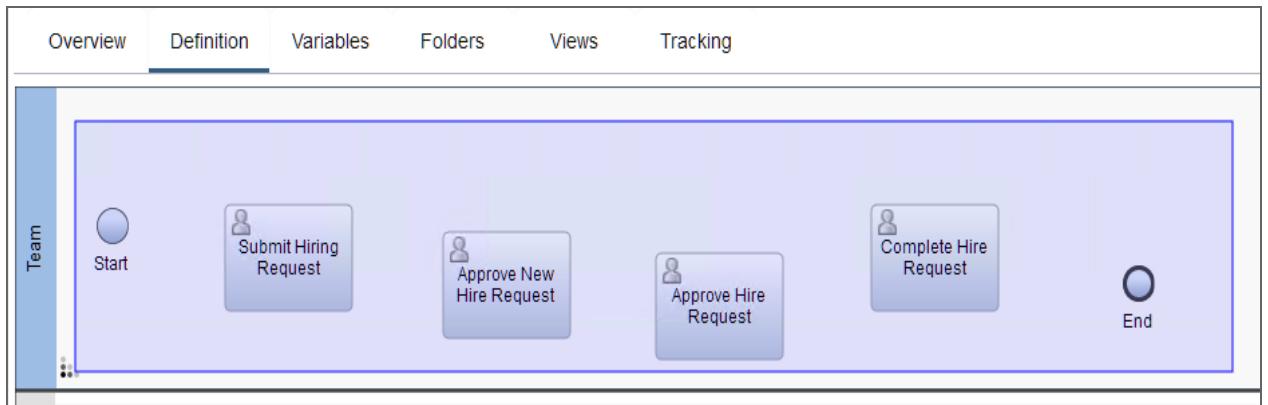


- __ c. Rename the task to: **Approve New Hire Request**.

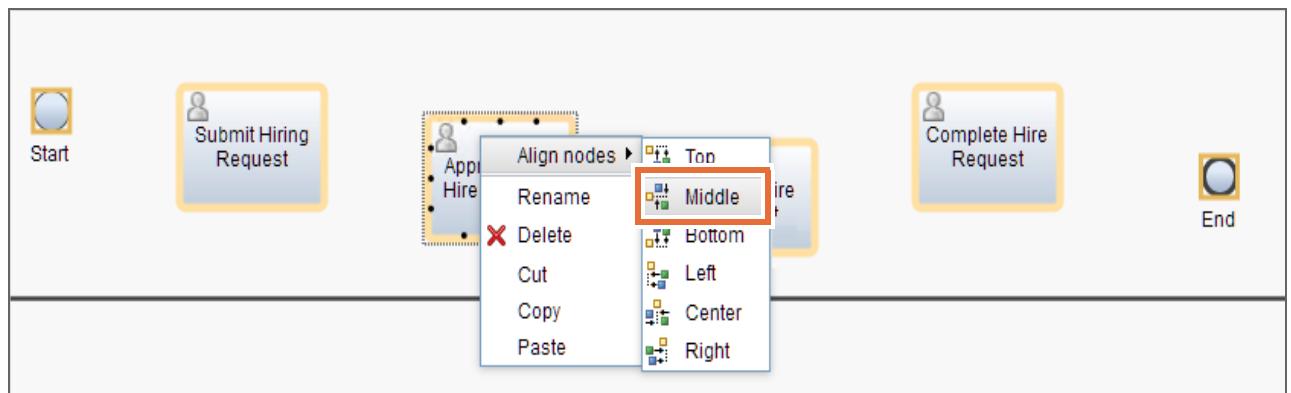
- ___ 4. Create an activity named **Complete Hire Request**.
 - ___ a. Expand the activity icon in the palette by clicking the down arrow and drag a **User Task** to the canvas.



- ___ b. Rename the activity to **Complete Hire Request**.
- ___ 5. Sequence the activities and add sequence flow lines.
 - ___ a. Place the activities in line by using the following sequence:
Submit Hiring Request > Approve New Hire Request > Approve Hire Request > Complete Hire Request
 - ___ b. Lasso all objects by holding down the left mouse button and highlighting all objects including the start and end events. After all are highlighted, release the mouse button.



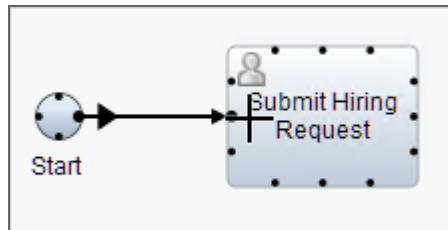
- ___ c. With all items highlighted (yellow halo denotes selection), right-click one of the items and select **Align > Middle**. The result is all artifacts are aligned.



- ___ d. Hover over the right portion of the **Start** event until you see the right-arrow connector.



- ___ e. Click on the arrow. While holding down the **left mouse button**, drag the **arrow** until it connects to the first activity. When the cross displays and the activity is highlighted, you can release the mouse button.

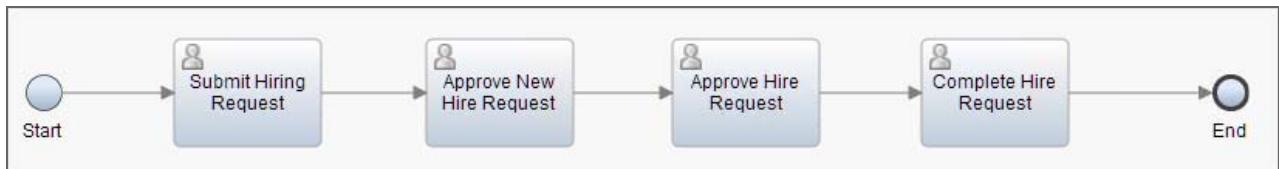


- ___ f. Repeat adding sequence flows for the remaining activities.

- ___ g. Click **Finish Editing** to save your changes.



- ___ h. Verify your process resembles the screen capture below.



Although this is the correct sequence of activities in the process, the roles that complete the activities have not been defined yet. You do this in the next section.



Optional

You completed parts 1-3 out of 6, so you are now approximately half done with this exercise. Now would be a good time to take a 10-minute break.

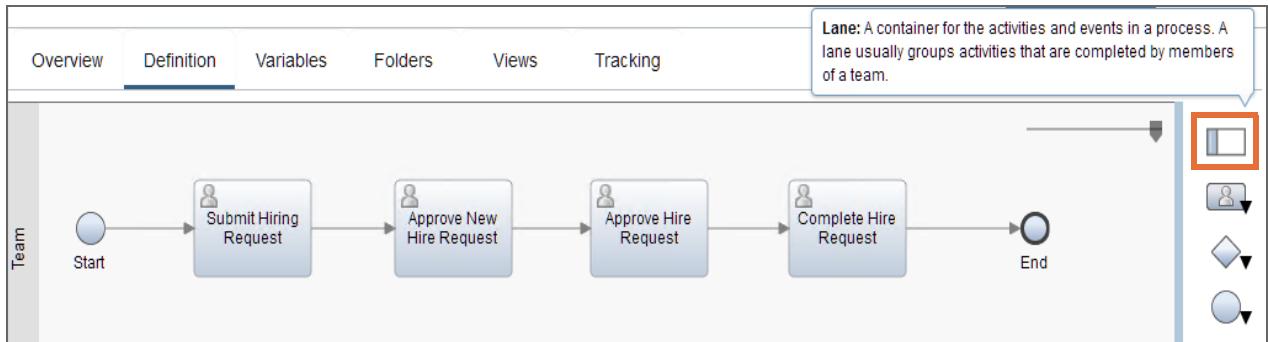
Part 3: Model the process teams

When creating a new process, by default you have a team lane, a system lane, a start event, and an end event. In this section, you add the necessary team lanes to the two default lanes (team and system) and rename the lanes.

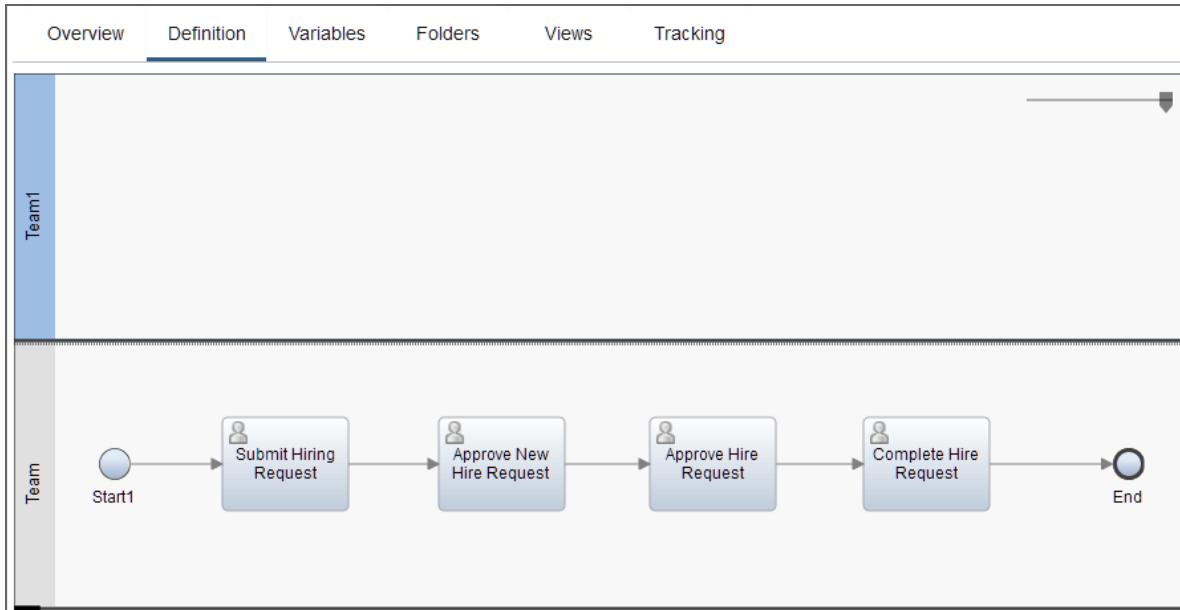
- ___ 1. Add a team lane to the process.

Read the core requirements in this exercise, pages 2-3 and 2-4, and identify the teams. In the process requirements, two teams are in the main process: Hiring Manager and General Manager. Also, the process has one System lane for a total of three lanes.

- a. Click the **Lane** icon and drag one lane from the palette to the canvas above the Team lane until you see the green (+) symbol. When you release your mouse button, the new lane appears.



- b. Verify a lane with name **Team1** is added to the process.



Hint

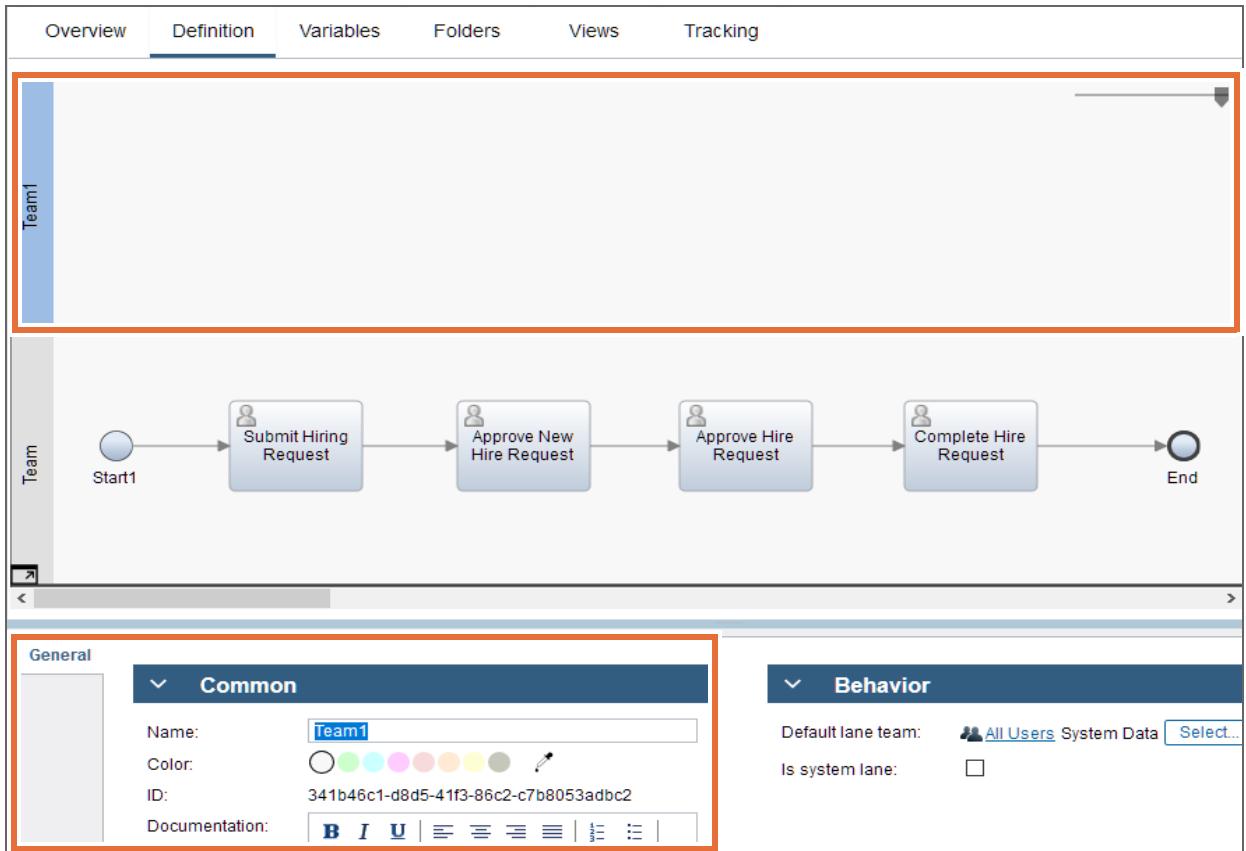
If the lane is below, you can drag it above the Team lane using the mouse. To drag the lane, left-click the Team lane label and drop it where appropriate.

__ 2. Set the properties of the team lanes.

__ a. Use the zoom slider at the upper right of the canvas to view the whole process if it doesn't fit on the screen.



__ b. Click the top lane to select the entire lane. The **General** properties tab populates at the bottom of the screen.



__ c. In the **General > Common** section, change the name of the top lane from **Team1** to: Hiring Manager

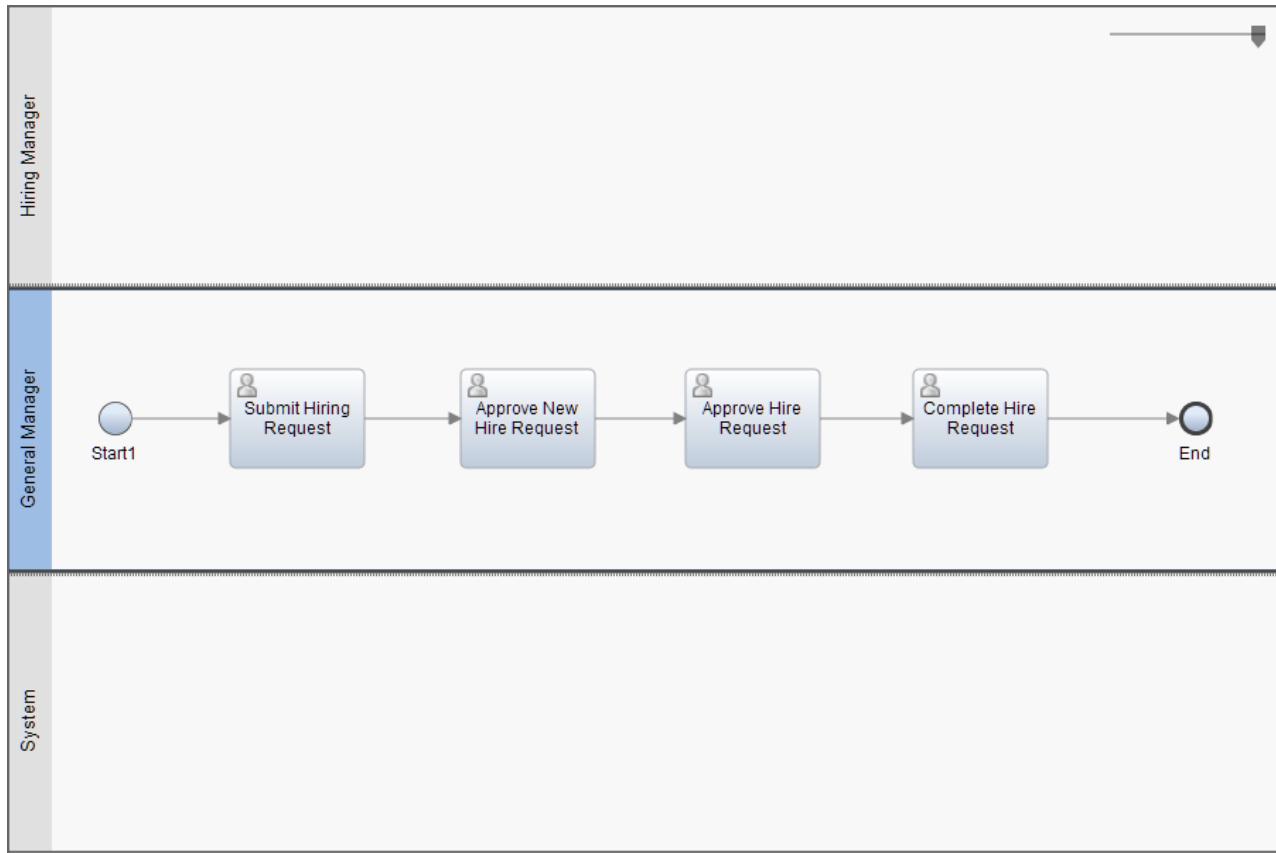
__ 3. Rename the second lane labeled **Team**.

__ a. Select the lane that is labeled **Team** and double-click the name.

__ b. Change the name of the middle lane from **Team** to: General Manager

This step leaves the bottom lane as the **System** lane. It is a good practice to keep the **System** lane at the bottom of your process, even if you have no system lane activities

Your process model now has the appropriate Team lanes defined. However, the activities are not assigned to the appropriate Team lanes. You complete this activity in the next section.

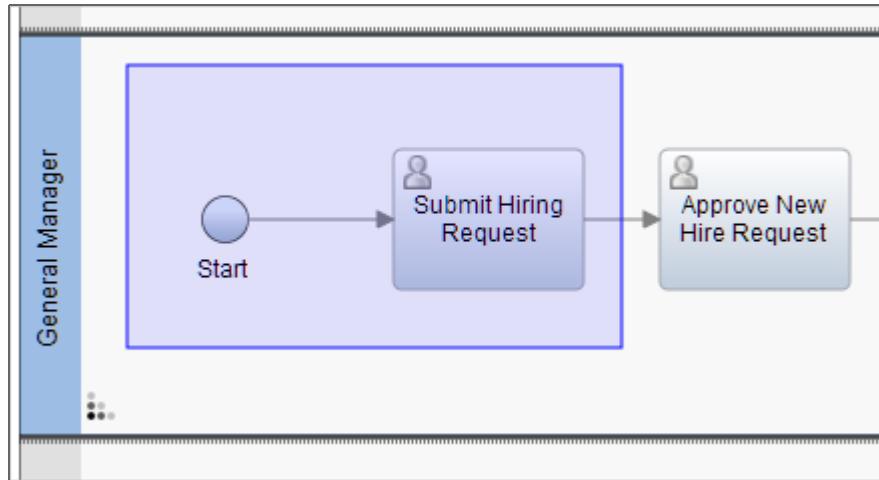


Part 4: Assign the activities to the appropriate Team

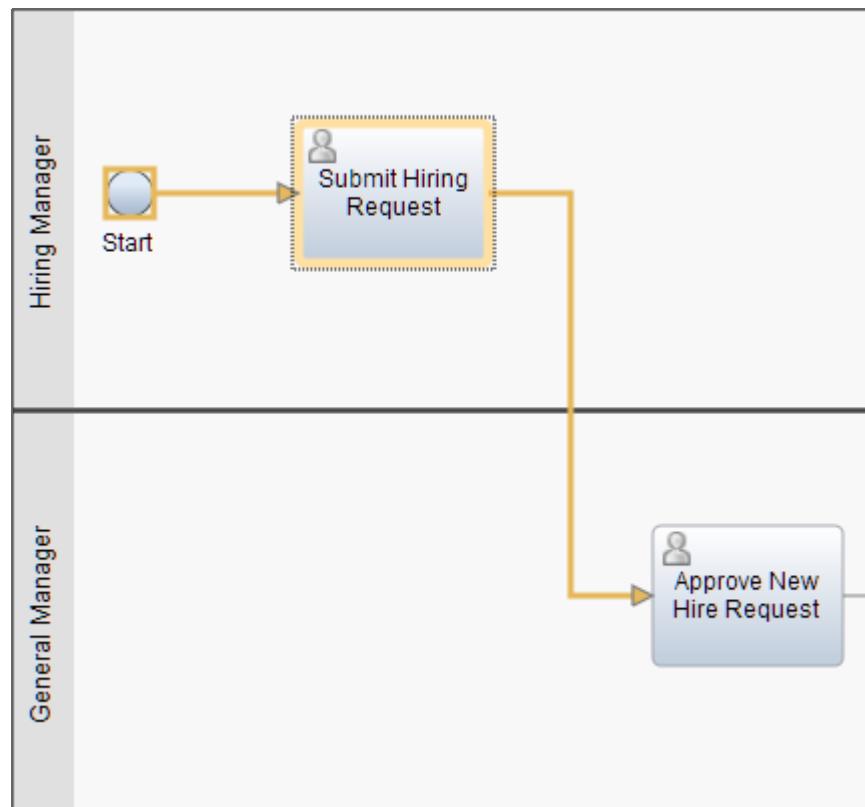
You identify the teams that are associated with the lanes instead of allowing all users to claim the tasks that are associated with each activity. Determine which teams conduct each of the four activities. From the process requirements, you determine the following assignments:

- Hiring Manager (team): Submit Hiring Request
- General Manager (team): Approve New Hire Request
- System (team): Approve Hire Request and Complete Hire Request

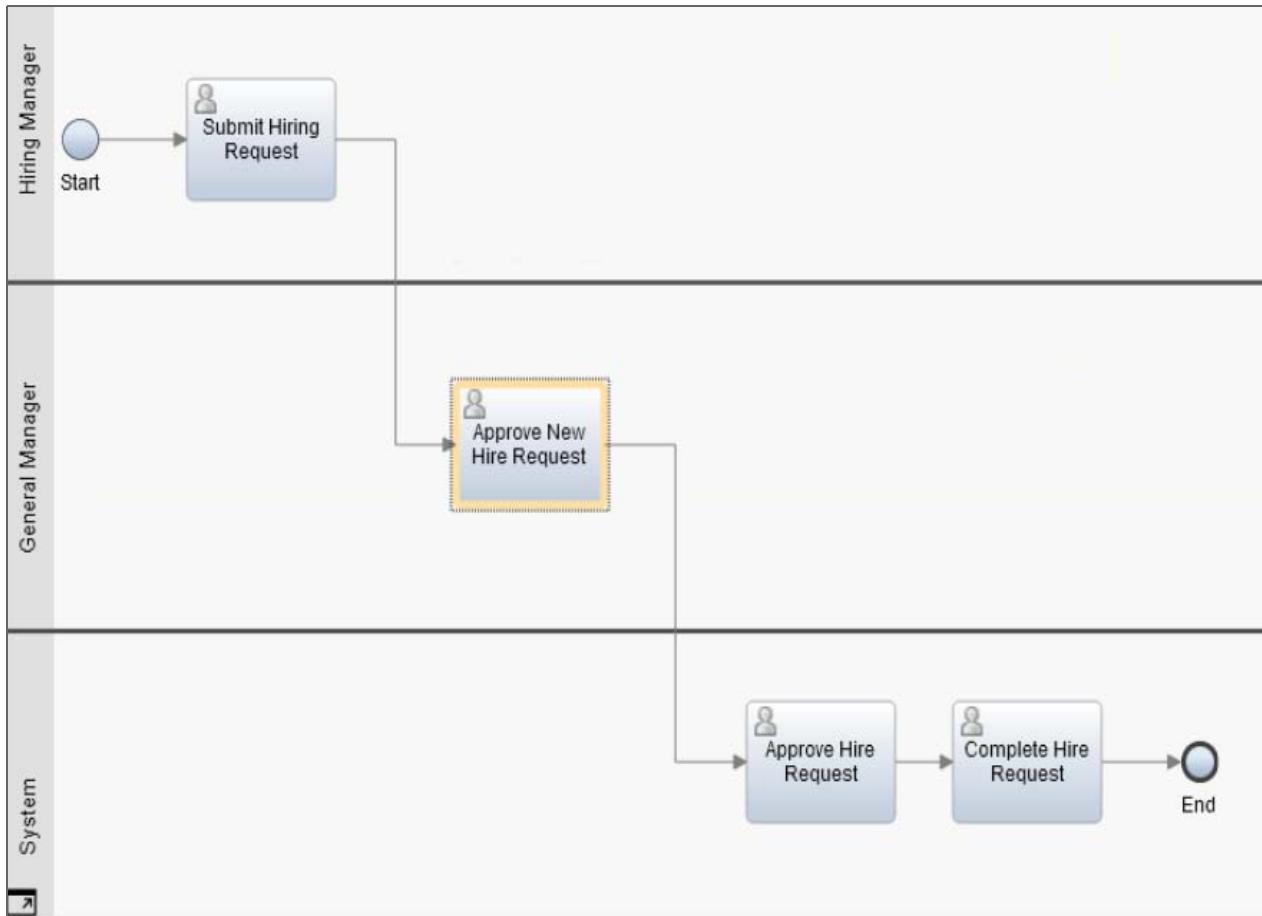
- ___ 1. Arrange the activities to the team lanes to establish the correct activity assignment.
 - ___ a. Select the first two items in the process. To drag the Start event and the Submit Hiring Request activity to the **Hiring Manager** lane, lasso both items by highlighting a box around both items with your cursor.



- ___ b. With the two items highlighted (yellow halo), drag the selected activities to the **Hiring Manager** lane.



- ___ c. Leave the Approve New Hire Request activity in the General Manager Team lane and drag the **Approve Hire Request** and **Complete Hire Request** activities, along with the **End** event, to the System lane. Later in the exercise you rearrange this activity in a new lane: Approvers.

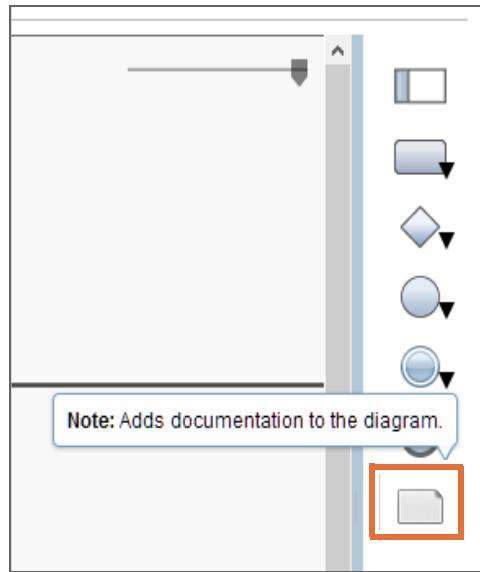


___ 2. Add an annotation

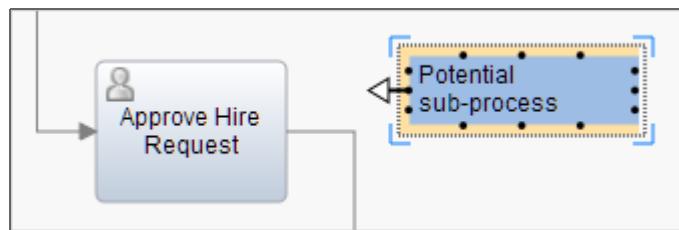
It is common to place annotations in the process when first modeling it. They can serve as placeholders for questions or concerns while the process is in development. The process owner expressed that the approval process might involve multiple steps and departments. Thus, an annotation is placed in the model mentioning this step might be a potential subprocess. At this point, it might not be known whether it is to be implemented as a linked process or subprocess.

- ___ a. Verify you are displaying the Definition tab of the Hiring Request Process.

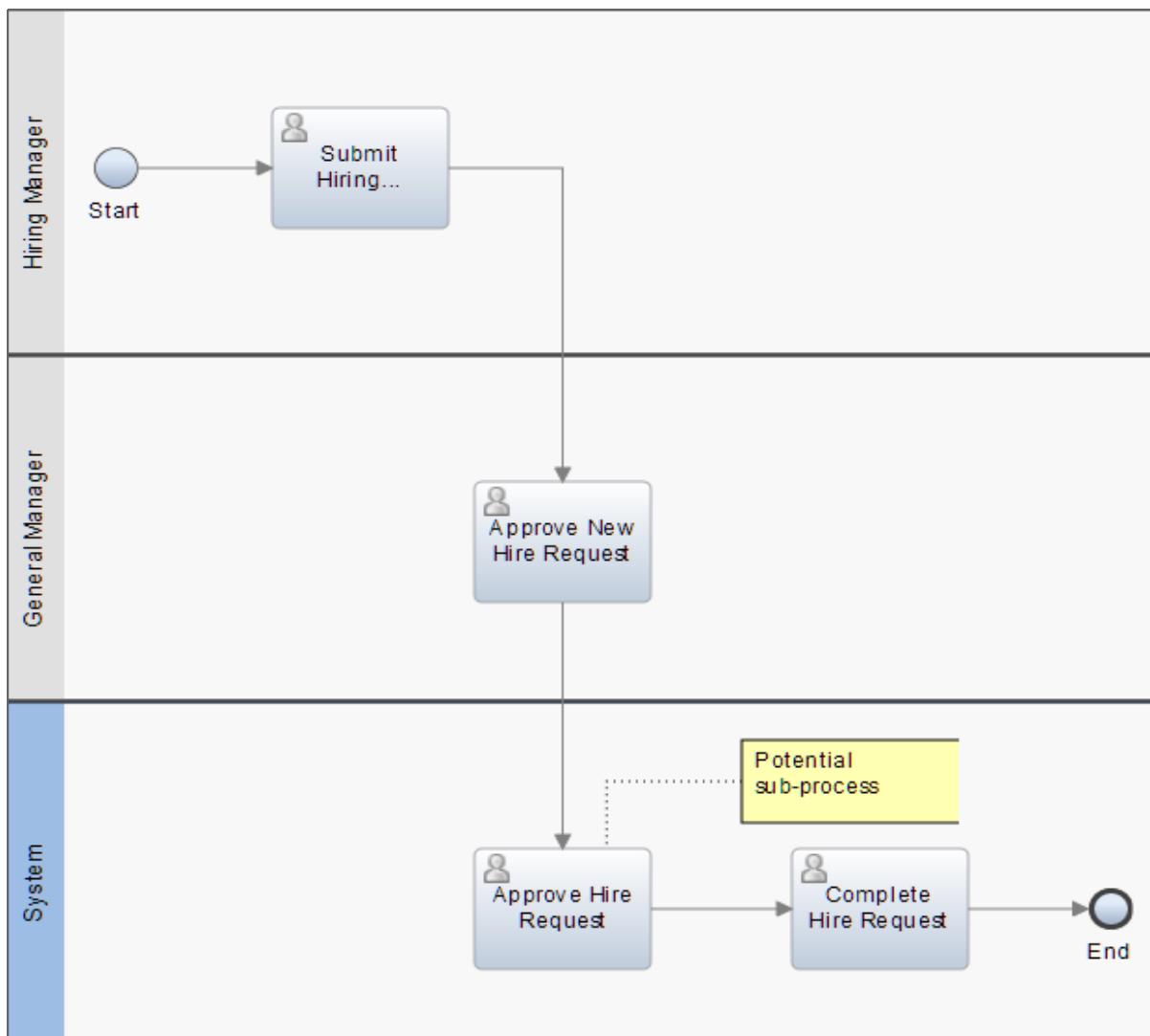
- __ b. Drag an **annotation** to the canvas next to the activity **Approve Hire Request**.



- __ c. Double-click the **annotation** and add the text: Potential sub-process.
__ d. Highlight the left portion of the **annotation** until an arrow appears and draw a line from the annotation to the activity Approve Hire Request.

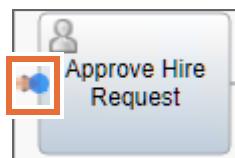


- ___ e. Save your process. The first process is complete for this exercise. Next, you create another process, a linked process.



Hint

Your sequence lines might be different. You can drag sequence endpoints from one activity to the other. To do this, highlight the endpoint until it displays as bright blue. Then, drag the endpoint to another activity and release.



It can be tricky to highlight the endpoint correctly. As you become familiar with the Process Designer tool, you can choose the methods that work best for you. For instance, if it becomes too cumbersome to drag sequence lines from one activity to another, you can delete them and rebuild them.

Part 5: Create a linked process

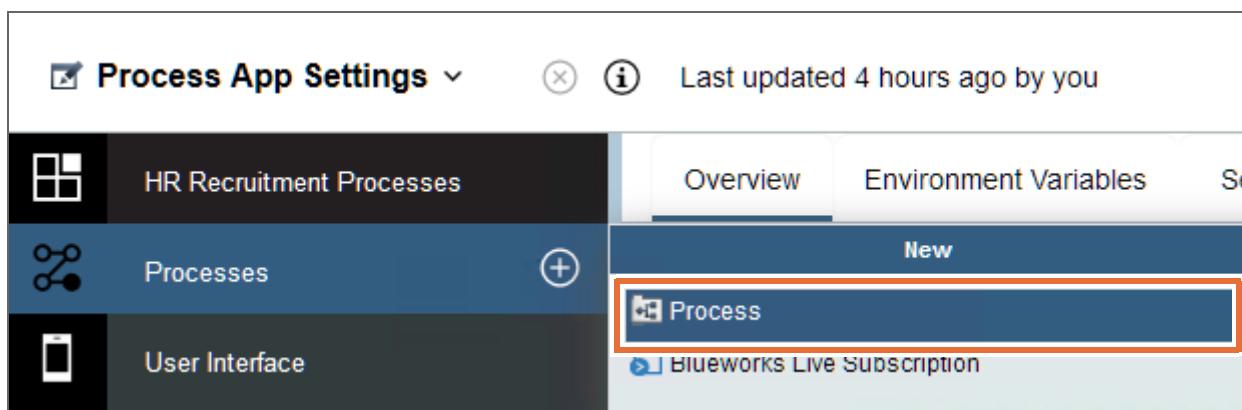
Determine whether any additional work is required for each activity and if so, what the work involves.

- **Submit Hiring Request:** Generally no rework is needed.
- **Approve New Hire Request:** Extra work that is required for the General Manager to record a rejection reason.
- **Approve Hire Request:** Extra team work that the HR Administrator and the Hiring Manager require.
- **Complete Hire Request:** Generally no rework is needed.

Because more than one team is involved in the **Approve Hire Request** activity, it is no longer a “task” but a linked process. The linked process has lanes, such as System, HR Administrator, and Hiring Manager. Activities include Check Hire Request, Override Hire Request, and Negotiate Hire Request.

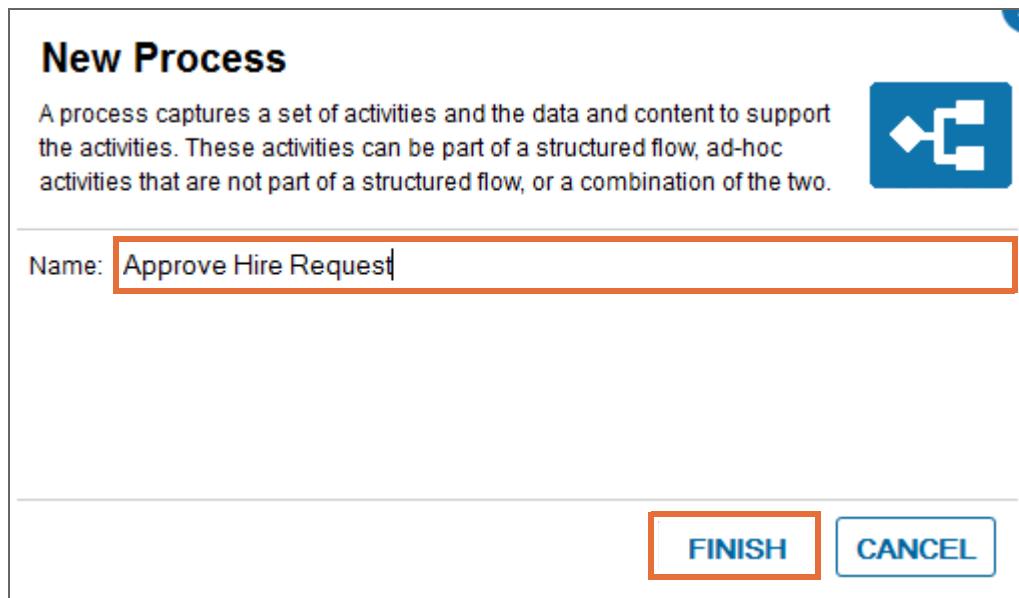
There are two ways to implement a nested process. If the process is not to be reused, the use of a subprocess is appropriate. The subprocess is part of the parent process and shares all data. This is done if the process is becoming too large and complex. A subprocess cannot be shared. However, a linked process is a completely separate process that can be shared. You must determine whether you want to use a subprocess or linked process. Because **Approve Hire Request** might be reused in other HR processes, you create a linked process for this scenario.

1. Create a process that is called **Approve Hire Request**.
 - a. From the process library, hover over **Processes** category and then click the (+) plus sign.
 - b. From the **New** menu, click **Process**.

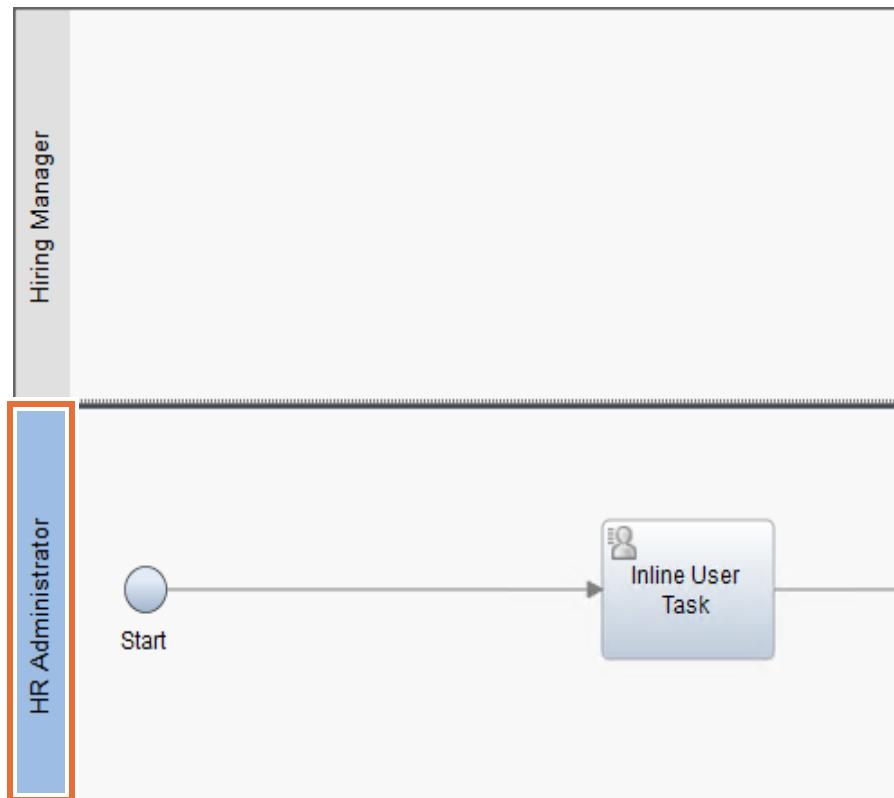


- c. Enter Approve Hire Request for the **Name**.

- ___ d. Click **FINISH**.

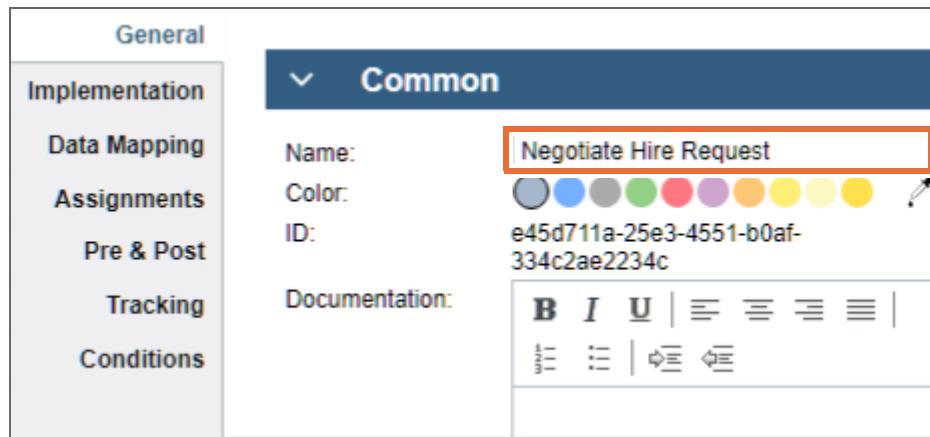


- ___ 2. Create the System, HR Administrator, and Hiring Manager lanes.
- ___ a. Click the **Lane** icon from the palette and drag one lane from the palette to the canvas above the default Team lane.
- ___ b. In the **General > Common** section for the new Team1 lane, change the name to: Hiring Manager
- ___ c. Highlight the second lane and rename it to: HR Administrator

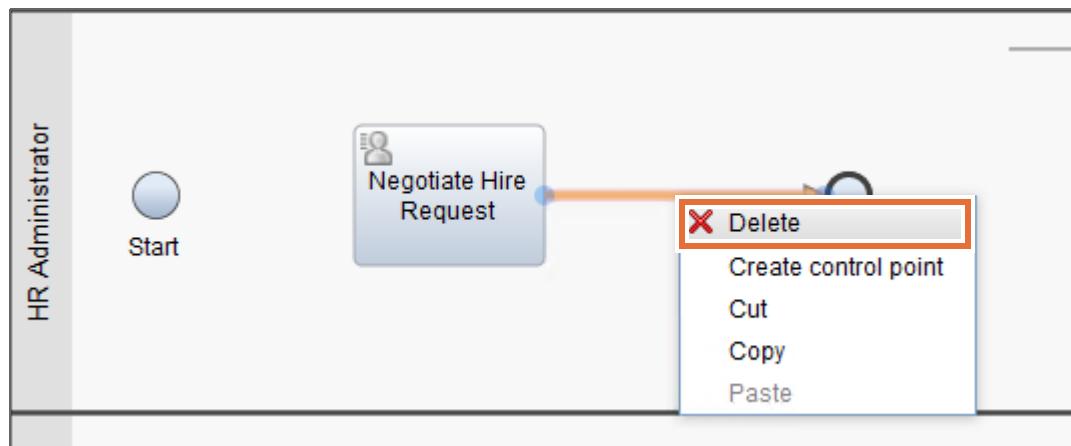


- ___ 3. Create the Check Hire Request, Override Hire Request, and Negotiate Hire Request activities.

- ___ a. Rename the **Inline User Task** activity to: Negotiate Hire Request



- ___ b. Delete all the **flows** on the canvas.

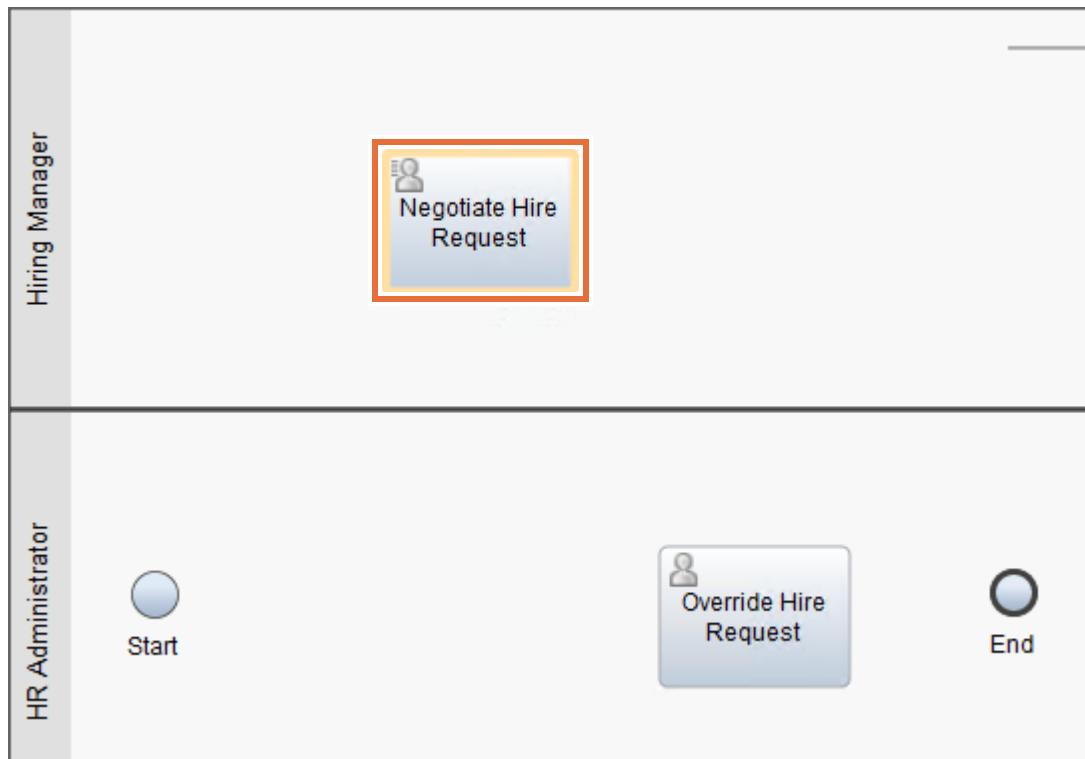


- ___ c. Expand the activity icon in the palette by clicking the down arrow and drag a **User Task** to the right of Negotiate Hire Request on the canvas.

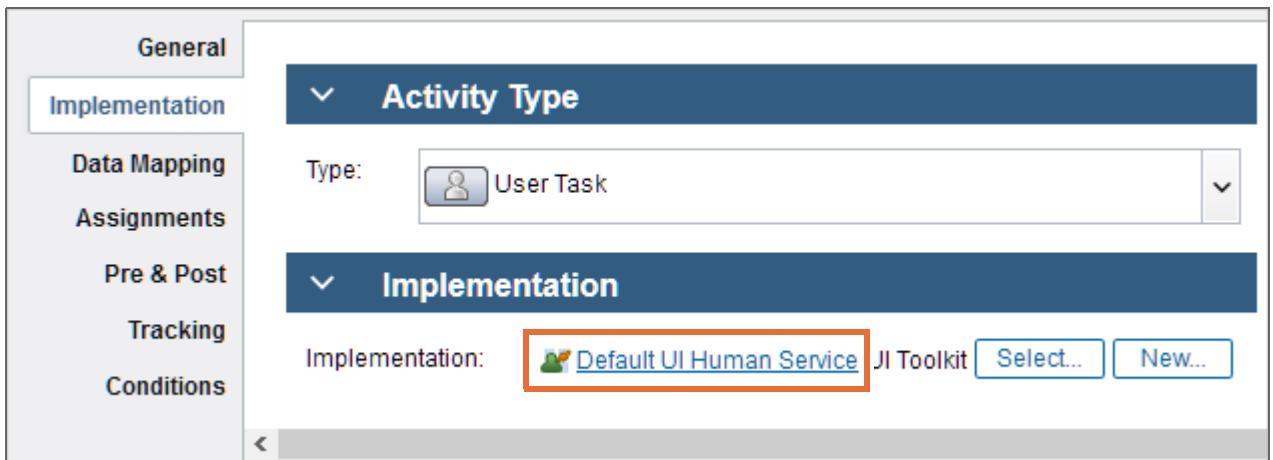


- ___ d. Name the activity **Override Hire Request**.

- __ e. Move the **Negotiate Hire Request** activity to the Hiring Manager lane.



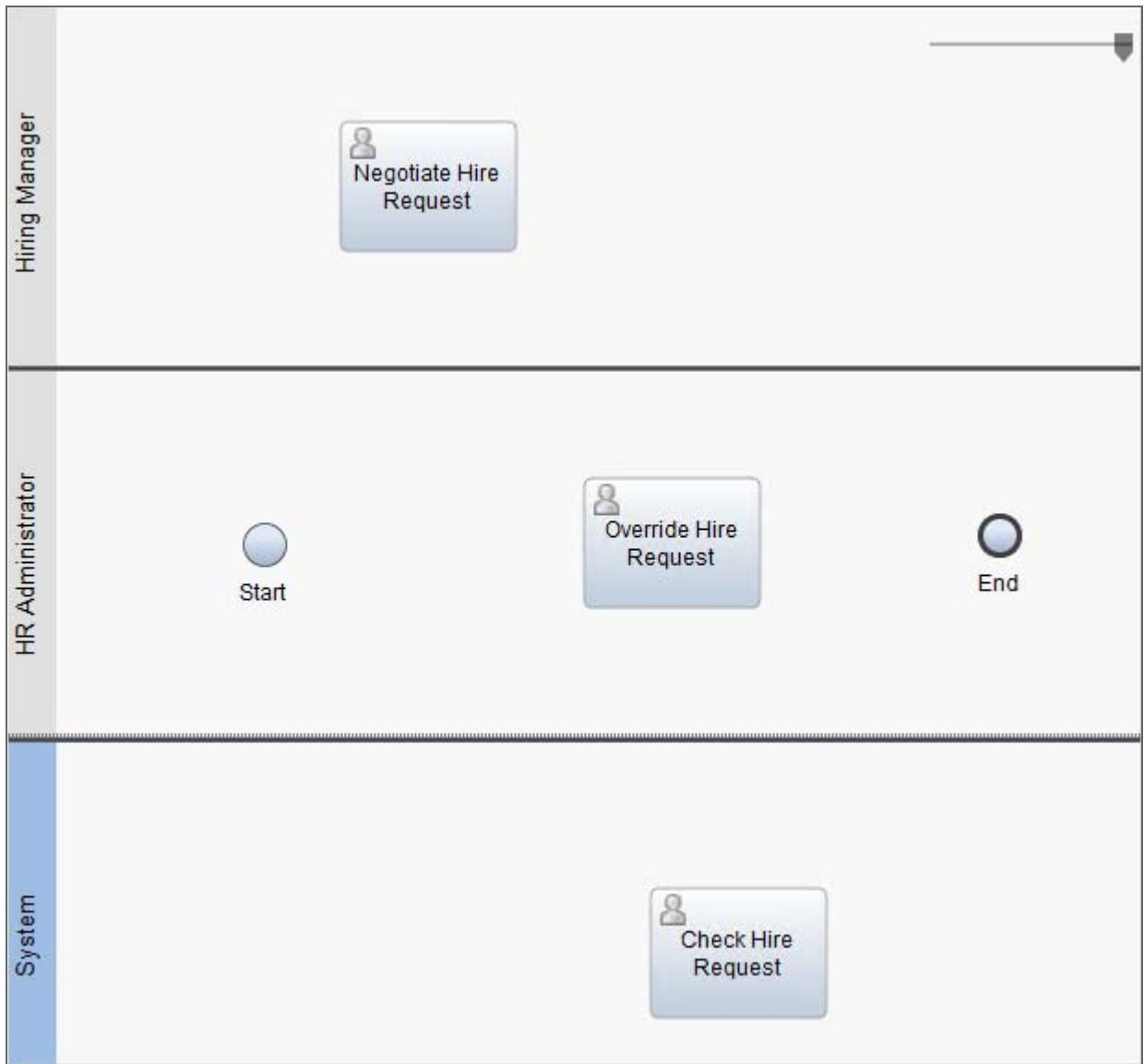
- __ f. While Negotiate Hire Request is selected, change the activity to a **User Task** in the **Implementation > Activity Type** panel.
 __ g. In the Implementation section, change the implementation to **Default UI Human Service**.



- ___ h. Drag another User Task to the System lane and name the task: Check Hire Request. The selection that you chose last becomes the default for each artifact in the palette. To drag the User Task now, you don't need to select the down arrow since it now defaults to the User Task selection.



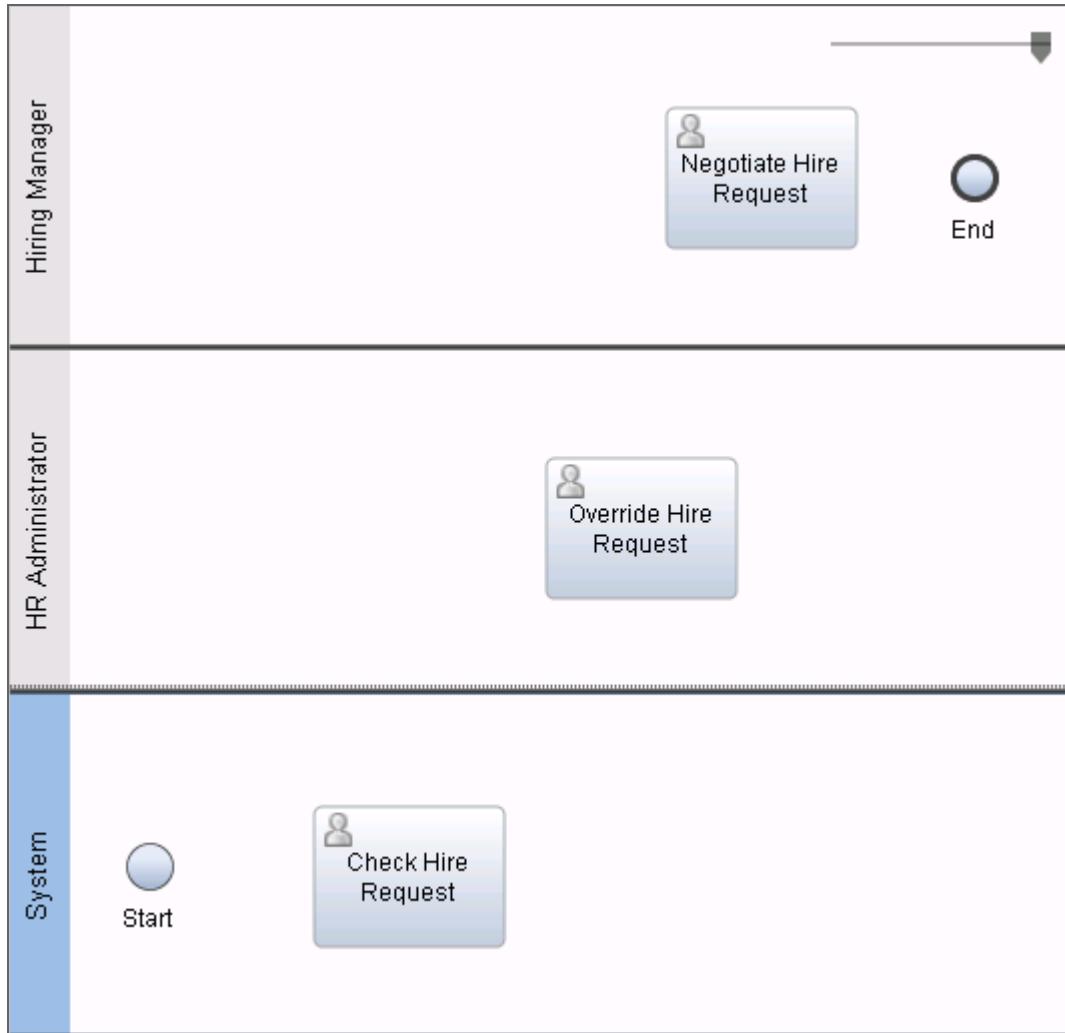
- ___ i. The locations of your activities should match the screen capture below.



- ___ 4. Wire the activities

- ___ a. Move your **Start** event before the first activity, **Check Hire Request** in the System lane

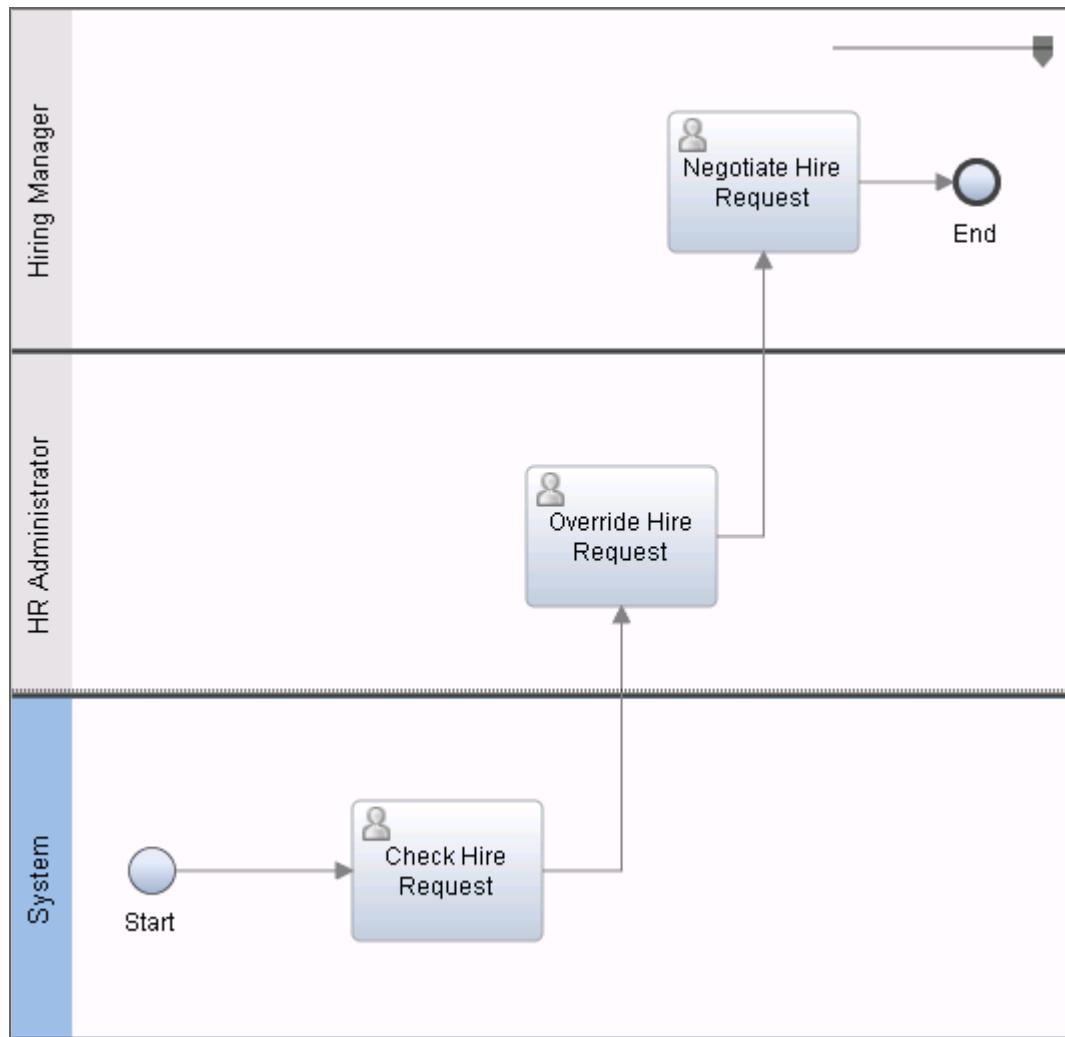
- ___ b. Move the **End** event after the **Negotiate Hire Request** activity in the Hiring Manager lane.
- ___ c. Arrange the activities in an upward diagonal line.



Although the layout breaks the preferred left-to-right, top-to-bottom approach because the process starts in the system lane, most developers keep the system lane as the bottom lane. Communicating a clear, concise model is the most important goal, and even though this model breaks the preferred practice, it meets the readability goal.

- ___ d. Create a sequence flow from the **Start** event to **Check Hire Request**.

- __ e. Repeat adding sequence flows for the remaining activities.



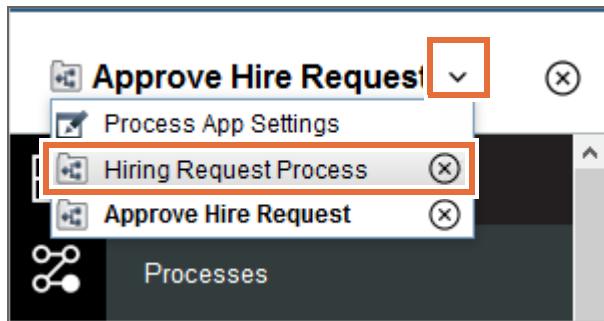
- __ f. Save your process application.

Part 6: Attach the linked process

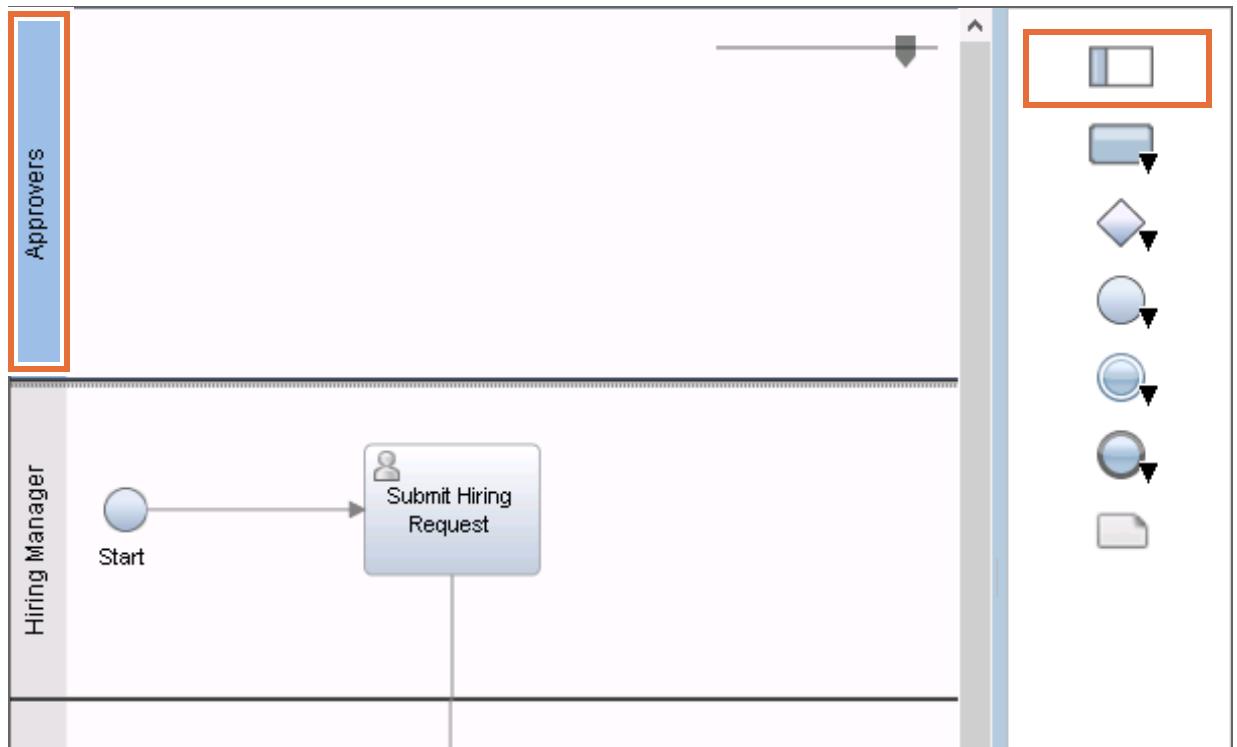
- 1. Add the **Approvers** lane to the parent process.

To accommodate a subprocess with multiple lanes, a new lane is added in the parent process to encompass the roles depicted in the subprocess.

- a. Return to the Hiring Request Process by opening the **History** menu at the top and selecting **Hiring Request Process**.



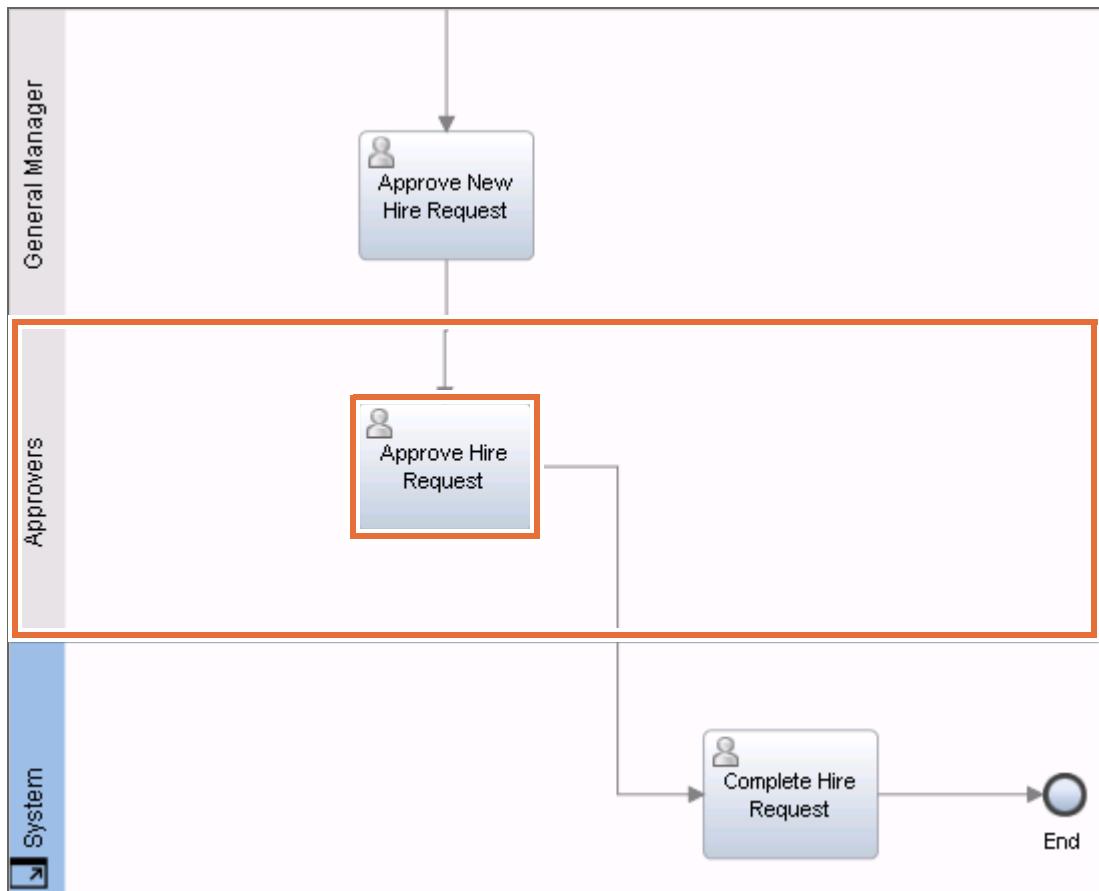
- b. Delete the **annotation** mentioning the potential subprocess. This annotation served as a temporary placeholders to help in understanding the process. Now that the linked process is being implemented, there is no need for the annotation. In a real-world implementation, this would likely occur over a period of time.
- c. Drag a new lane above the Hiring Manager lane in the diagram and name it: **Approvers**



- ___ d. Right-click the **Approvers** lane and click **Move lane down** twice. Continue doing this until it is above the **System** lane.

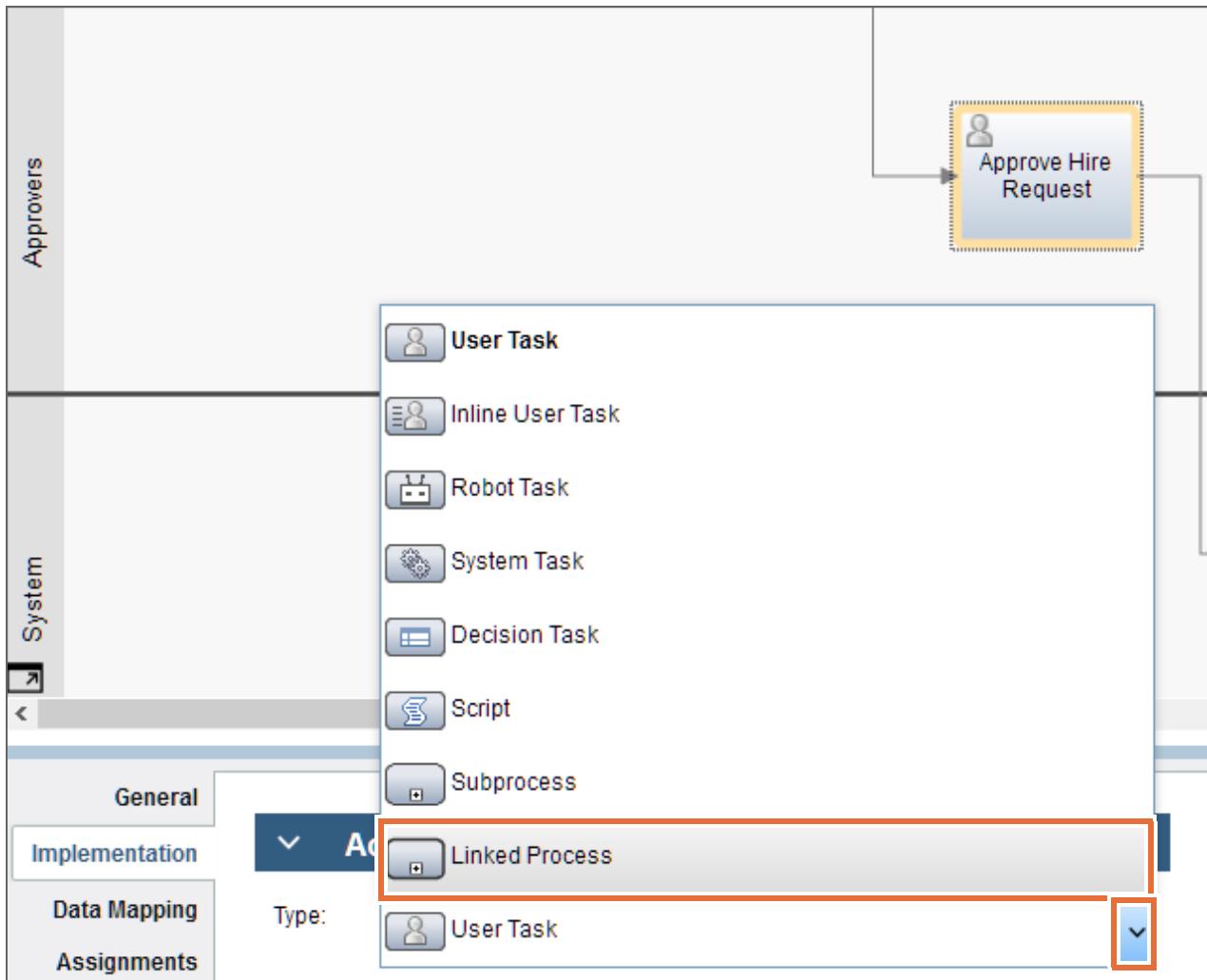


- ___ e. Drag the existing **Approve Hire Request** activity that is in the System lane into the **Approvers** lane.

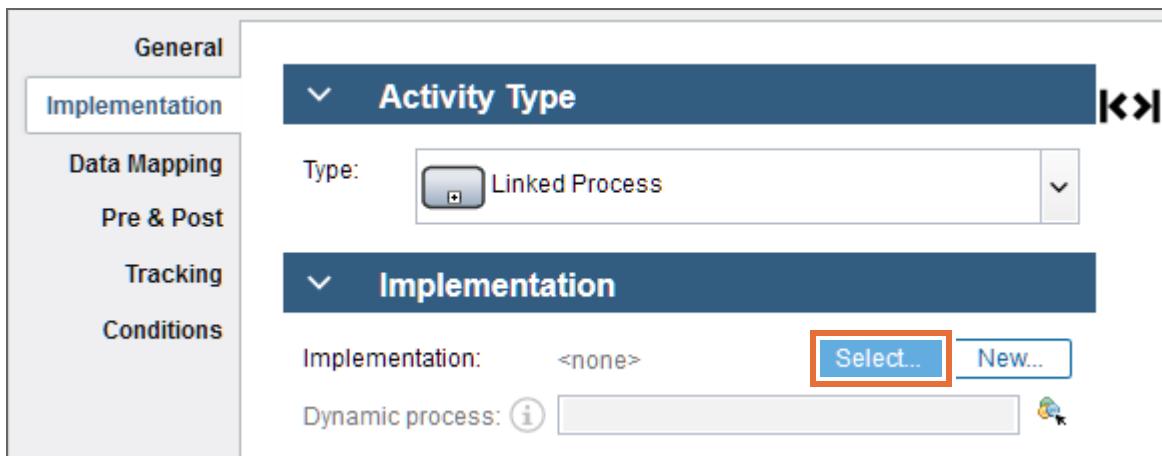


- ___ f. Realign the activities in their respective lanes so that the sequence flow lines are straight.
- ___ g. Save your work.
- ___ 2. Change the implementation of the existing **Approve Hire Request** activity to a linked process.
- ___ a. With the **Approve Hire Request** activity selected, open the **Implementation** menu.

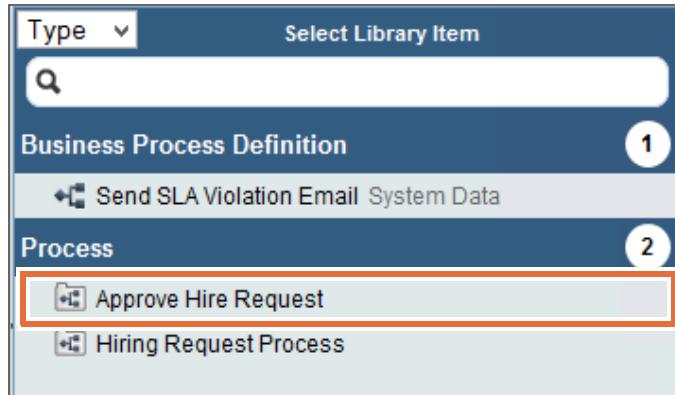
- __ b. In the Activity Type section, select the Type as **Linked Process**.



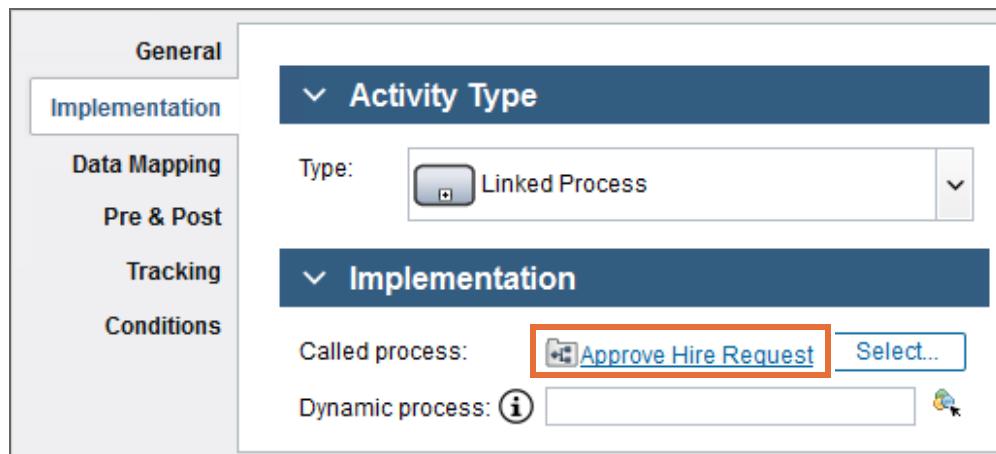
- __ c. Under the Implementation section, click **Select** next to Implementation.



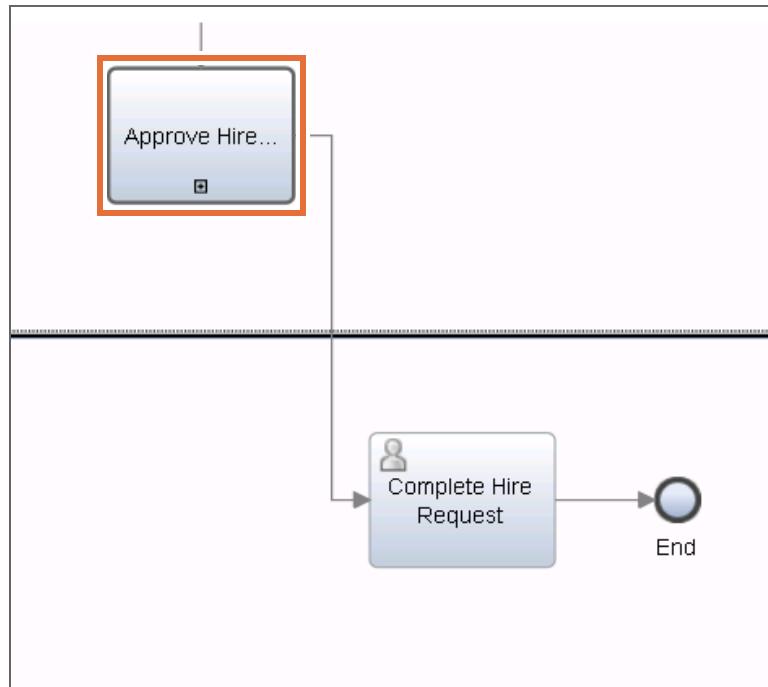
- ___ d. Select **Approve Hire Request** from the Process menu.



- ___ e. The process now shows **Approve Hire Request** as a linked process.



- ___ f. The Approve Hire Request activity is now changed to a linked process. In Business Process Model and Notation, this is denoted with a **plus-sign**.



__ g. Save your work.

You have completed this exercise.

In this exercise, you modeled the parent and linked processes in the Workflow Center for the HR Recruitment Processes solution. First, you converted business process workflow steps that are documented in the process discovery and analysis into process model tasks. Then, you defined and modeled team lanes in the business process and assigned the tasks to the appropriate teams. Finally, you created a subprocess and linked it to the parent process. The next step is to add gateways to direct the process. This is covered in the next exercise.

End of exercise

Exercise 3. Playback 0: Controlling process flow

Estimated time

01:30

Overview

This exercise covers how to create gateways in a business process, and how to create timer intermediate events. Validation is accomplished through a review session with all business stakeholders, business users, and the BPM development team. This unit describes the Playback 0 validation goals and requirements, explains how to validate that a process model meets the goals and requirements, and describes how to reach consensus on the process model. At the end of this exercise, you complete Playback 0.

Objectives

After completing this exercise, you should be able to:

- Add gateways to a process
- Model the appropriate sequence flows for each gateway
- Add a timer intermediate event to a process based on business requirements
- Model an escalation path in a process with IBM Process Designer
- Add a new swimlane and activity for legal review to meet additional requirements.
- Validate the process application and create a snapshot

Introduction

The purpose of this exercise is to add all the gateways necessary to model the flow control for the process.

In this exercise, you add a timer intermediate event that helps satisfy newly identified requirements for the Hiring Requisition process. You are also provided new requirements regarding a legal review. After developing from the new requirements, you validate the process application and create a snapshot.

Throughout the exercise, you switch back and forth several times between the Hiring Request Process and Approve Hire Request processes. Take additional care when working through the lab to make sure that you work in the right process.

This exercise is long, so a break is included in the middle of the instructions.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Create gateways for parent process](#)

[Part 2: Create gateways for the linked process](#)

[Part 3: Modeling timer intermediate events](#)

[Part 4: Add new process requirements to the process](#)

[Part 5: Validate the process model and take a snapshot](#)

Exercise instructions

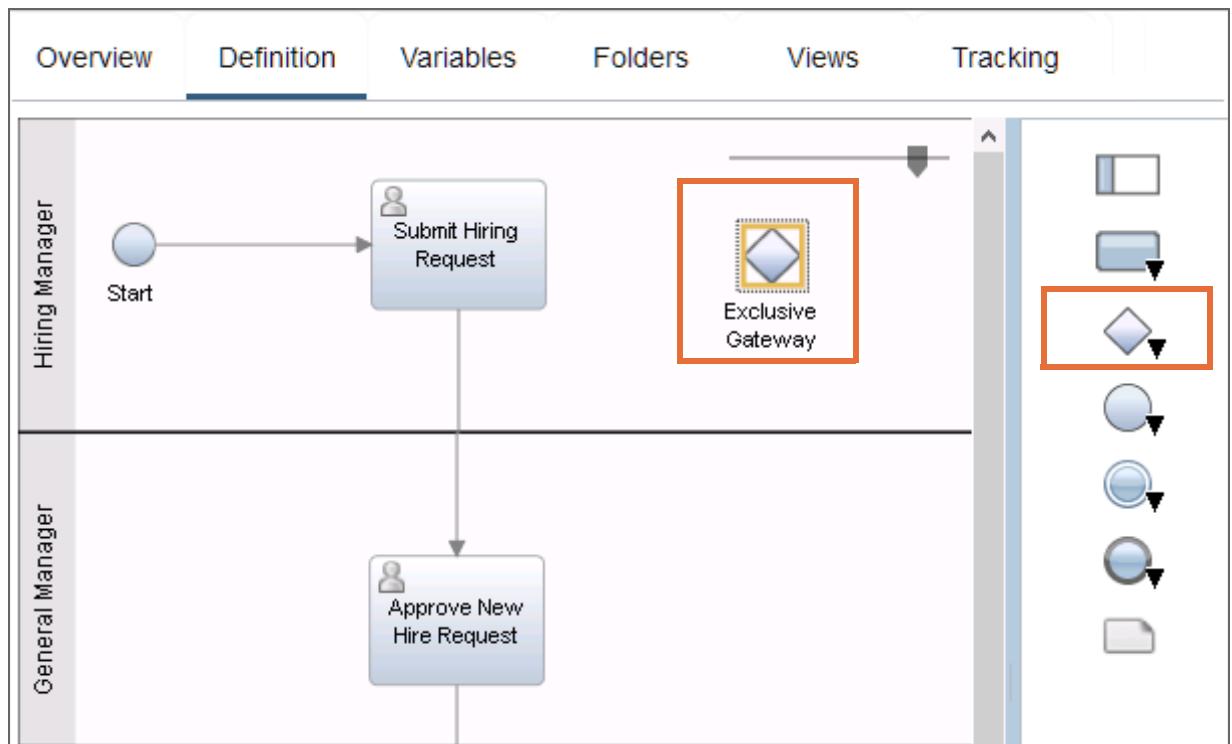
Part 1: Create gateways for parent process

A gateway is represented as a diamond, and can be thought of as a question at a point in the process flow.

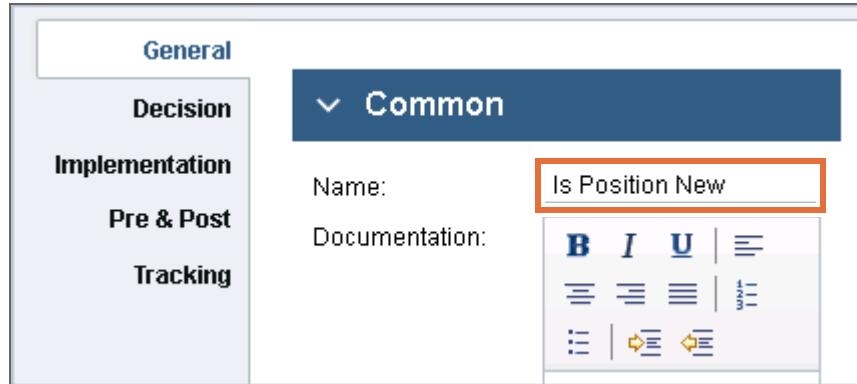
Gateways are added to the parent process, the Hiring Request Process. In this part of the exercise, you create a gateway that is called Is Position New.

In the Hiring Request Process, you need a gateway to direct the process for the General Manager to review the salary after the Submit Hiring Request activity.

- 1. Create the **Is Position New** gateway
 - a. Open the **HR Recruitment Process** process application.
 - b. Open the **Hiring Request Process**.
 - c. In the Hiring Request Process, drag an **Exclusive Gateway** from the palette onto the canvas to the right of the **Submit Hiring Request** activity. You do not need to open the menu to drag an exclusive gateway to the canvas, as the exclusive gateway is the default.



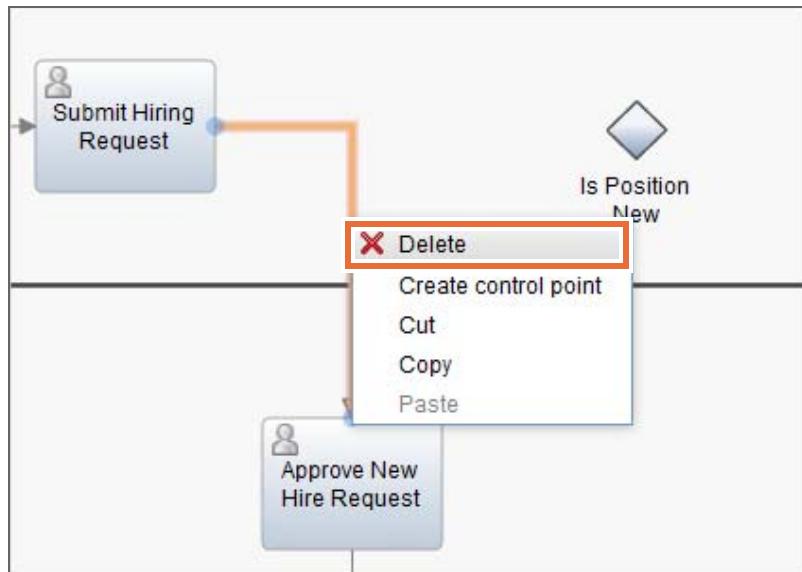
- ___ d. Select the gateway and in the **General > Common** section below, enter **Is Position New** as the **Name**. Because exclusive gateways imply a question, the question mark is unnecessary, and anyone who views the process automatically assumes the question mark.



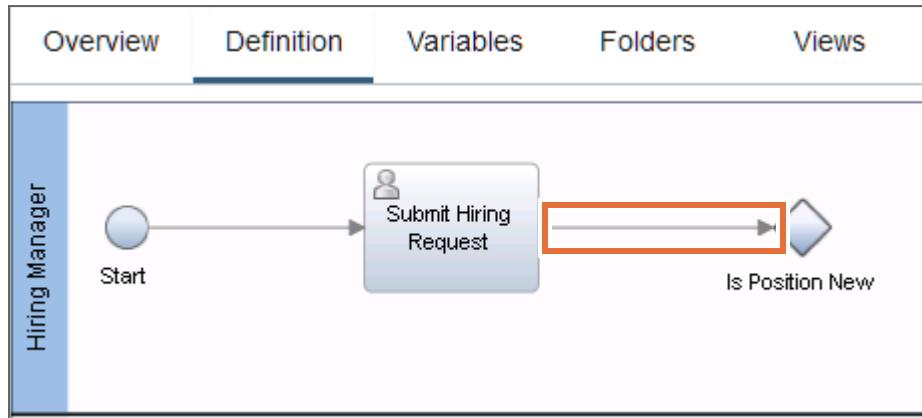
- ___ e. In the **General > Behavior** section, verify that the Gateway type is: **Exclusive Gateway**.



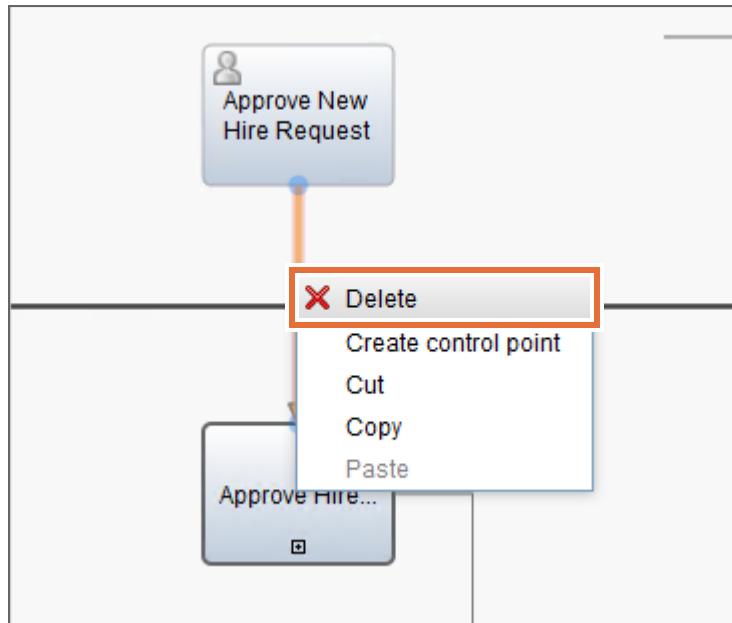
- ___ 2. Add sequence flows to connect the gateway.
- ___ a. Select the sequence flow between **Submit Hiring Request** and **Approve New Hire Request**.
- ___ b. While selected, right-click the sequence flow and click **Delete**.



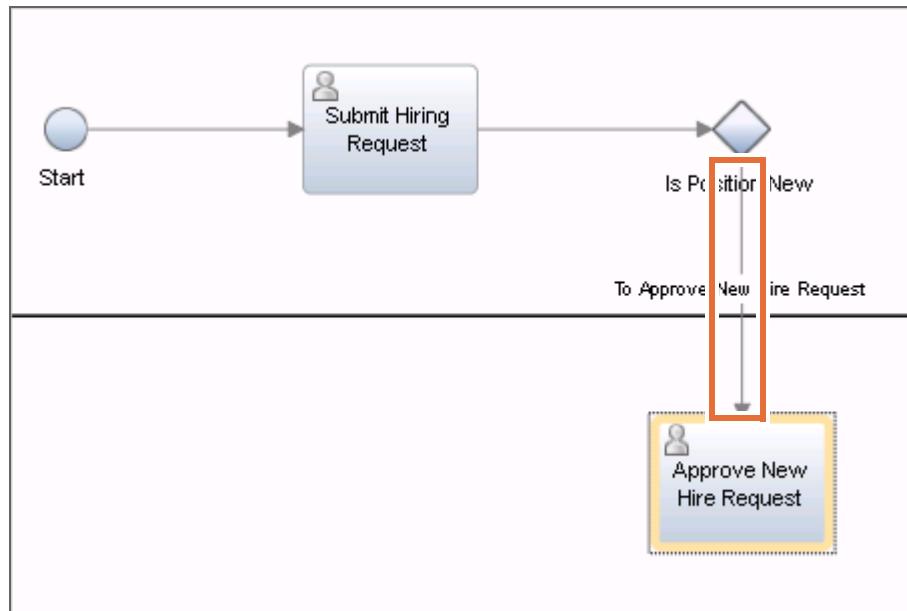
- ___ c. Create a sequence flow between the **Submit Hiring Request** activity and the **Is Position New** gateway. Hover over the right anchor point on the activity until an arrow appears, then drag the arrow to the left anchor point on the gateway.



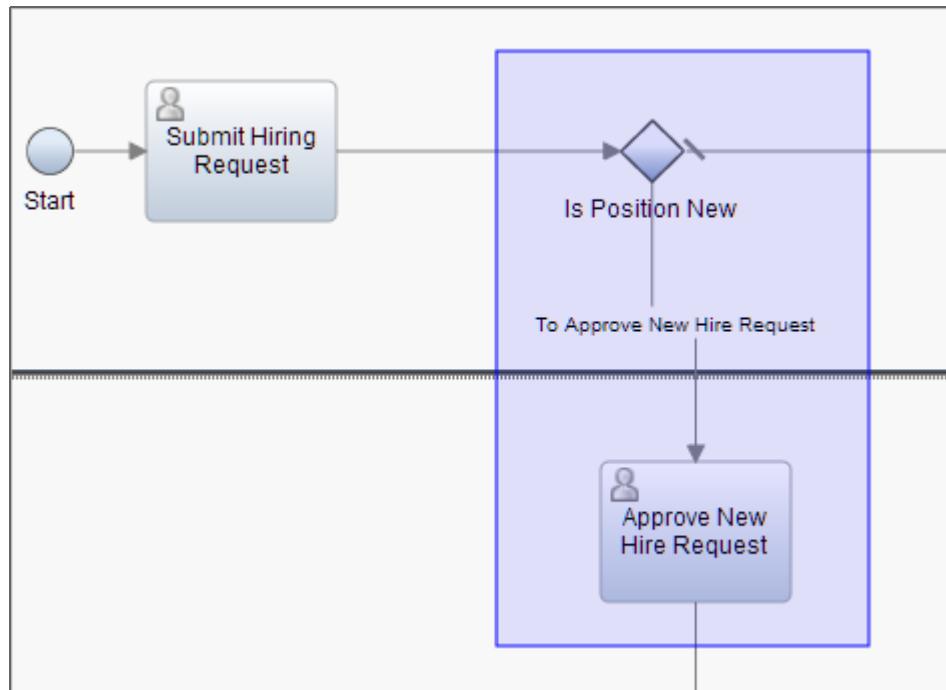
- ___ d. Delete the sequence flow between **Approve New Hire Request** and **Approve Hire Request**.



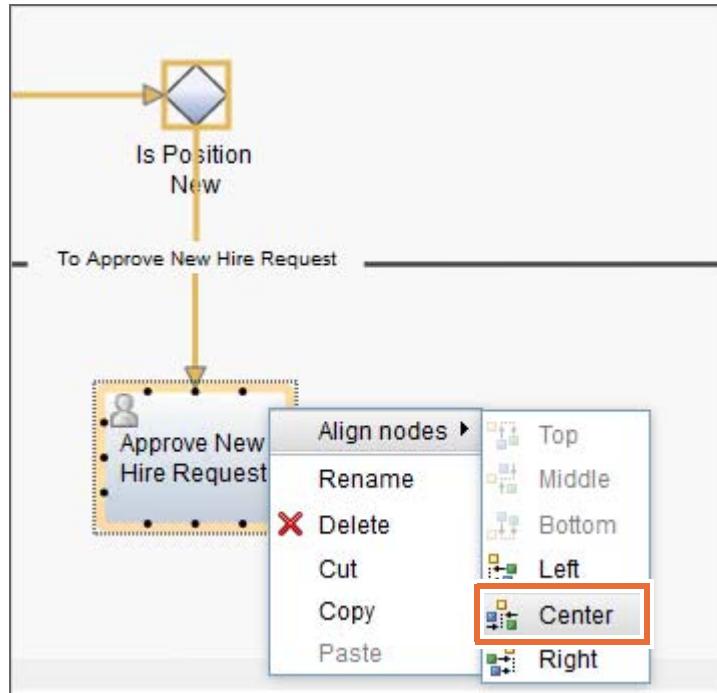
__ e. Connect the **Is Position New** gateway to the top of **Approve New Hire Request**.



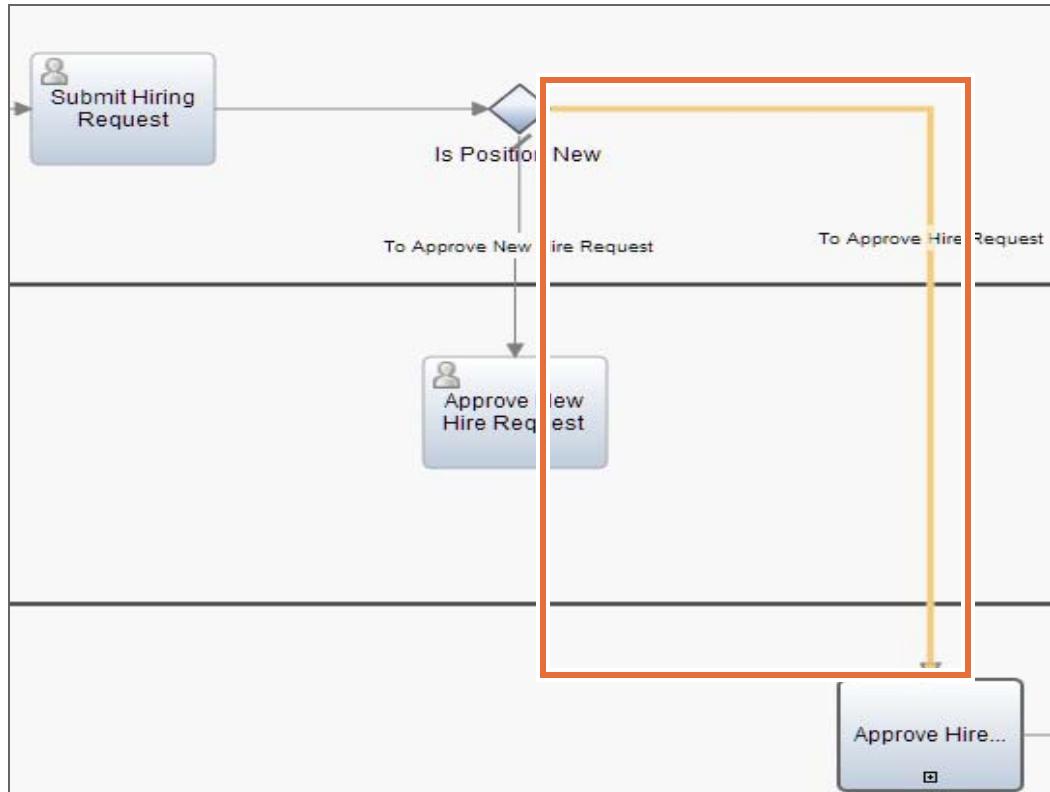
__ f. Lasso the **Is Position New** gateway and the **Approve New Hire Request** nodes.



- g. Although the sequence flow is highlighted, Process Designer will only align the nodes. After highlighting the nodes (depicted by yellow halo), right-click the mouse on one of the nodes to bring up the Align nodes menu. Select **Center** from the Align nodes menu.



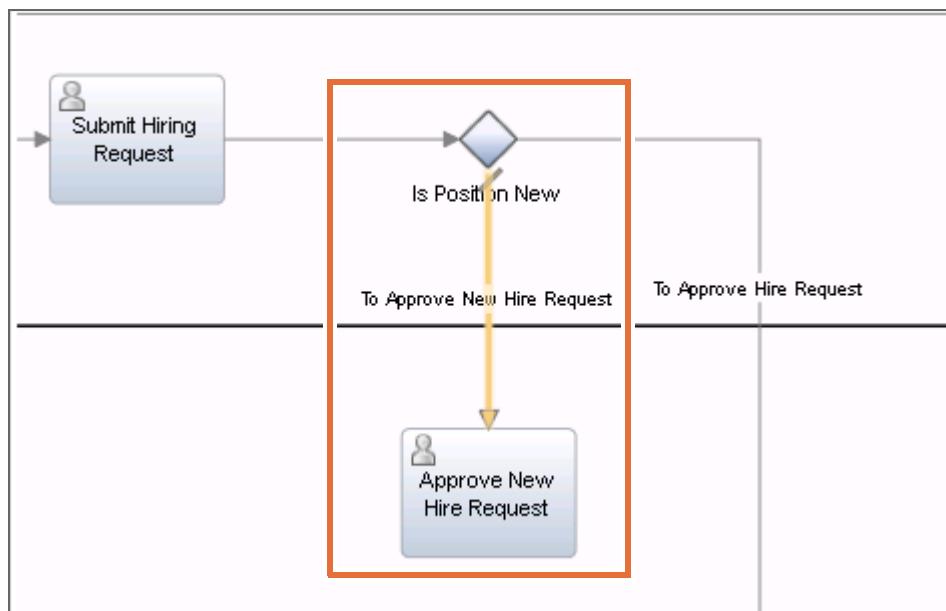
- h. Connect the **Is Position New** gateway to the top of **Approve Hire Request**.



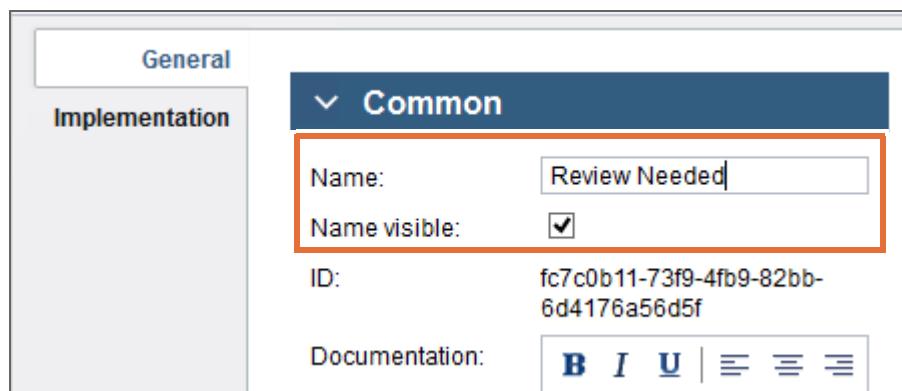
**Note**

Your sequence flows do not have to exactly match the exercise diagrams in this lab. The sequence flows change after implementation of the gateways. The flows become default or conditional (represented by a diagonal hash marker) according to the order you draw them, so your flows might vary from the diagrams in the labs. You learn how to set the default flow in a later exercise.

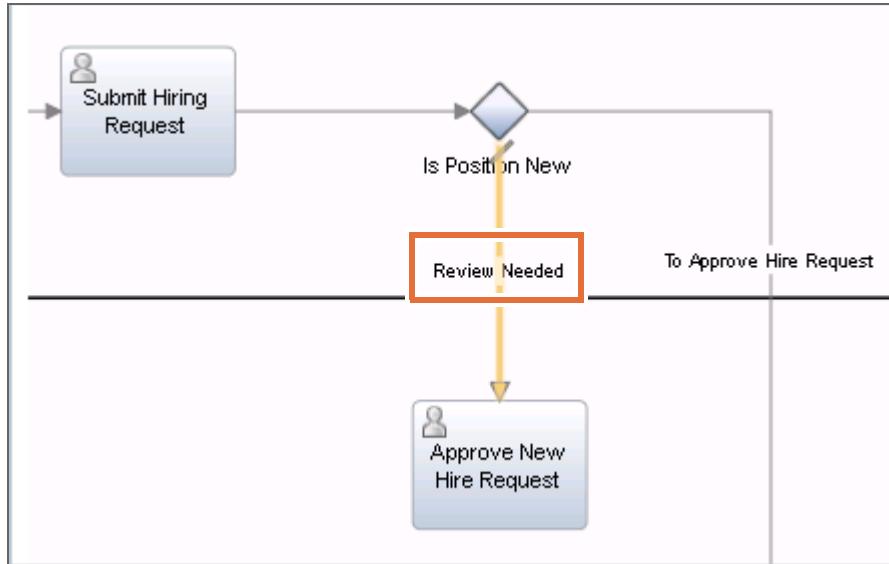
- ___ 3. Label the sequence flows.
 - ___ a. Select the flow between the **Is Position New** gateway and **Approve New Hire Request**.



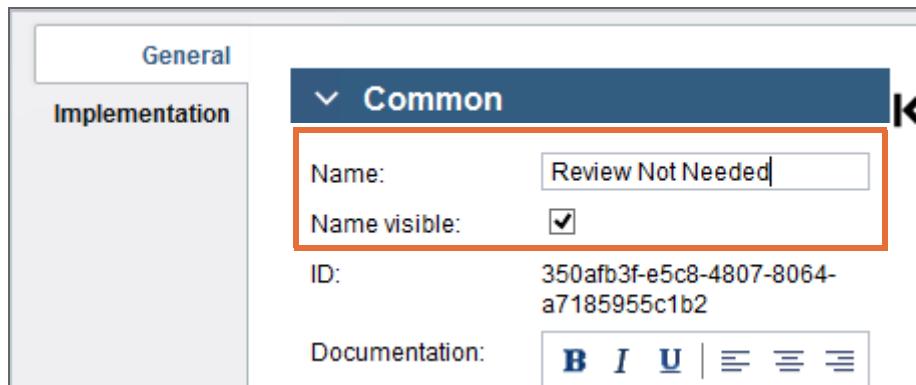
- ___ b. In the **General > Common** section, enter Review Needed as the **Name** and verify **Name visible** is selected.



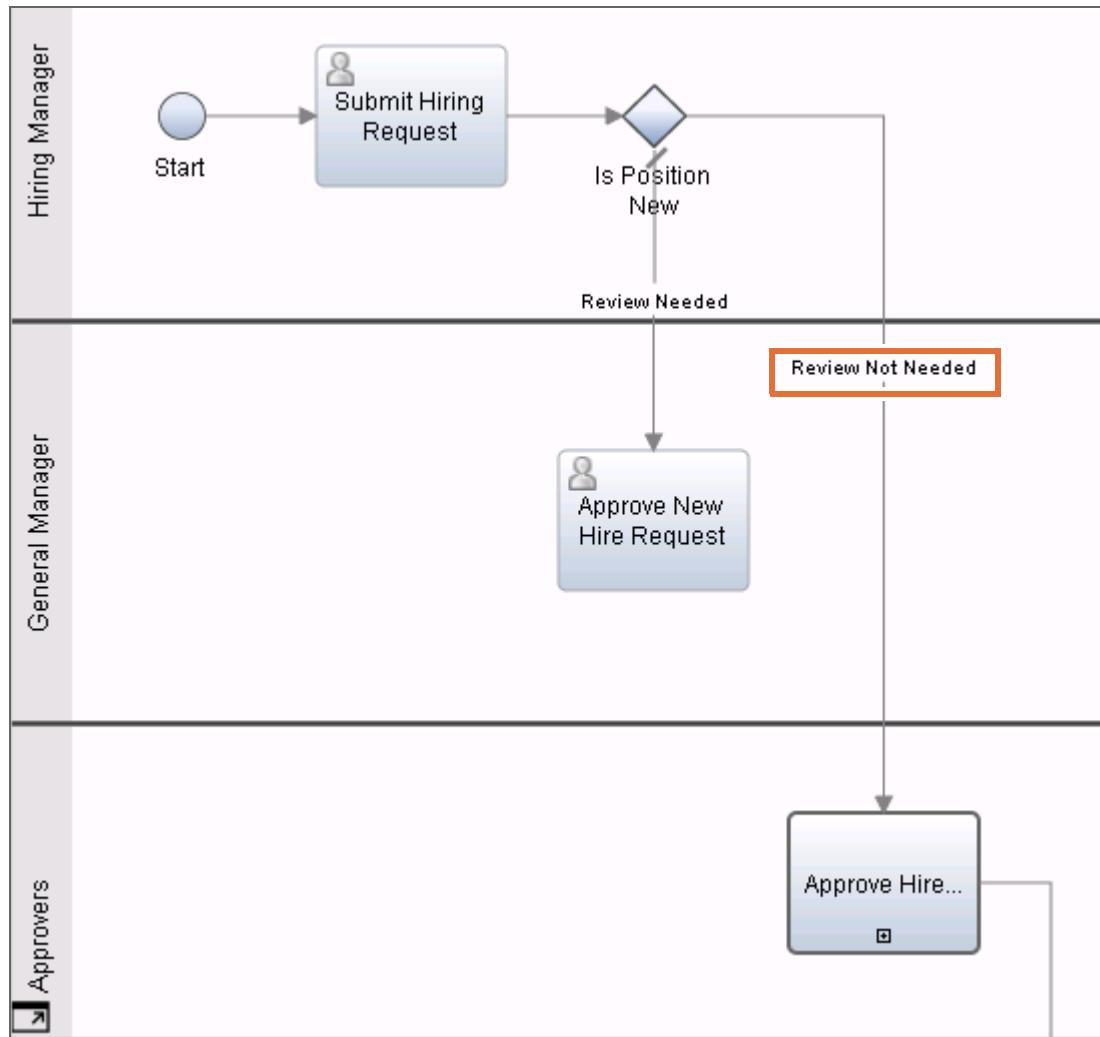
The flow displays the label.



- ___ c. Select the flow between the **Is Position New** gateway and **Approve Hire Request**.
- ___ d. In the **General > Common** section, enter Review Not Needed as the Name and verify **Name visible** is selected.

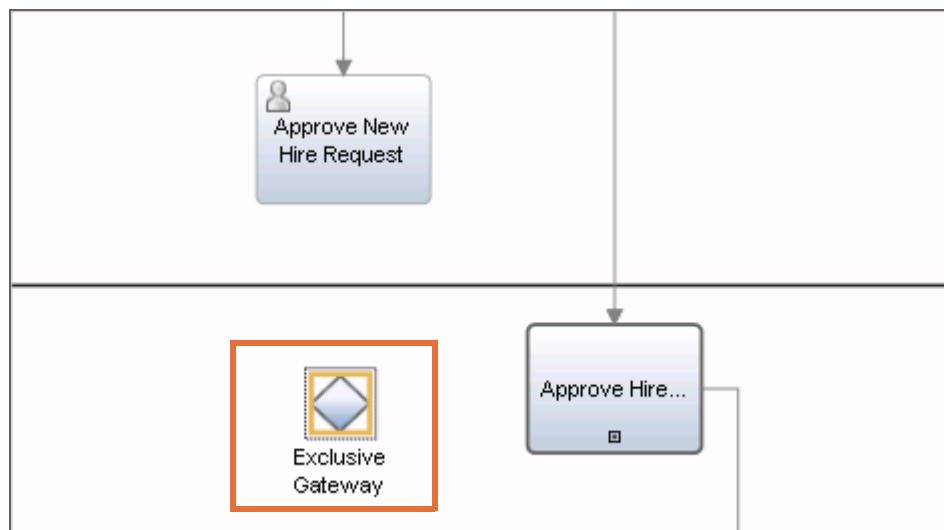


- e. Save your changes. The flow displays the label.

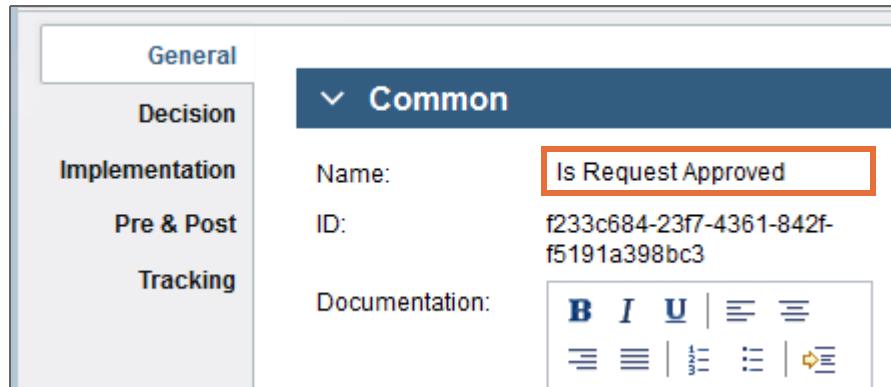


- 4. Create the **Is Request Approved** gateway

- a. Drag a **Gateway** from the palette onto the canvas to the left of the **Approve Hire Request** activity.



- ___ b. Select the **Exclusive Gateway** and in the **General > Common** section, enter **Is Request Approved** as the Name.

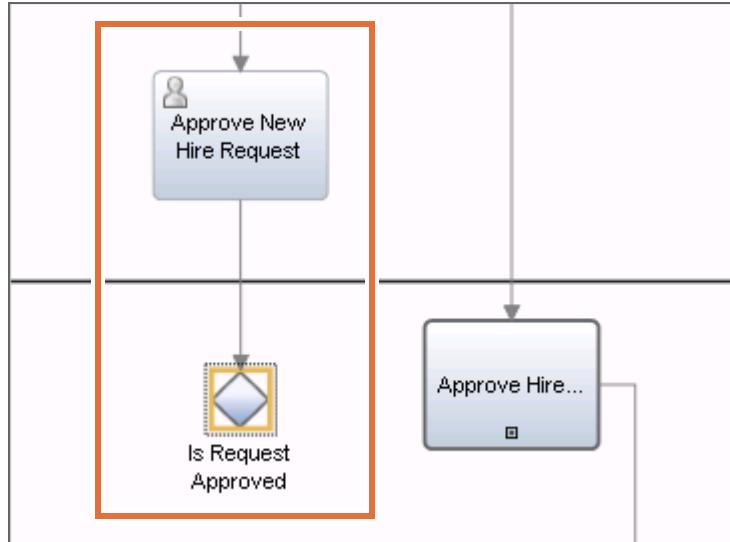


- ___ c. In the **General > Behavior** section, verify **Exclusive Gateway** as the Gateway type.



5. Connect and label the sequence flows.

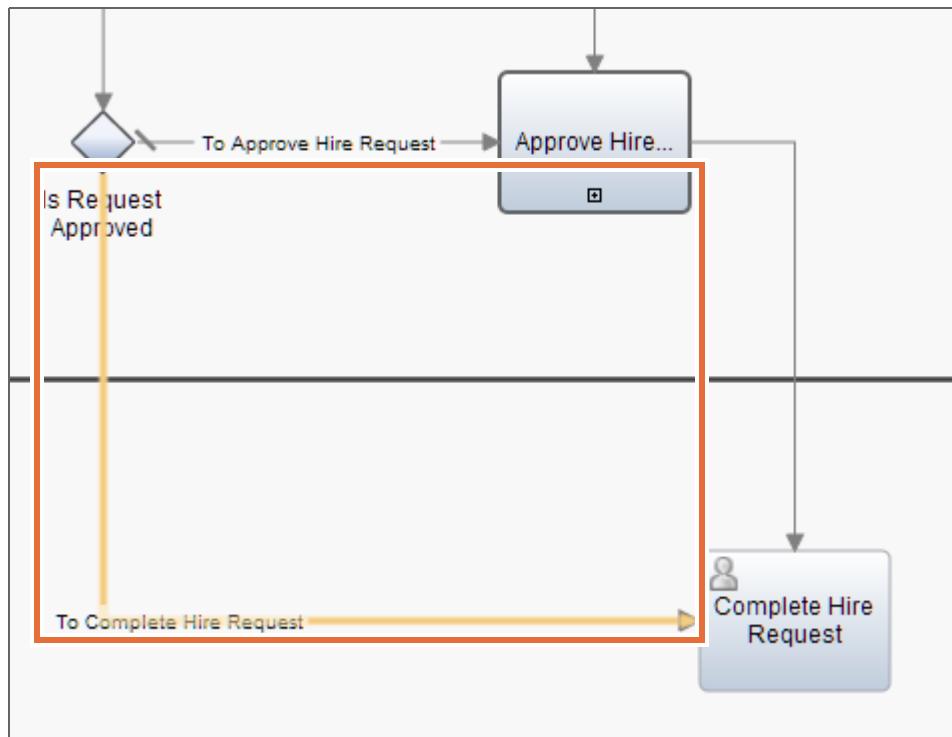
- ___ a. Connect the **Approve New Hire Request** to the top of the **Is Request Approved** gateway.



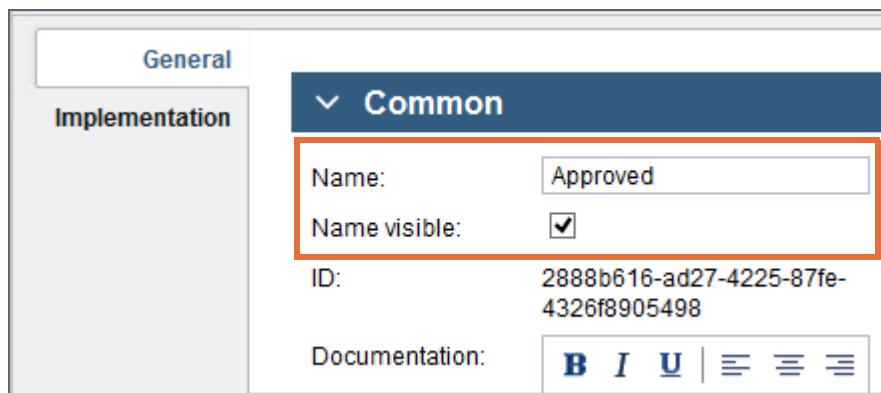
- __ b. Connect the **Is Request Approved** gateway to the left anchor point of **Approve Hire Request**.



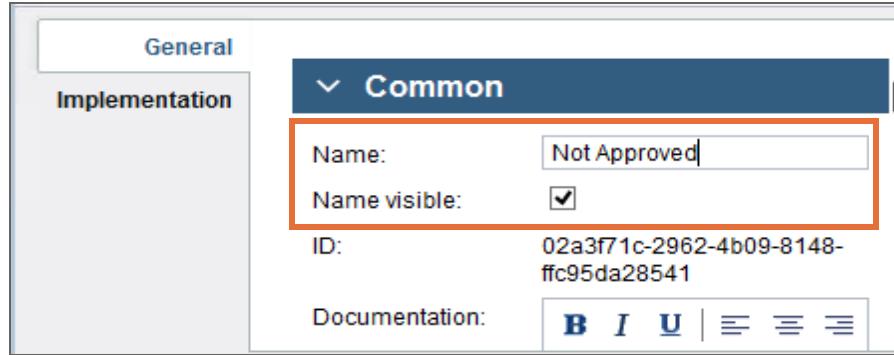
- __ c. Connect the **Is Request Approved** gateway to the **Complete Hire Request** activity.



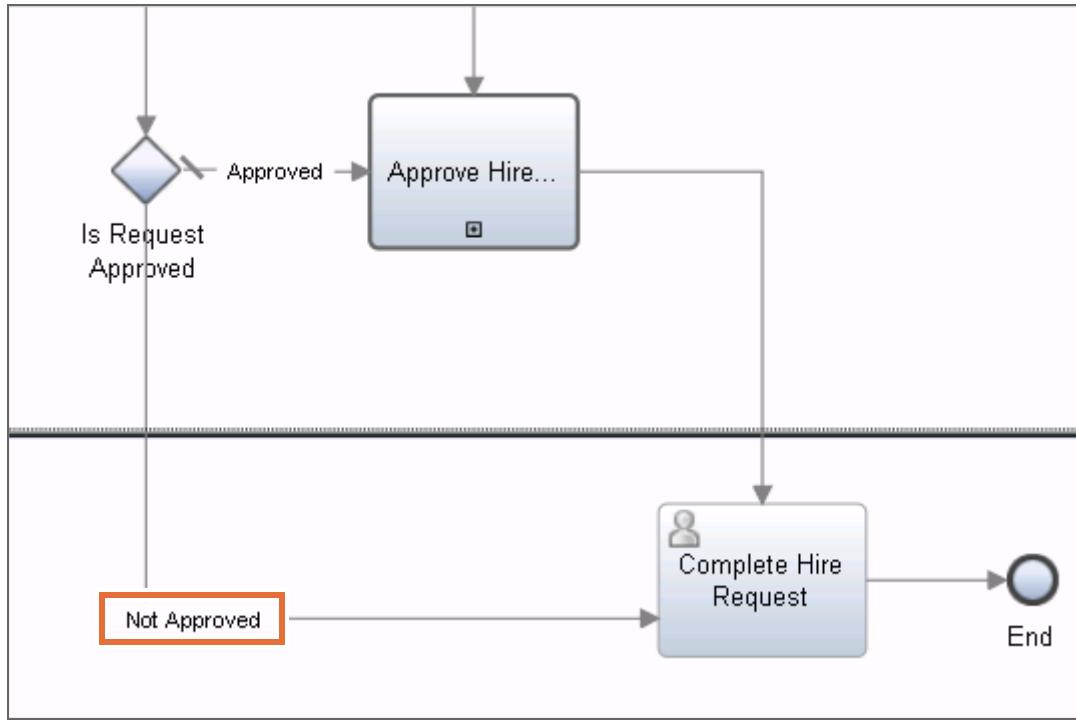
- __ d. Select the flow between the **Is Request Approved** gateway and the **Approve Hire Request** linked process.
 __ e. Rename the default name to **Approved** and verify **Name visible** is selected.



- ___ f. Select the flow between the **Is Request Approved** gateway and **Complete Hire Request**.
- ___ g. Rename the default name to `Not Approved` and verify **Name visible** is selected.



The flow displays the label:

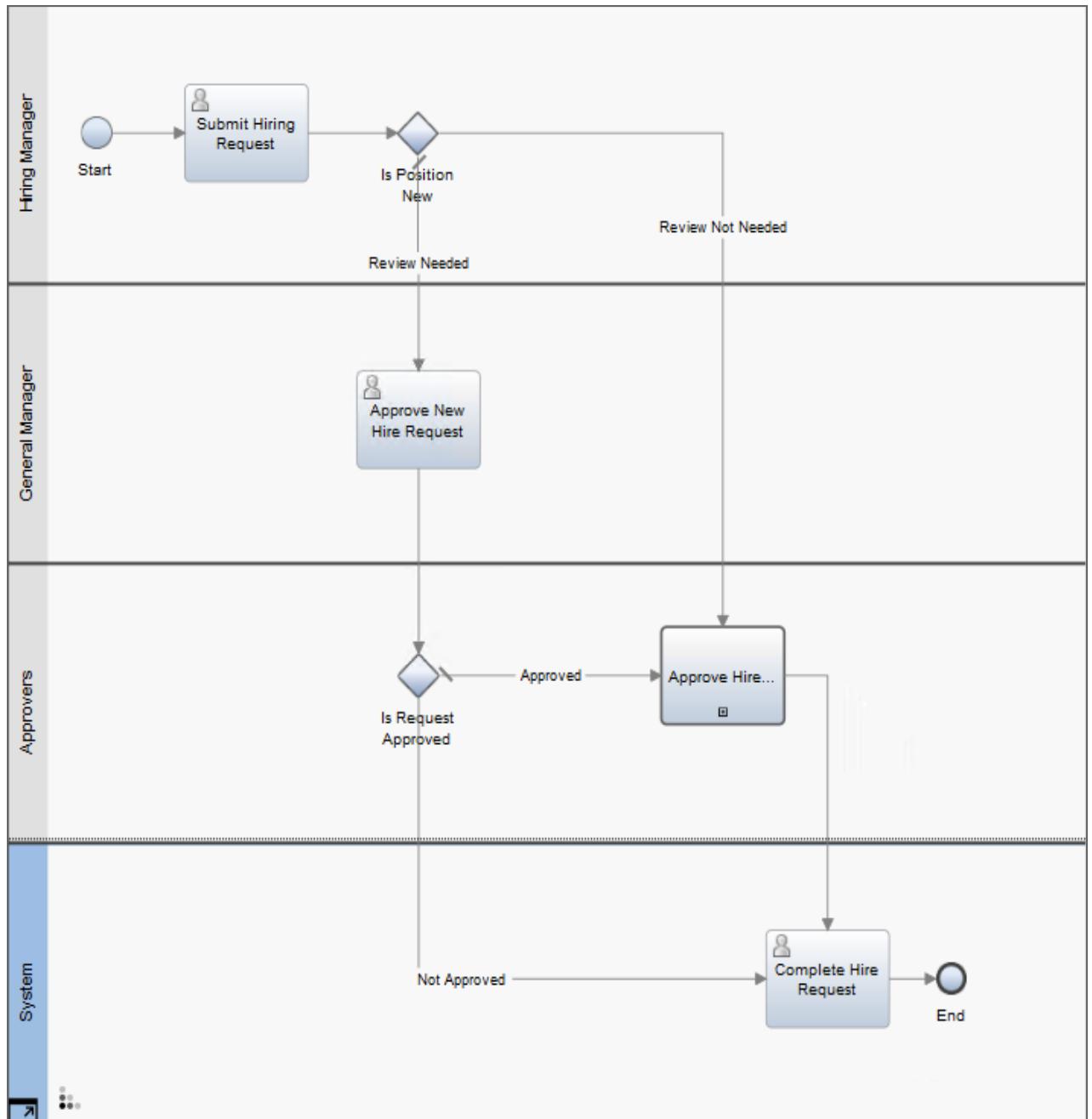


- ___ h. Save your work.
- ___ 6. Verify the process model

If you have a colleague or others nearby, see whether you can explain your new model to them to simulate a playback. Being able to clearly communicate the process is key when implementing business process management.
- ___ a. Use the zoom slider at the upper right of the canvas to view the whole process if it doesn't fit on the screen.



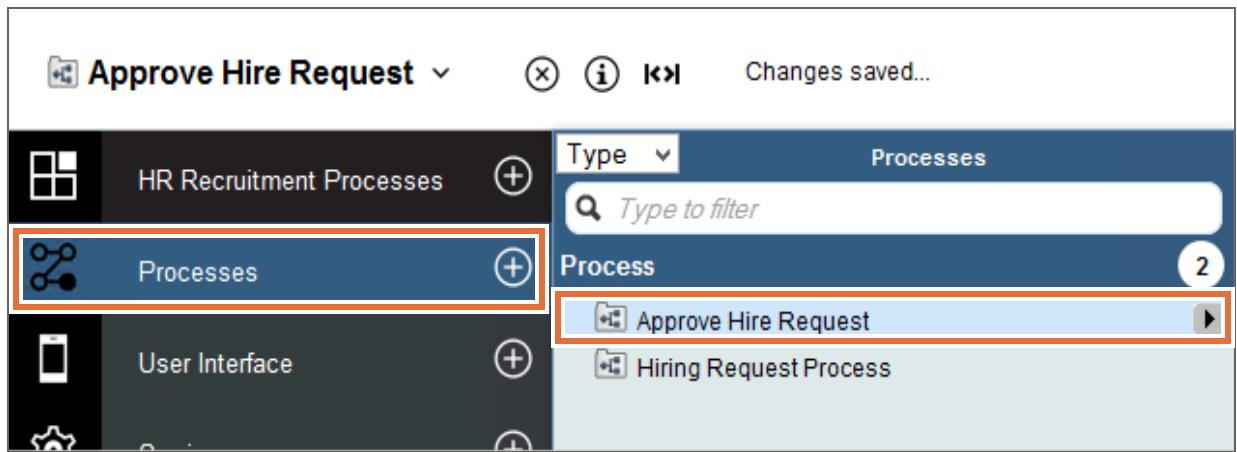
- __ b. Ensure that the **Hiring Request Process** process looks similar to the screen capture before moving on:



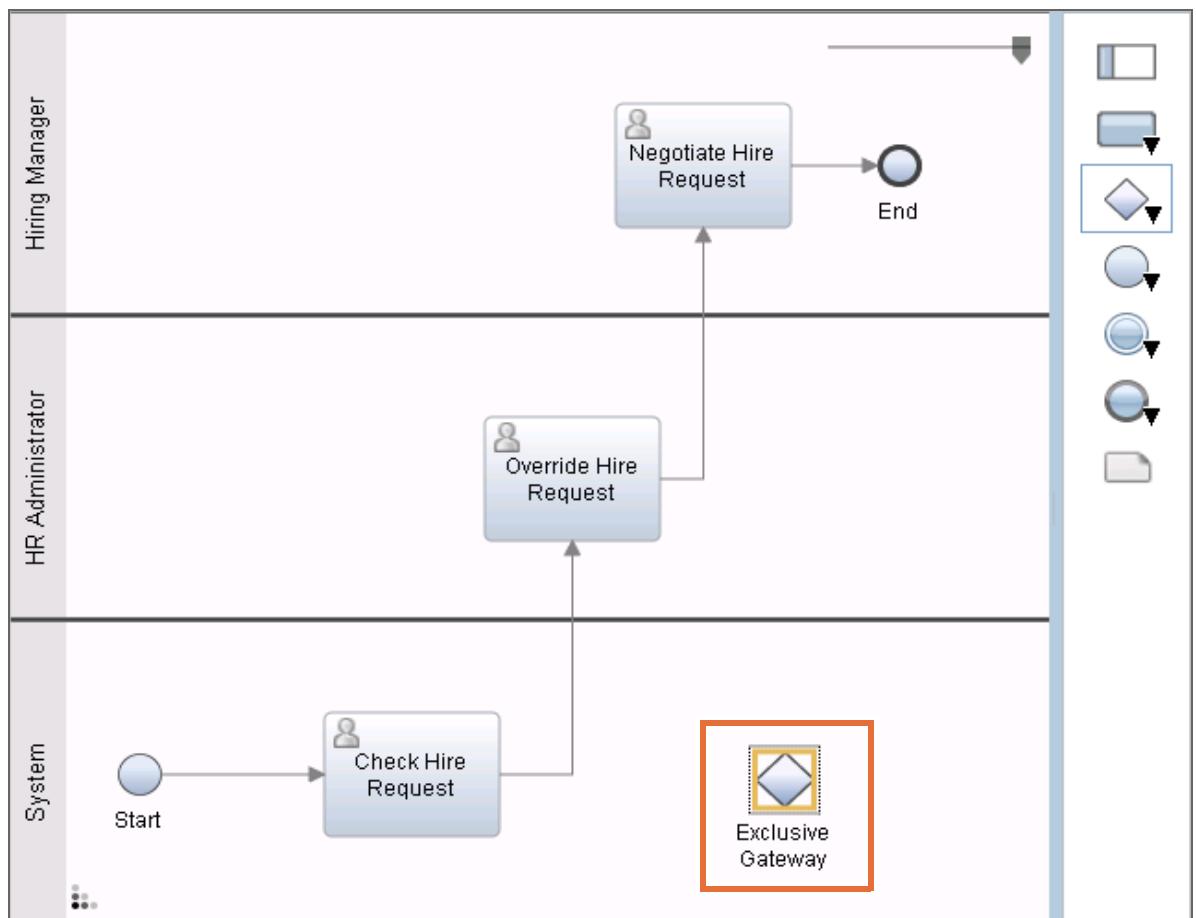
Part 2: Create gateways for the linked process

Add gateways to the nested process Approve Hire Request.

- 1. Create the **Is Salary Compliant** gateway.
- a. Click **Processes** in the library and click **Approve Hire Request** from the menu.

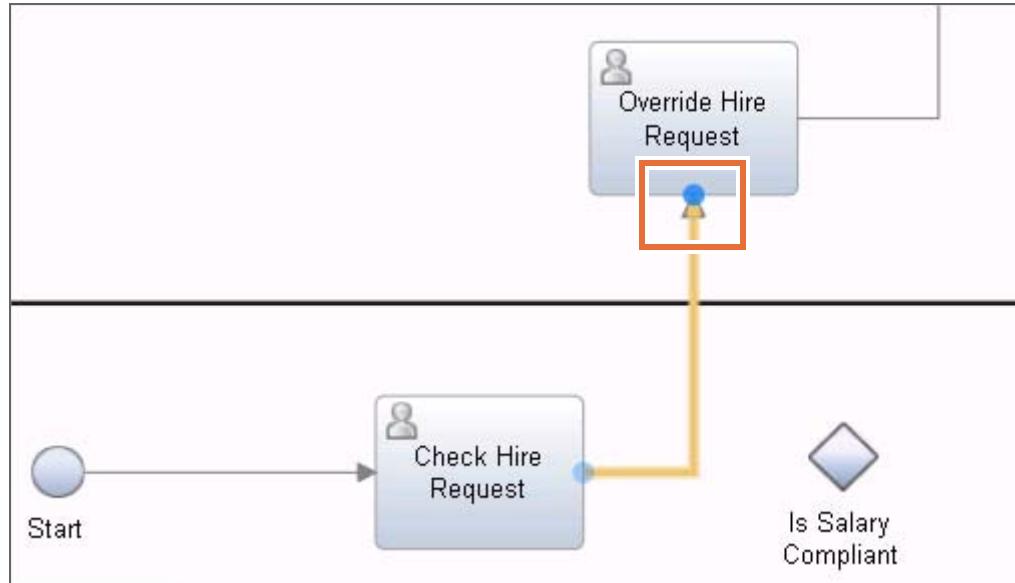


- b. Drag a **Gateway** from the palette onto the canvas to the right of the **Check Hire Request** activity.



- c. Rename the gateway to: **Is Salary Compliant**

- ___ d. Verify that the Gateway type is **Exclusive Gateway**.
2. Connect and label sequence flows.
- ___ a. Select the sequence flow between **Check Hire Request** and **Override Hire Request**.
- ___ b. Hover over the tip of the sequence flow to see a blue point.

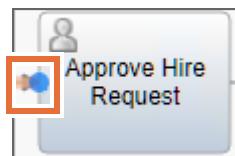


- ___ c. Select the blue point and drag the flow to connect **Check Hire Request** to the **Is Salary Compliant** gateway.



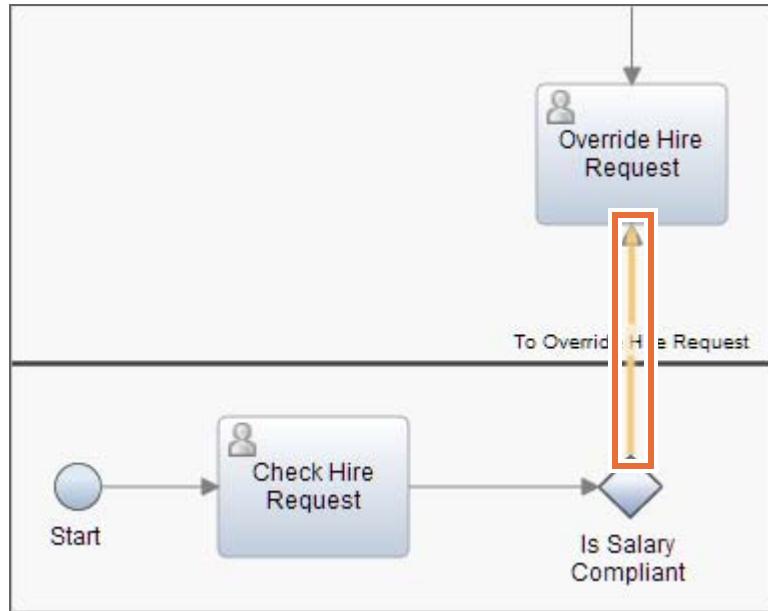
Hint

You can drag sequence endpoints from one activity to the other. To do this, highlight the endpoint until it displays as bright blue. Then, drag the endpoint to another activity and release.



It can be tricky to highlight the endpoint correctly. As you become familiar with the Process Designer tool, you can choose the methods that work best for you. For instance, if it becomes too cumbersome to drag sequence lines from one activity to another, you can delete them and rebuild them.

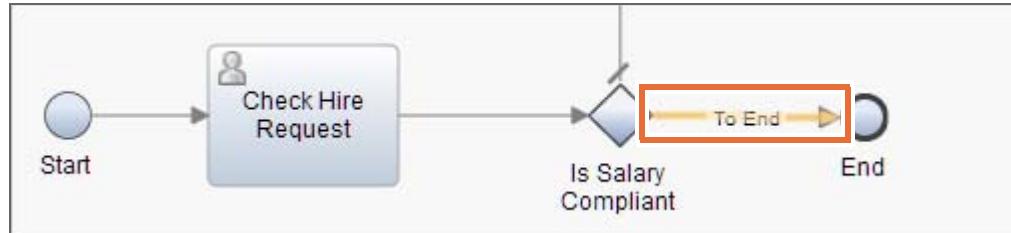
- ___ d. Connect **Is Salary Compliant** to the bottom of **Override Hire Request**.



- ___ e. Remove the existing sequence flow between **Negotiate Hire Request** and the **End** event by highlighting the flow then pressing the Delete button on the keyboard.
- ___ f. Move the **End** event from the top of the diagram and place it to the right of **Is Salary Compliant**.

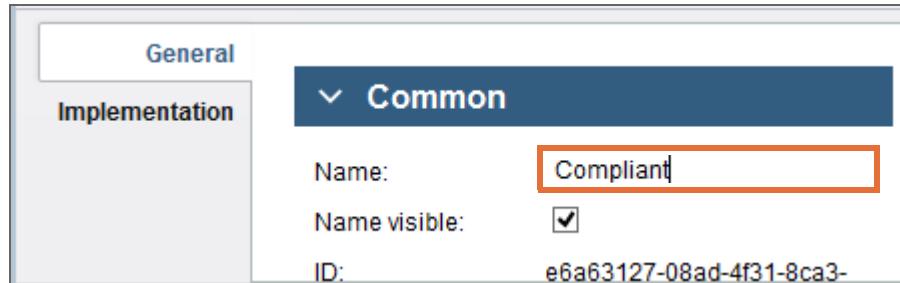


- ___ g. Connect **Is Salary Compliant** to the left anchor point of the **End** event.

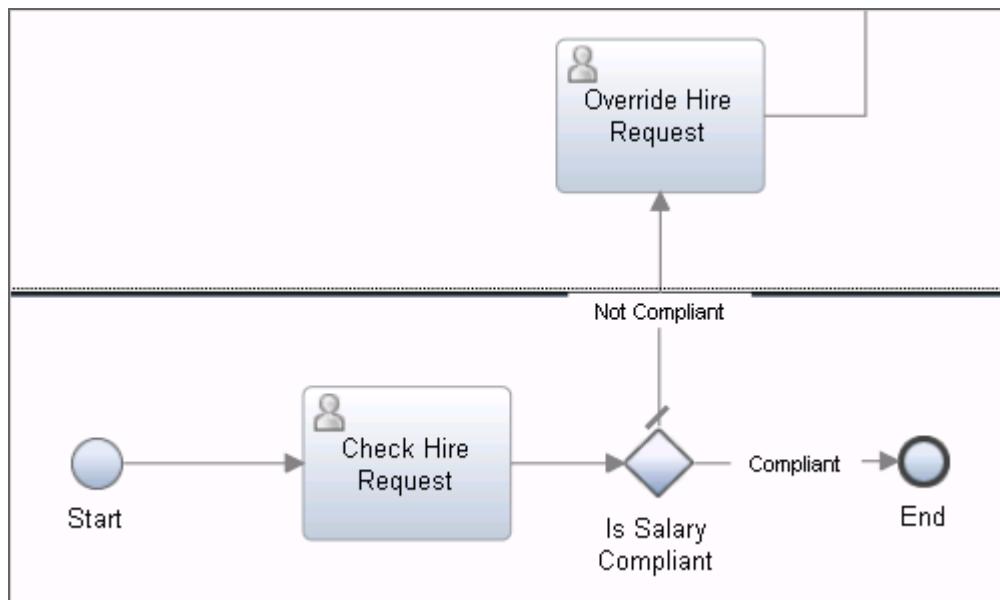


- ___ h. Select the flow between the **Is Salary Compliant** gateway and **Override Hire Request**.
- ___ i. Rename the flow: Not Compliant
- ___ j. Select the flow between the **Is Salary Compliant** gateway and the **End** event.

- __ k. In the **General > Common** section, rename the flow: Compliant



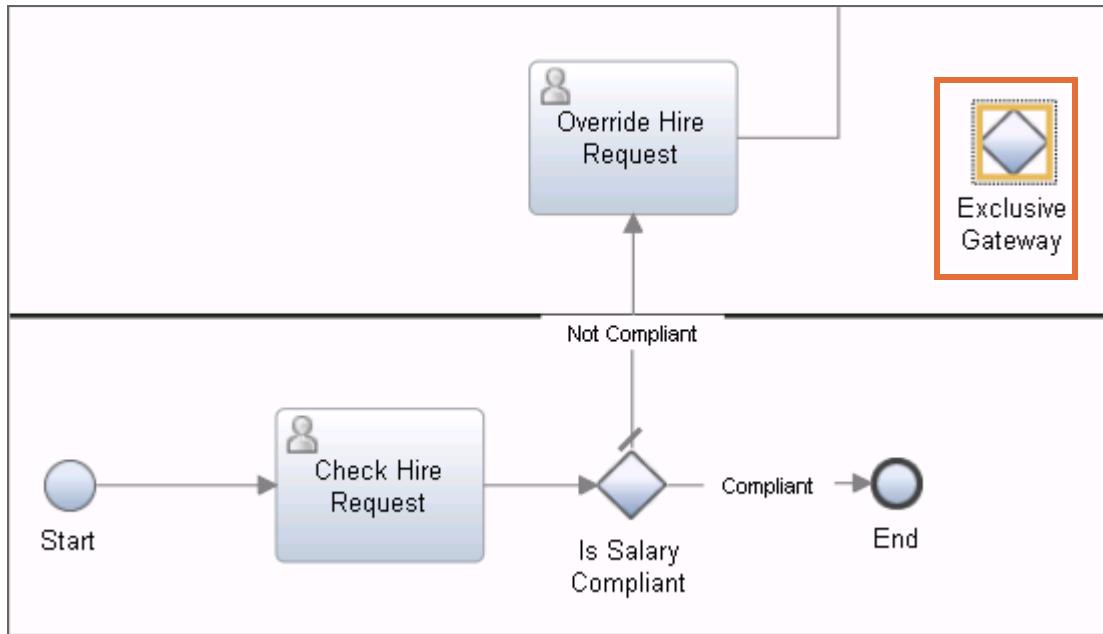
- __ l. Verify your process. Your process should look similar to this screen capture.



- __ m. Save your work.

It is okay if the default sequence flows do not match. These are implemented later.

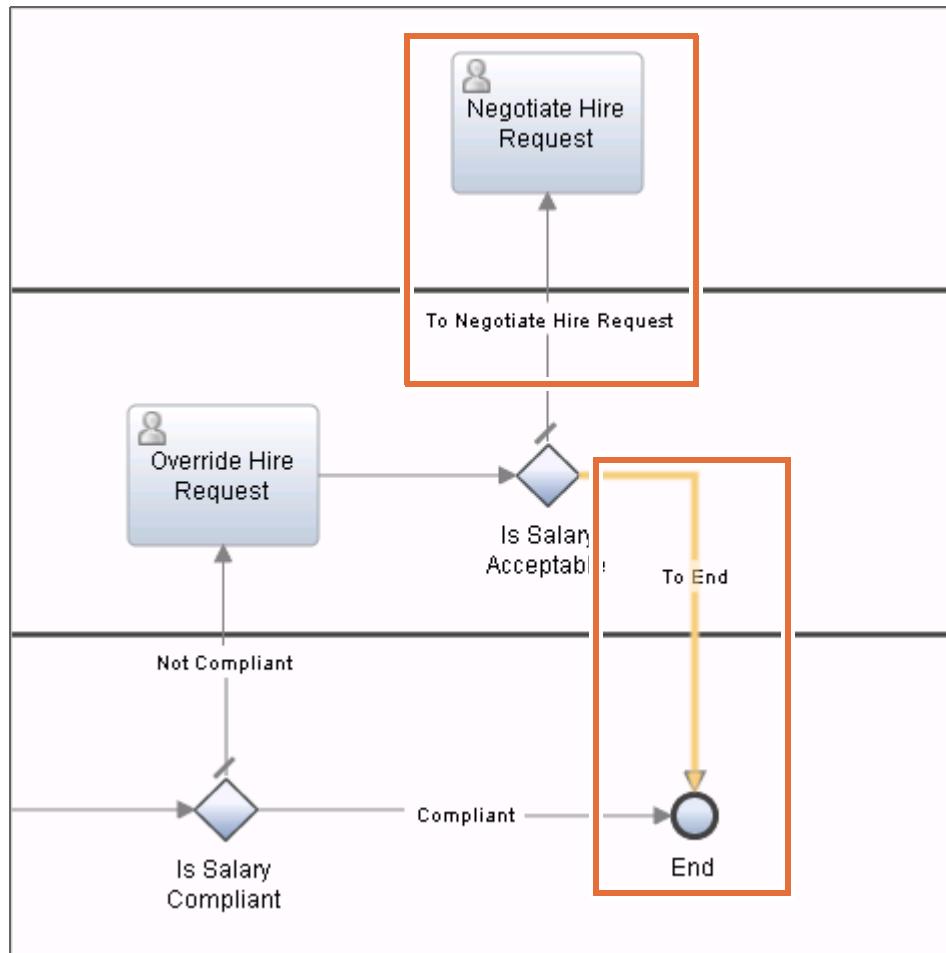
- ___ 3. Create the **Is Salary Acceptable** gateway.
- ___ a. Drag a **Gateway** from the palette onto the canvas to the right of the **Override Hire Request** activity.



- ___ b. Set the gateway name to: **Is Salary Acceptable**
- ___ c. Delete the flow between the **Override Hire Request** and the **Negotiate Hire Request** activities.
- ___ d. Connect the **Override Hire Request** to the left anchor of the **Is Salary Acceptable** gateway.



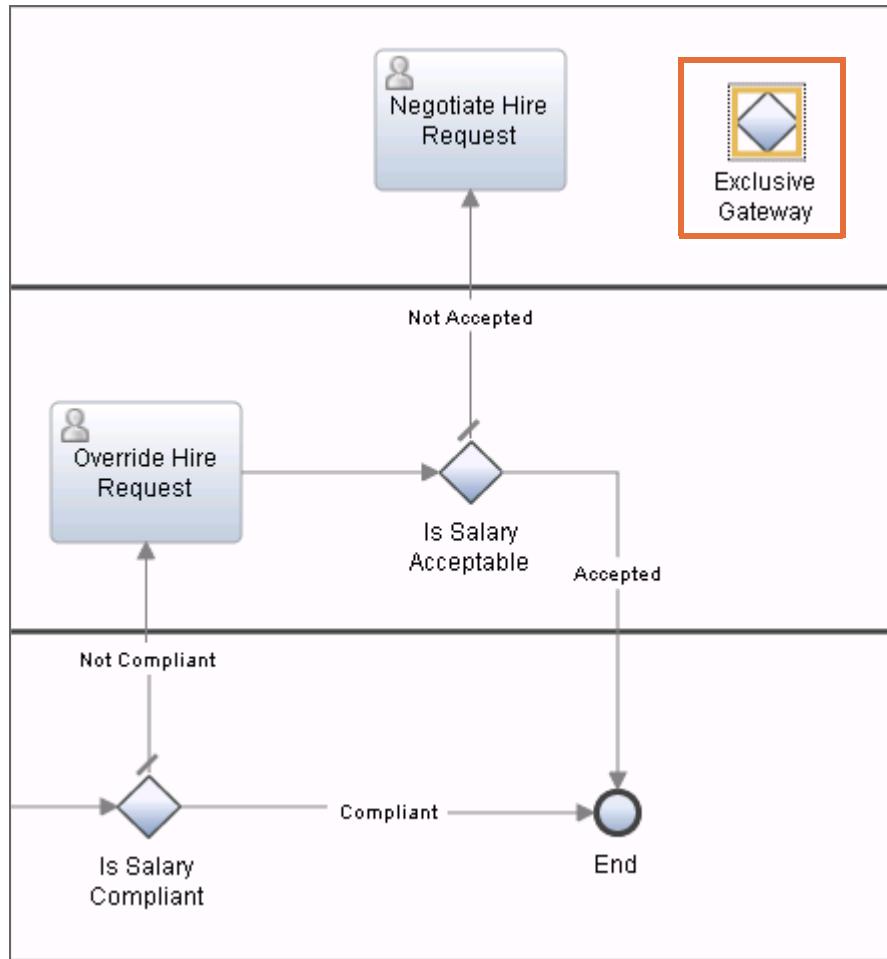
- ___ e. Connect the **Is Salary Acceptable** gateway to the bottom anchor of **Negotiate Hire Request** and to the top anchor of the **End** event.



- ___ f. Rename the flow between the **Is Salary Acceptable** gateway and **Negotiate Hire Request** as: **Not Accepted**
 ___ g. Rename the flow between the **Is Salary Acceptable** gateway and the end event as: **Accepted**

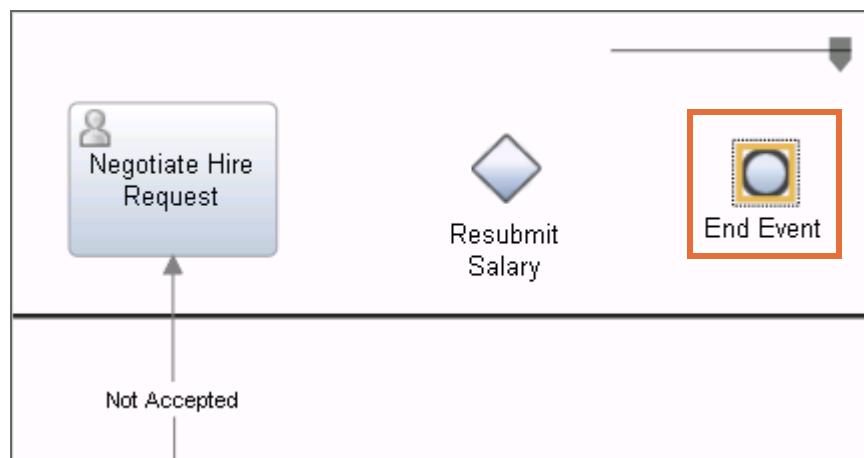
__ 4. Create the **Resubmit Salary** gateway.

__ a. Drag a **Gateway** from the palette onto the canvas to the right of the **Negotiate Hire Request** activity.

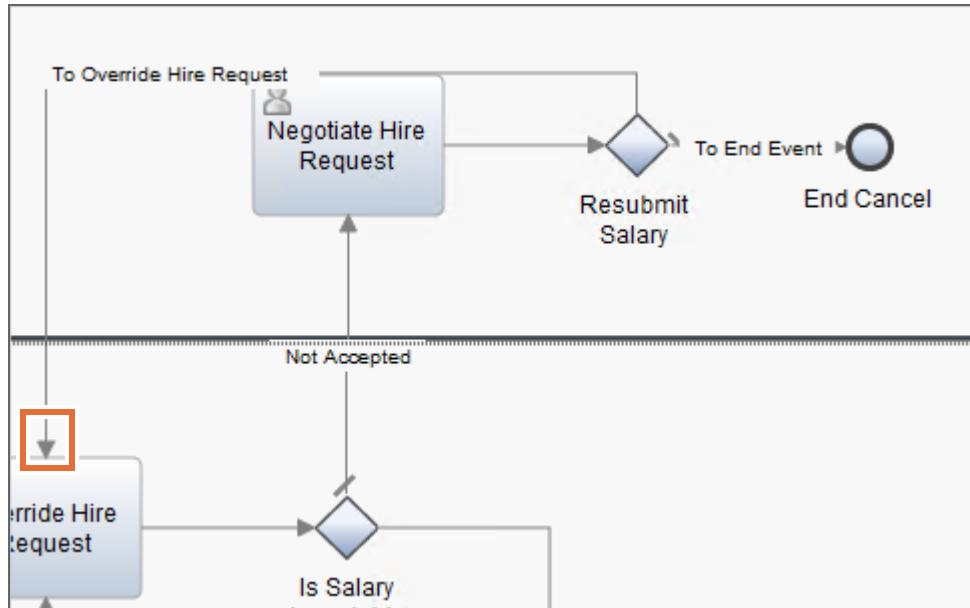


__ b. Rename the gateway to **Resubmit Salary** and verify that the Gateway type is **Exclusive Gateway**.

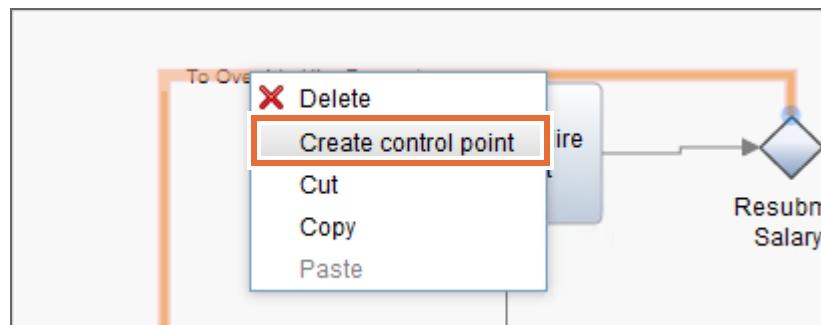
__ c. Drag an **End event** from the palette onto the canvas to the right of the **Resubmit Salary** gateway.



- d. Rename the **End** event as: End Cancel
- e. Connect **Negotiate Hire Request** to the **Resubmit Salary** gateway.
- f. Connect the **Resubmit Salary** gateway to the **End Cancel** event.
- g. Connect the **Resubmit Salary** gateway to the top of **Override Hire Request**.

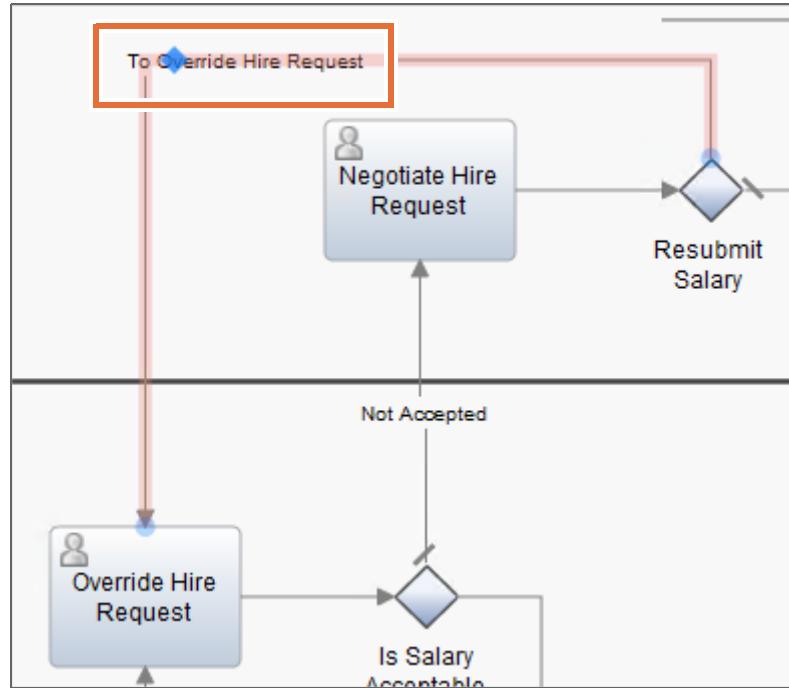


- h. To fix the flow that overlaps the Negotiate Hire Request activity, select the **To Override Hire Request** flow and right-click the flow. Click **Create control point**.

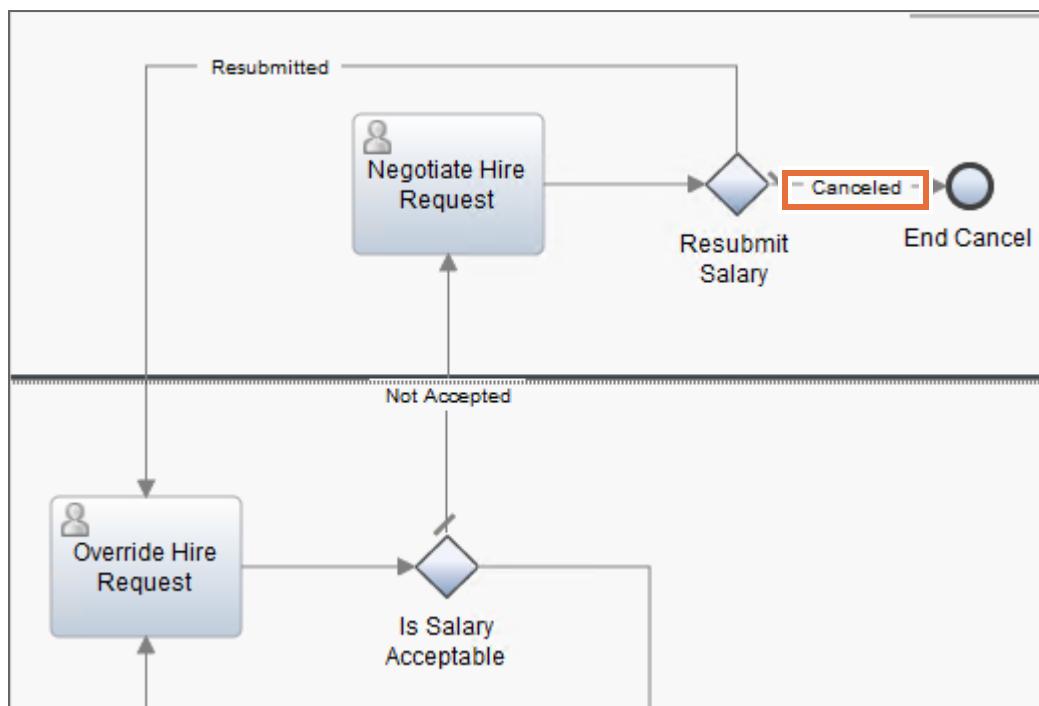


You use control points to manipulate the sequence flow path between two connection points. This helps to make the diagram more readable.

- __ i. Drag the blue diamond to move the flow above the activity.



- __ j. Name the flow between the **Resubmit Salary** gateway and **Override Hire Request** as Resubmitted.
 __ k. Label the flow between the **Resubmit Salary** gateway and the **End** event as Canceled.



- __ l. Straighten the sequence flows as necessary.
 __ m. Save your work.

**Optional**

You completed part 2 of the 5, so you are now approximately half done with this exercise. Now would be a good time to take a 10-minute break.

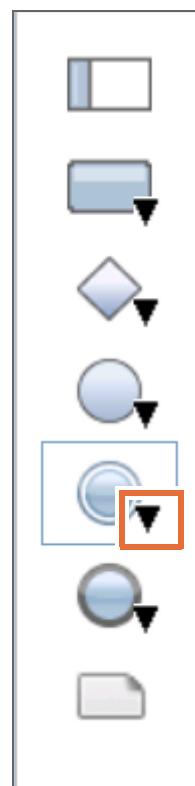
Part 3: Modeling timer intermediate events

Examine the process requirements and determine where an escalation exists.

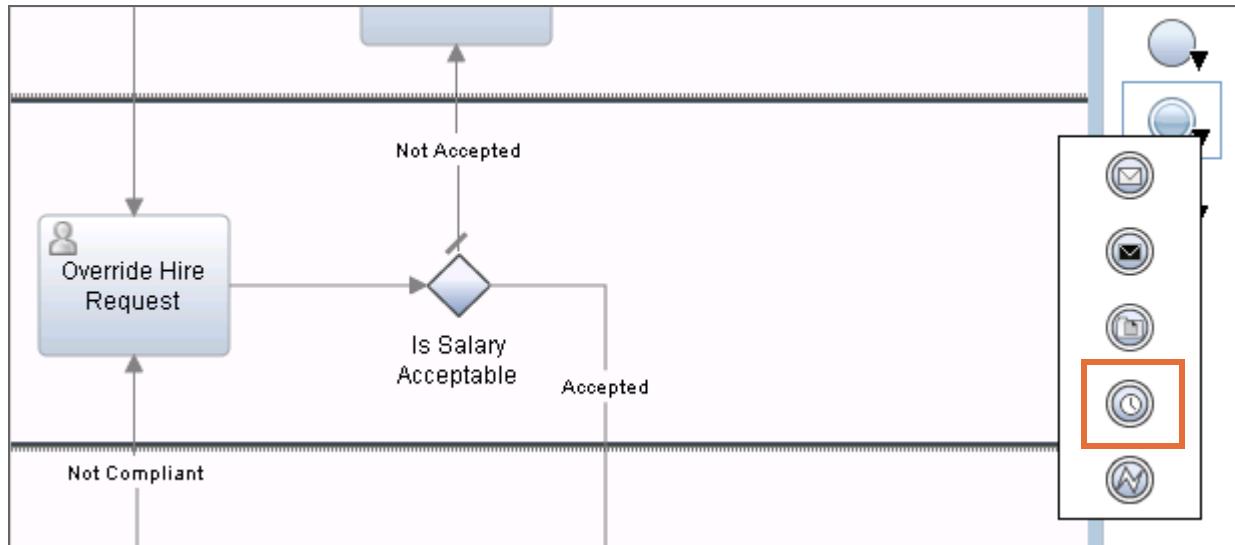
Refer to the core requirements for an escalation in section 2.10 that is listed in Exercise 2. The HR Administrator has 4 hours to complete the review. If the review is not completed within 4 hours, an email is sent to the HR Administrator notifying the HR Administrator of the missed deadline.

In this part of the exercise, you use the timer implementation option to model an escalation path in your processes. Using a timer intermediate event, you can specify a time interval after or before which some activity is conducted. The timer implementation option is available for events that are included in the process flow and events that are attached to an activity.

- ___ 1. Attach a four-hour timer event to the Override Hire Request activity.
 - ___ a. In the **Approve Hire Request** process, select the **drop-down icon** beside the **Intermediate** event.



- ___ b. Select the **Timer event** from the set of intermediate events.

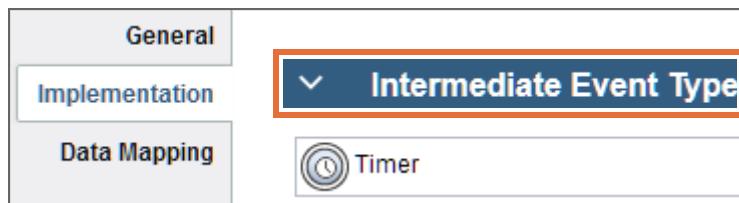


- ___ c. Drag the **Timer event** from the palette onto the lower-left anchor of the **Override Hire Request** activity.



Important

If **Boundary Event Type** is not displayed and you see **Intermediate Event Type** in its place, you have not successfully anchored the Timer Event to the boundary of the activity.

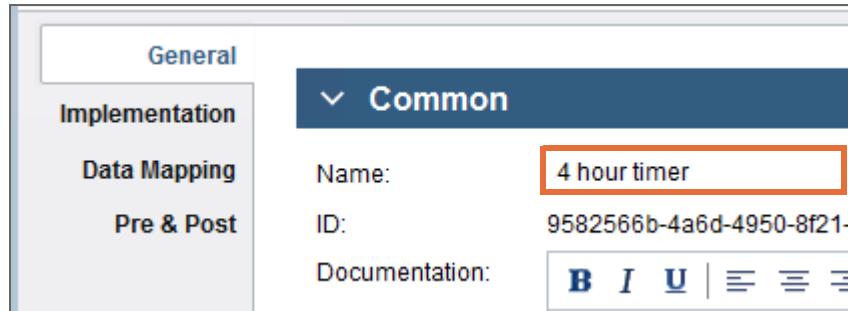


Sometimes attaching the timer event to the activity can be tricky. Look for the anchor points to appear on the activity while the mouse is hovering over the activity, then drag the timer to one of the anchor points. You see the green plus if you hover over one of the anchor points.

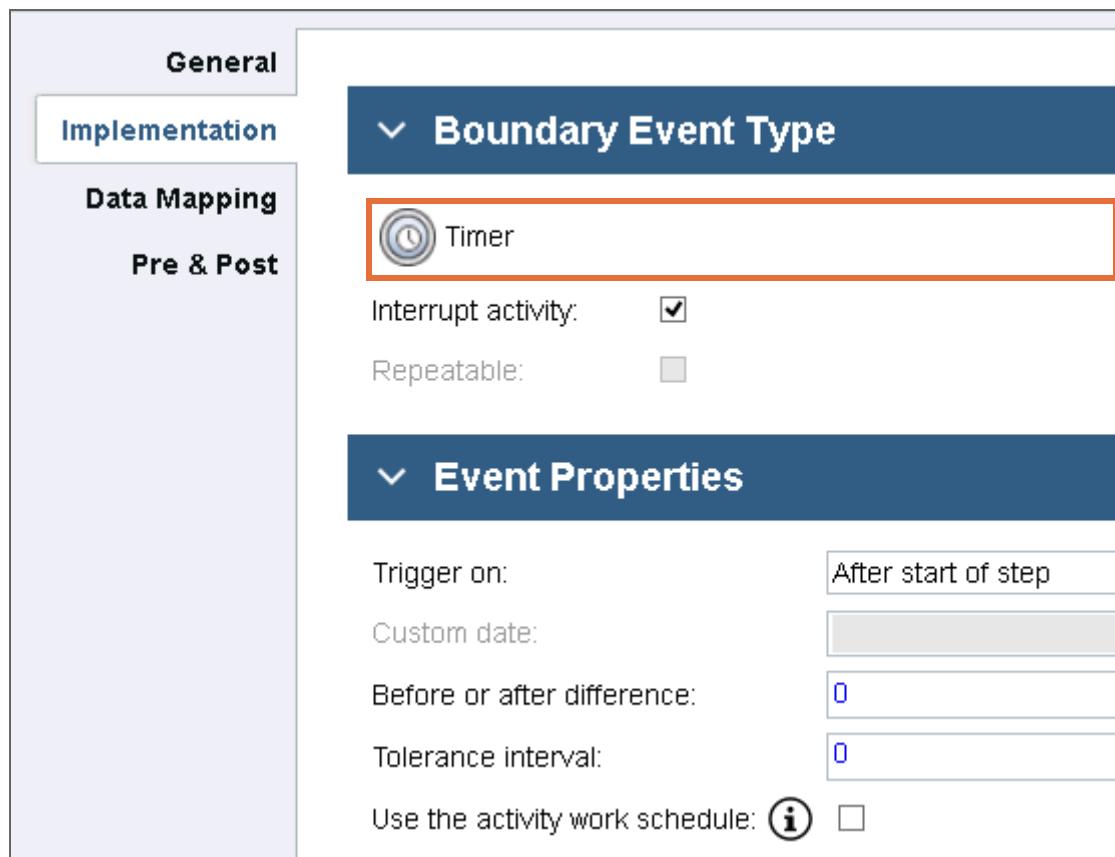
For this exercise, make sure that you drag the timer event onto the activity instead of outside of the activity. In this case, you want a boundary or attached intermediate event instead of a sequence flow intermediate event. If you correctly create an attached timer event, the name for the attached intermediate event is also changed to Boundary Event.

- ___ d. Select the **Timer event**.

- ___ e. In the **General > Common** section, change the name to: 4 hour timer

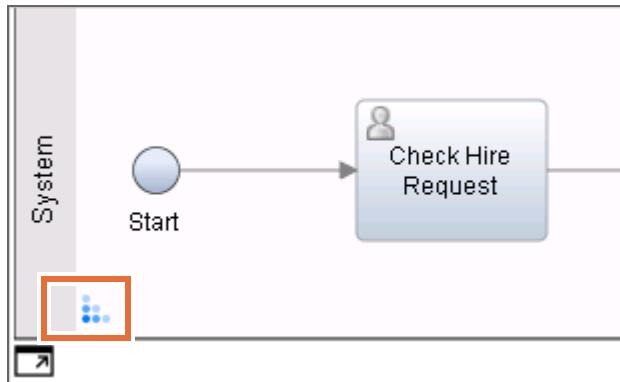


- ___ f. Select the **Implementation** menu. In the **Boundary Event Type** section, make sure that the Timer boundary event is selected. Keep the remaining default properties that were set during implementation.

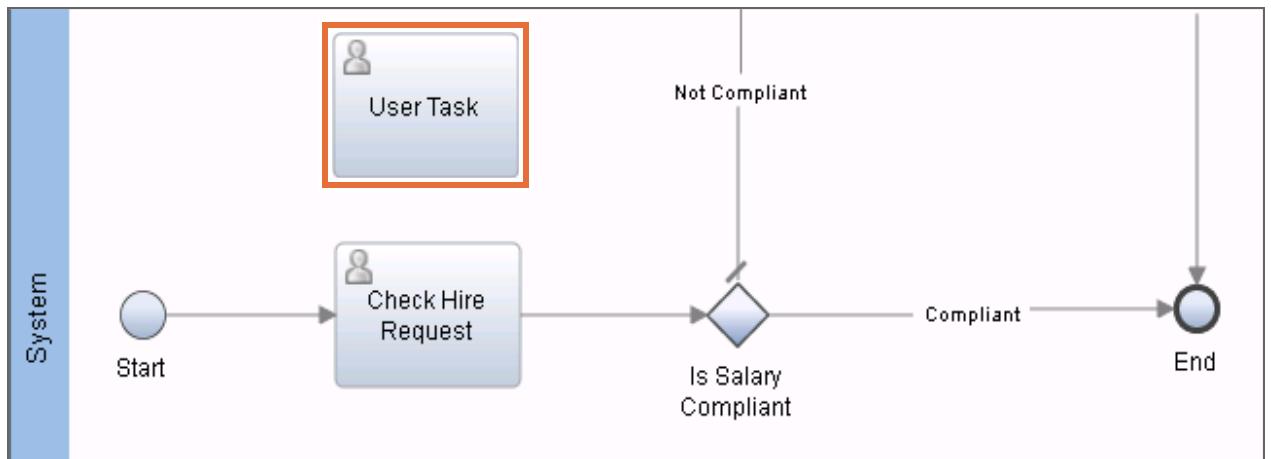


__ 2. Add the Notify HR Administrator activity.

__ a. Resize the **System** lane. To do this, click and hold the left mouse button down, then drag the mouse outside of the lane to increase the height of the lane.

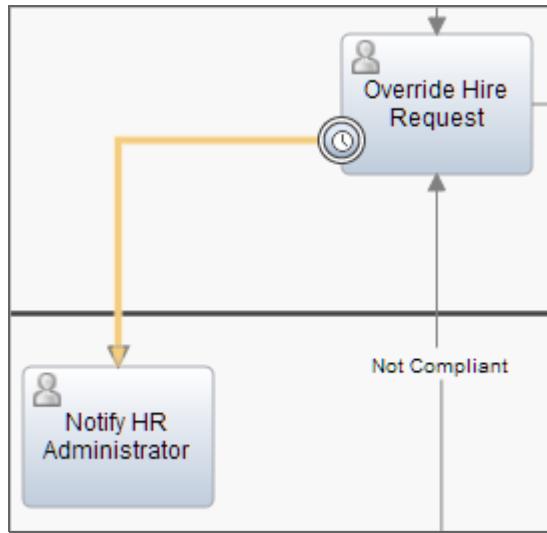


- __ b. Drag a box around the activities, events, and the gateway in the System lane (lasso), and move them down together to make space for this new activity. Once items are highlighted, you can also use the arrow keys on the keyboard to move them.
- __ c. Expand the Activity icon in the palette and drag **User Task** from the palette to the **System** lane. Place it above the **Check Hire Request** activity.

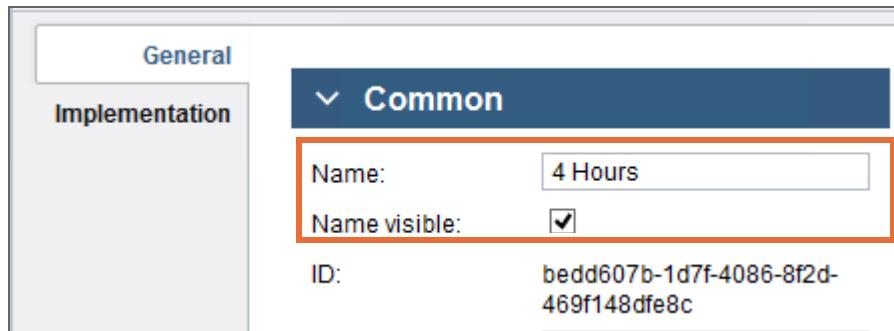


- __ d. Rename the activity to: Notify HR Administrator

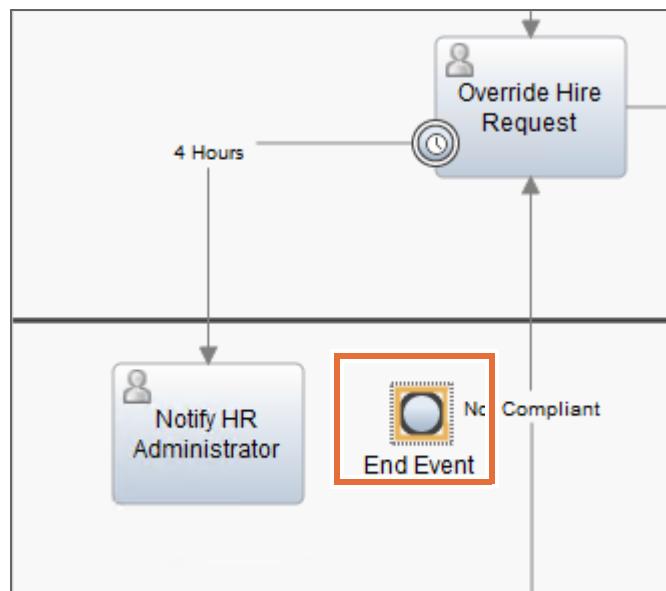
- ___ e. Connect the **Timer Intermediate Event** to the **Notify HR Administrator** activity.



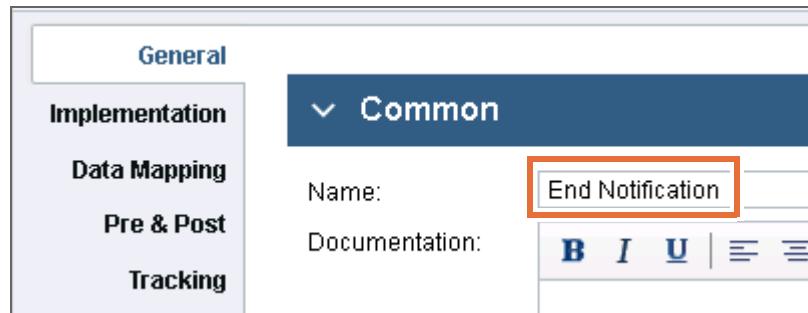
- ___ f. With the sequence line highlighted, name the flow 4 Hours and select **Name visible** in the **General > Common** properties section.



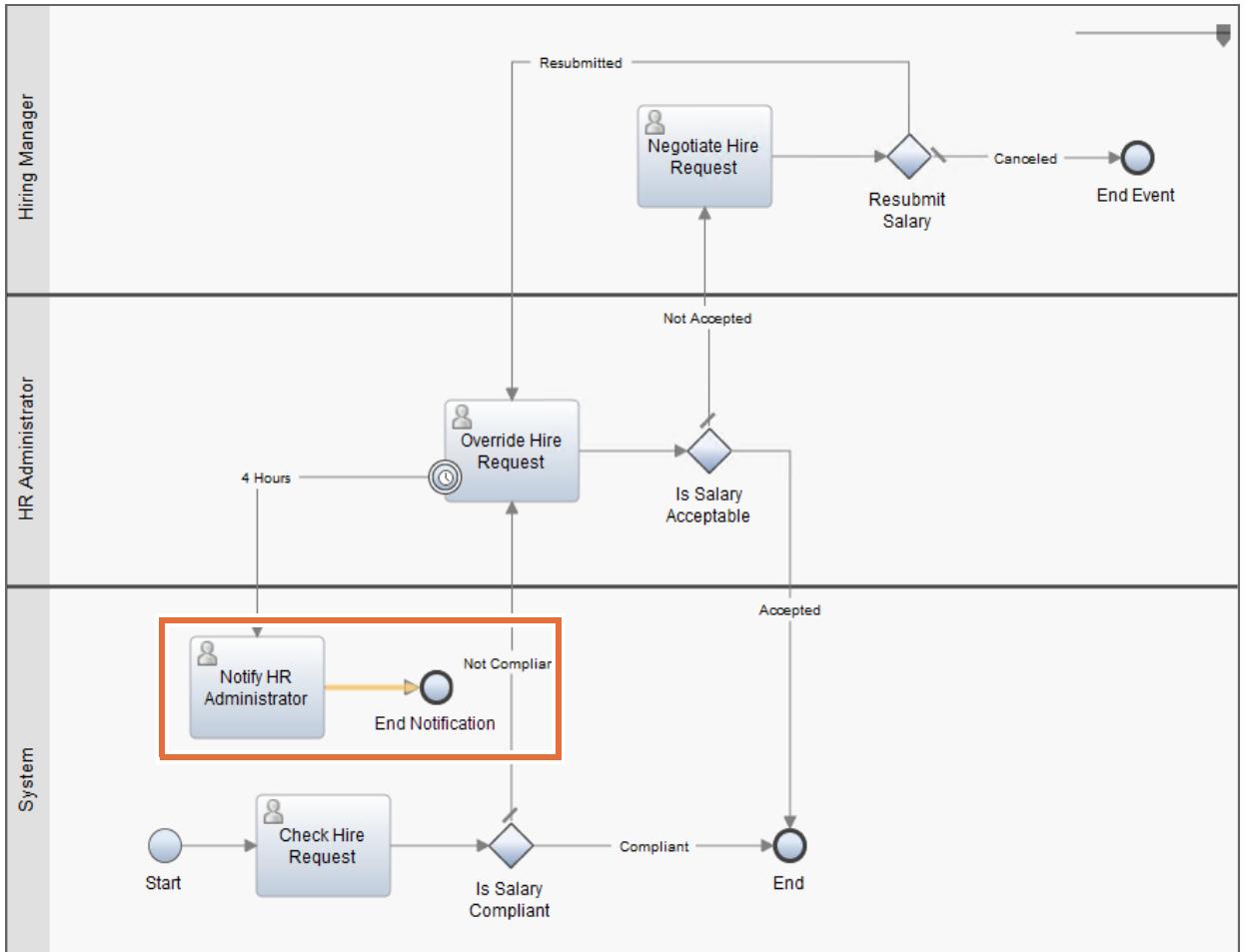
- ___ 3. Add an End Notification event.
- ___ a. Drag an **End Event** from the palette onto the canvas to the right of **Notify HR Administrator**.



- ___ b. Rename the End event to: End Notification



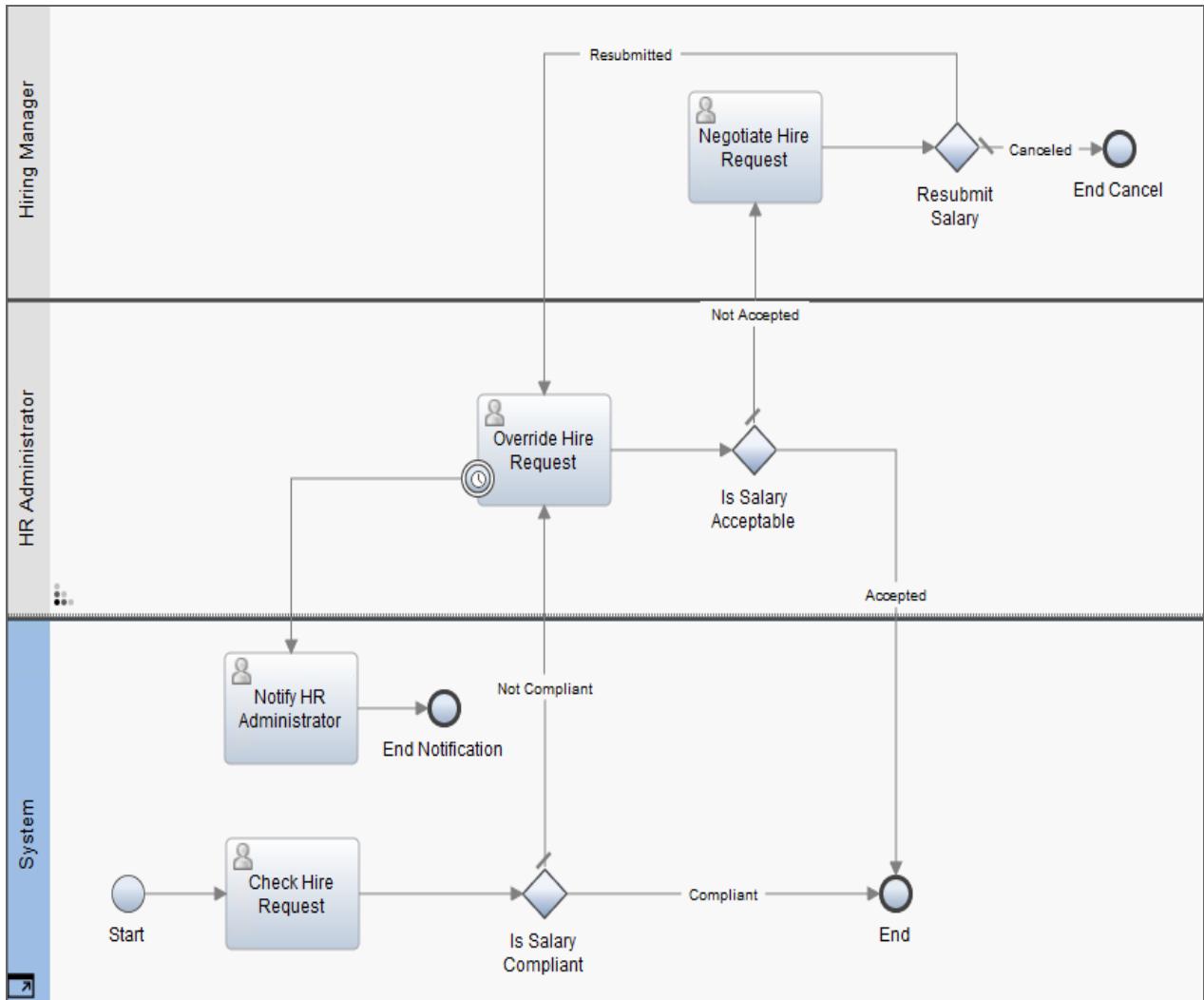
___ c. Connect the flow from **Notify HR Administrator** to **End Notification**.



___ d. Save your work.

4. Verify the process model

- a. Ensure that the **Approve Hire Request** process looks similar to the diagram before moving on:



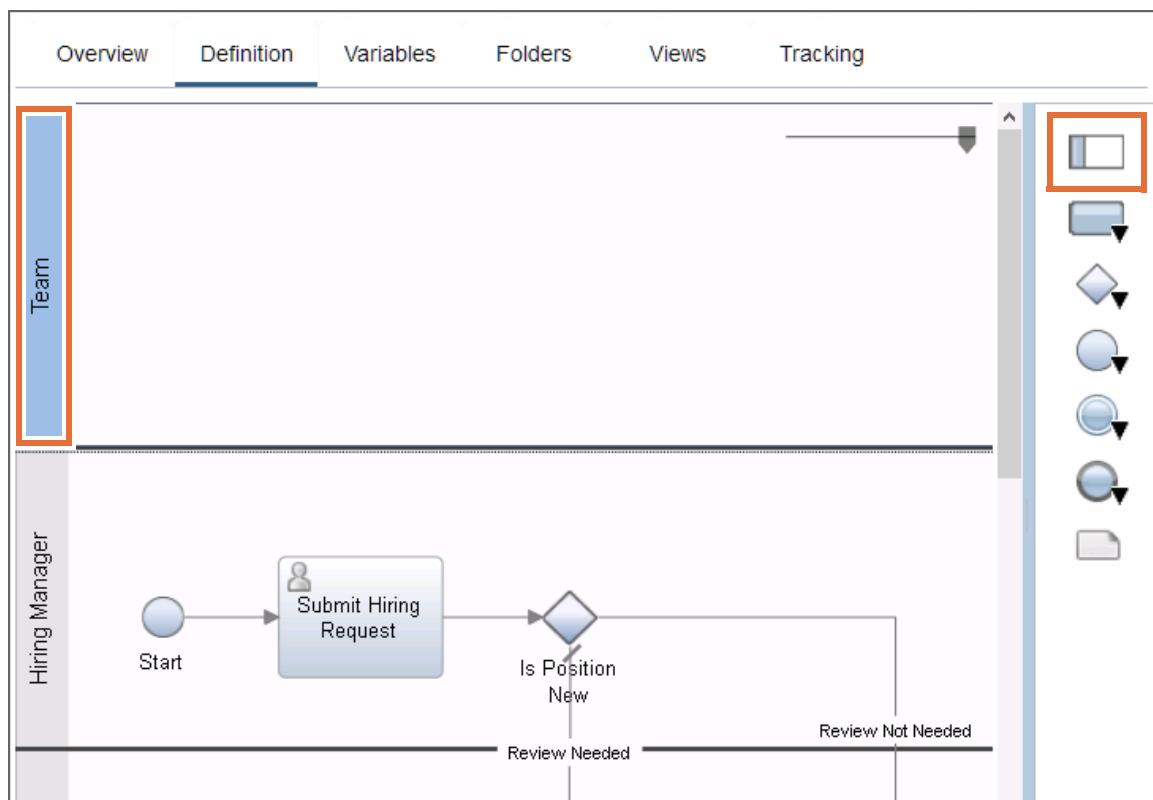
Part 4: Add new process requirements to the process

It is common to clarify requirements throughout the playback sessions. During the initial playback, the vice president mentions new requirements.

Several job postings contain wording that violates a new set of hiring laws. To address this issue, the vice president suggests that a lawyer must review every job post at some point before the job opening is released and posted to the public. The lawyer checks for legal compliance and edits if necessary.

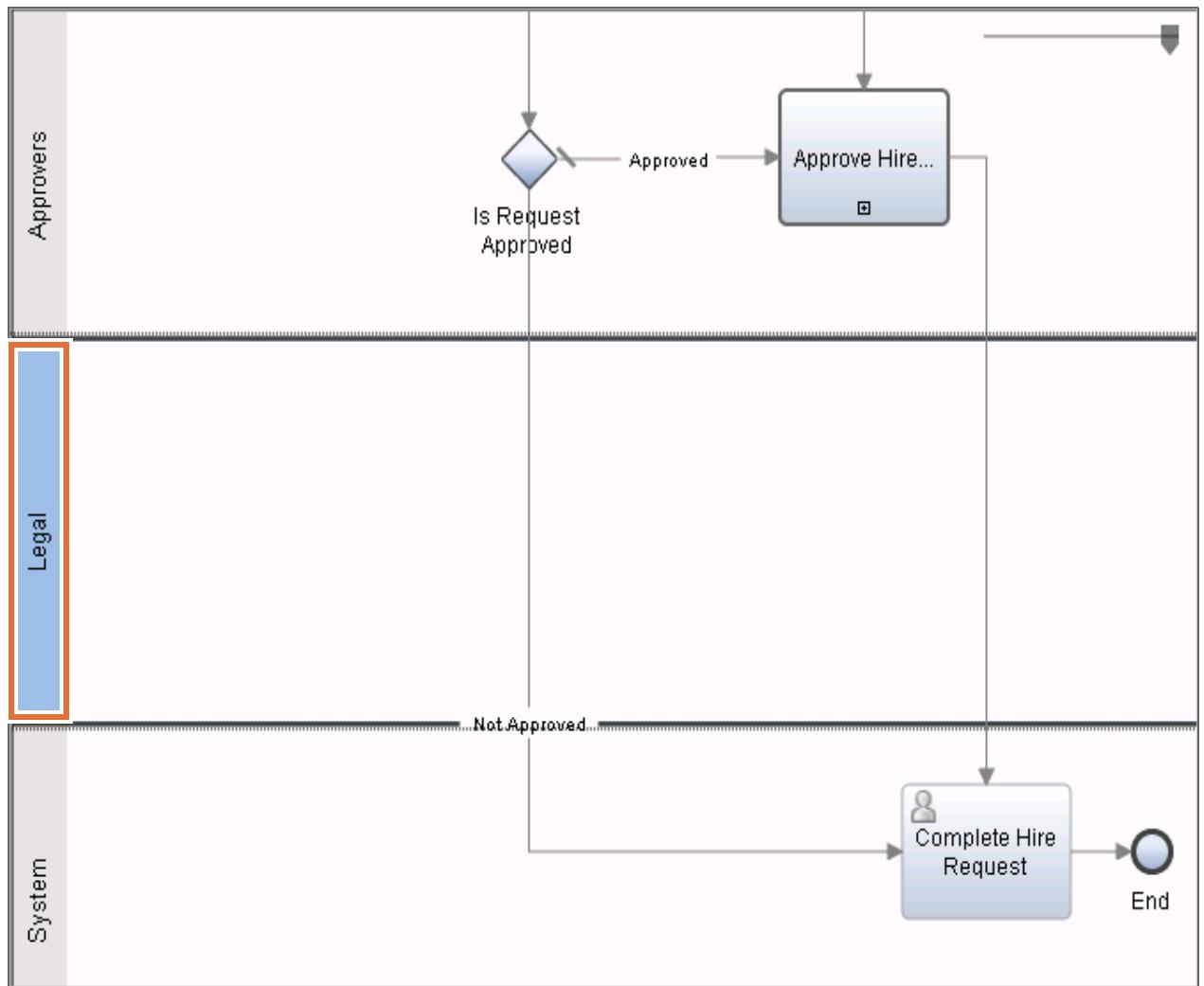
Examine the requirement and come up with the solution for this requirement.

- ___ 1. Add a lane in the Hiring Request Process for the legal team.
 - ___ a. Open the **Hiring Request Process**.
 - ___ b. Drag a **Lane** from the palette to the canvas above the Hiring Manager lane.



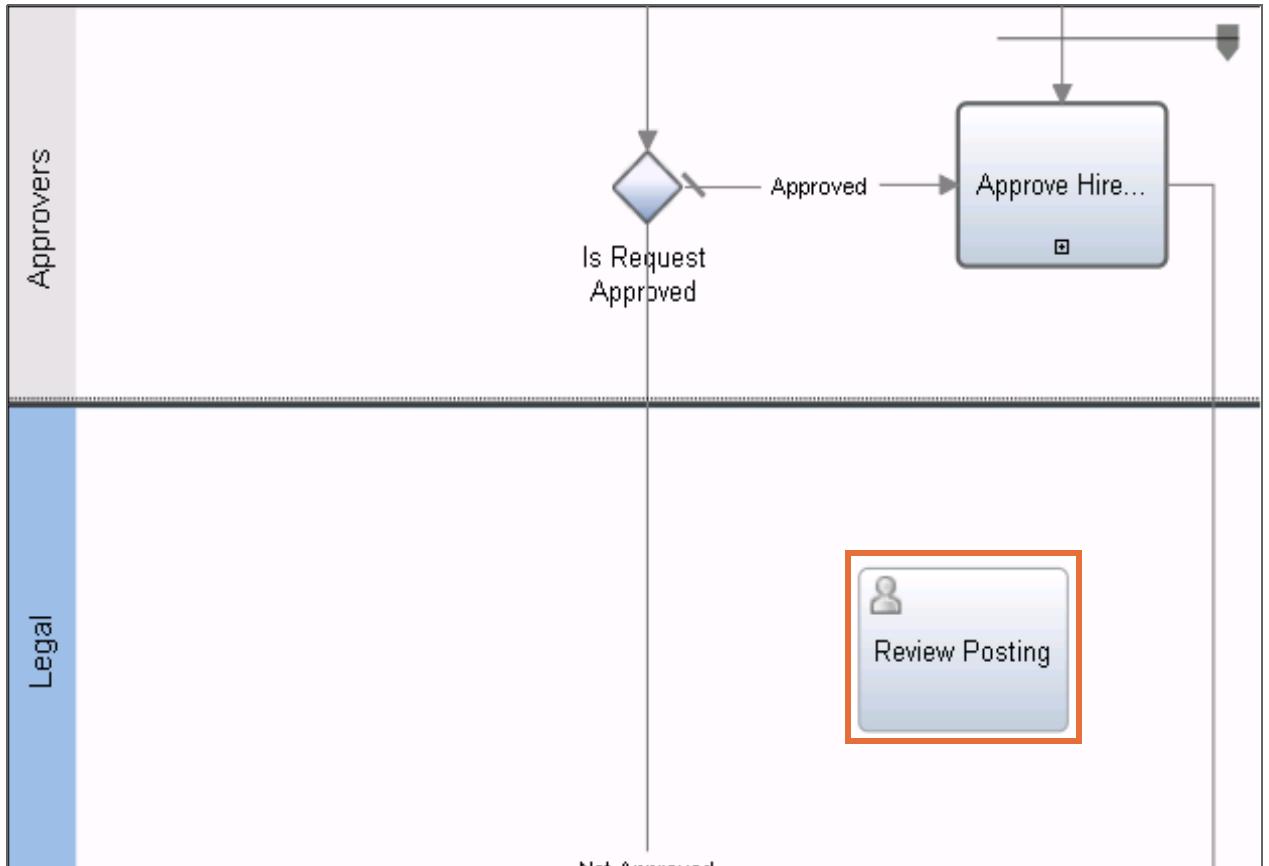
- ___ c. Name the lane: Legal

- ___ d. Right-click the **Legal** lane. Click **Move lane down**. Repeat moving the lane down three times until the Legal lane is between the **Approvers** and **System** lanes.

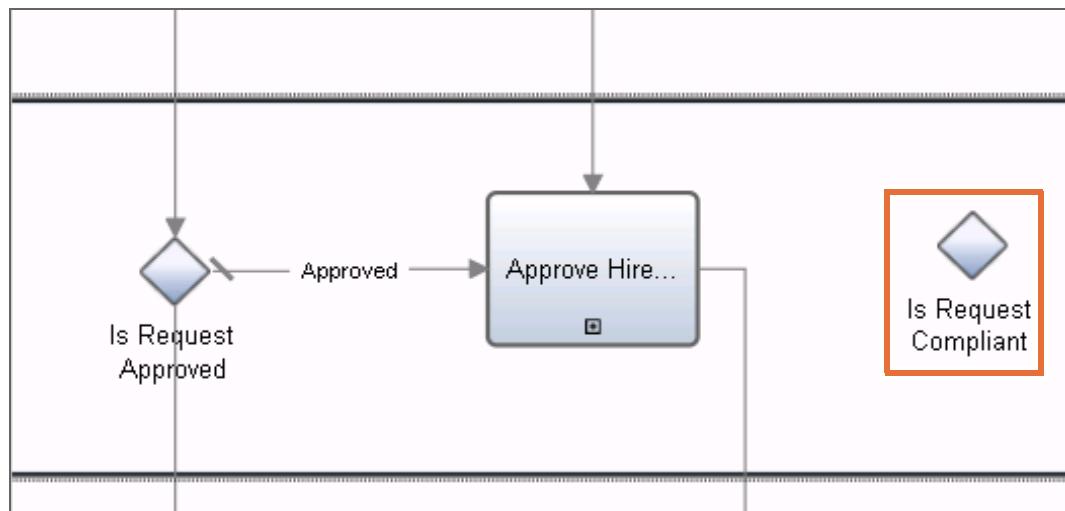


- ___ 2. Create an activity for a lawyer to review the hire request.
___ a. Drag **User Task** from the palette to the **Legal** lane in the process.

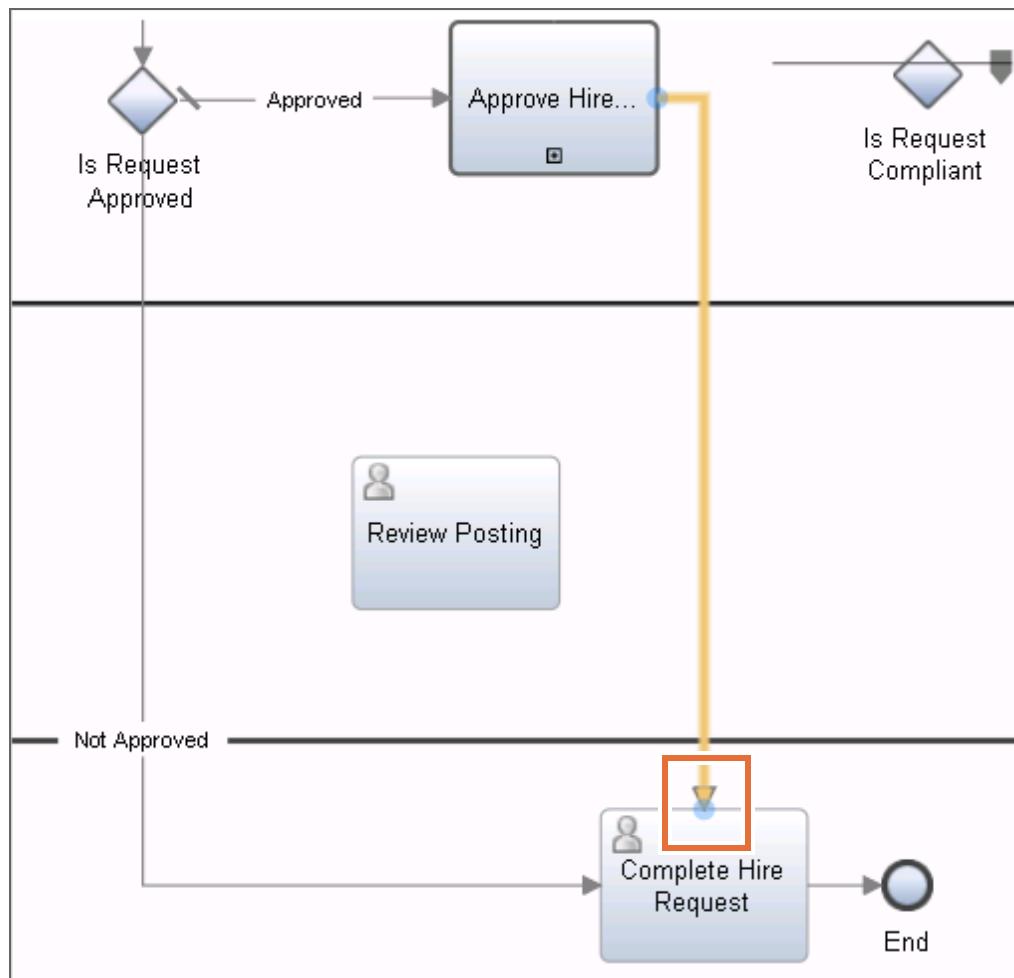
- __ b. Name the activity: Review Posting



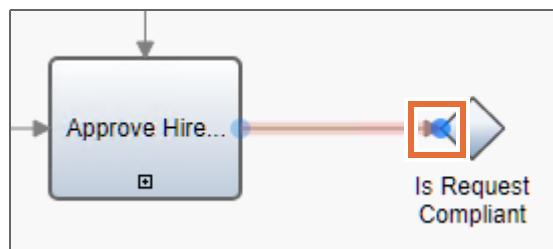
- __ 3. Add a gateway to check the compliance of the request.
- Drag a **Gateway** to the right of **Approve Hire Request**.
 - Rename the gateway to: Is Request Compliant



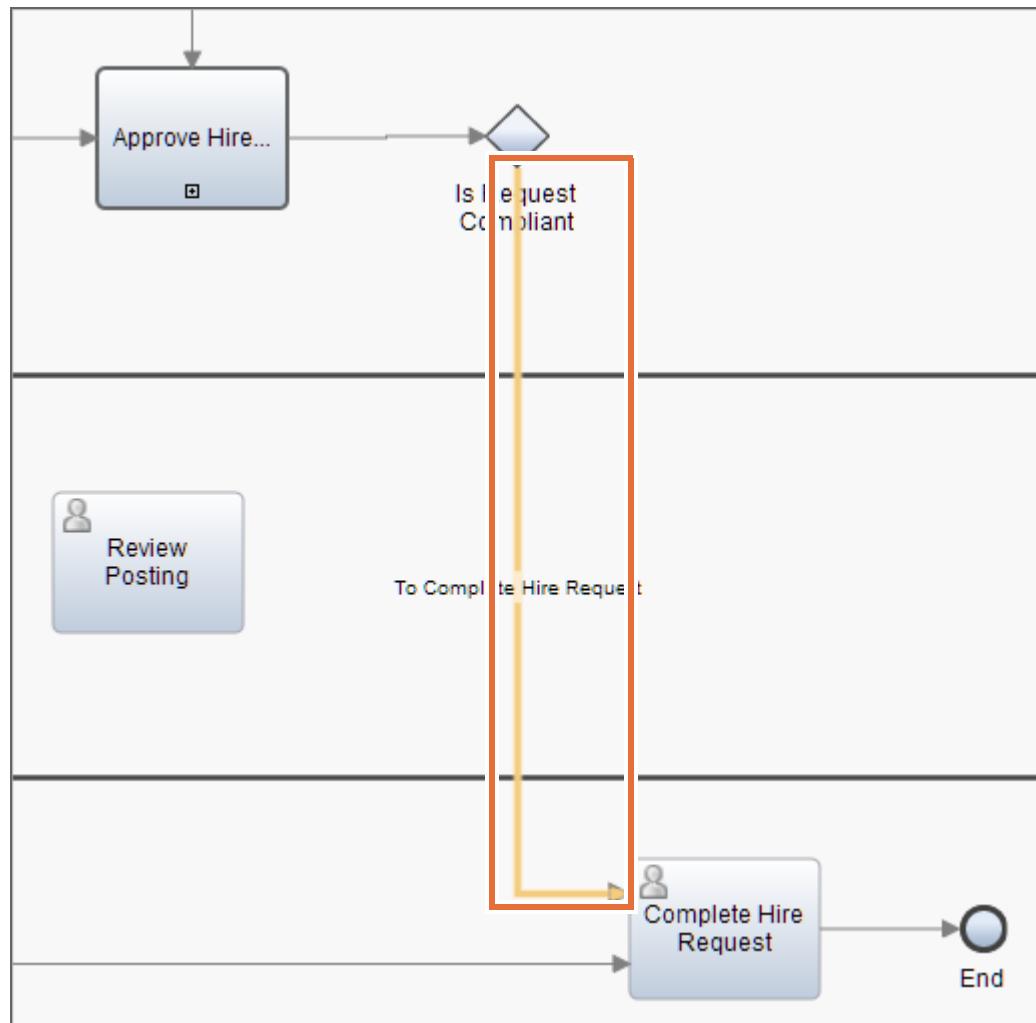
- __ 4. Connect the **Review Posting** activity.
- __ a. Select the sequence flow between **Approve Hire Request** and **Complete Hire Request** activities.



- __ b. Drag the blue point that appears at the tip of the flow line at the **Complete Hire Request** activity to the **Is Request Compliant** gateway.

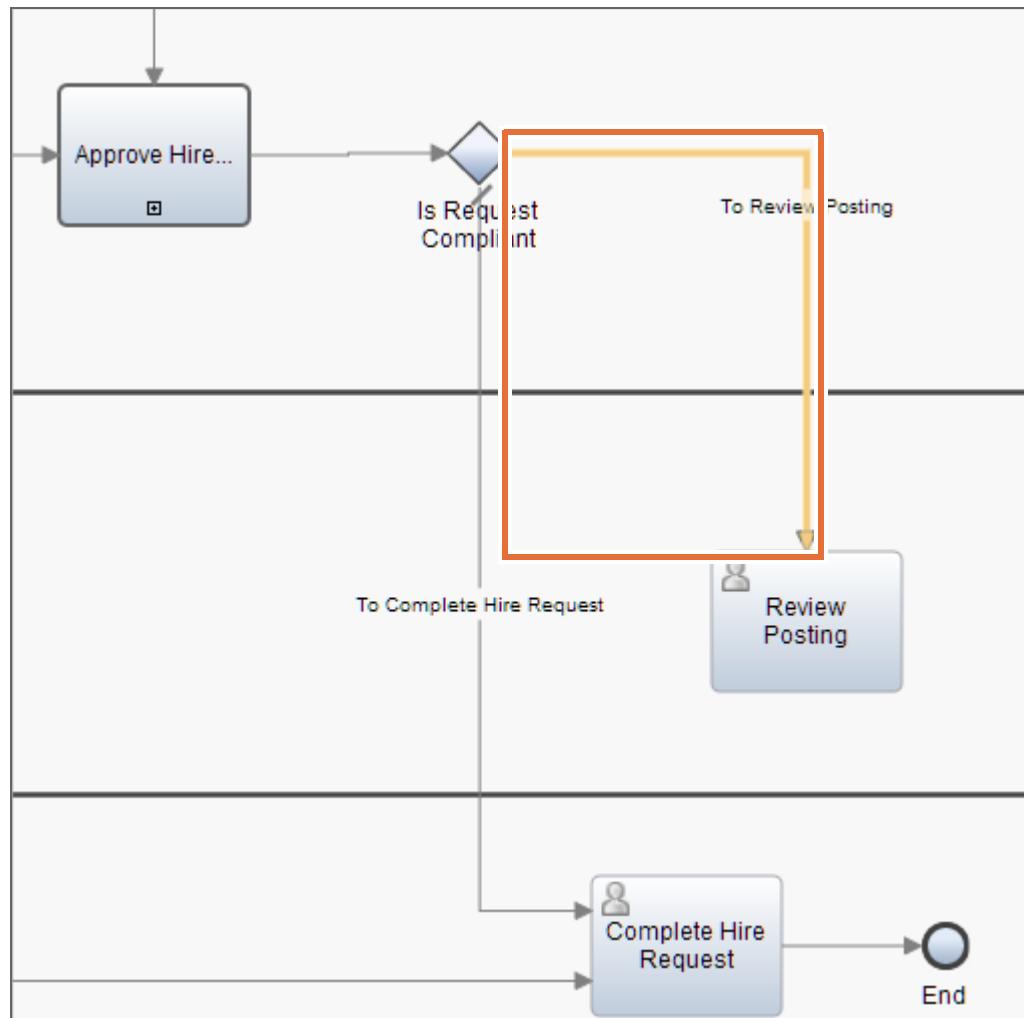


- ___ c. Connect the **Is Request Compliant** gateway to **Complete Hire Request**.

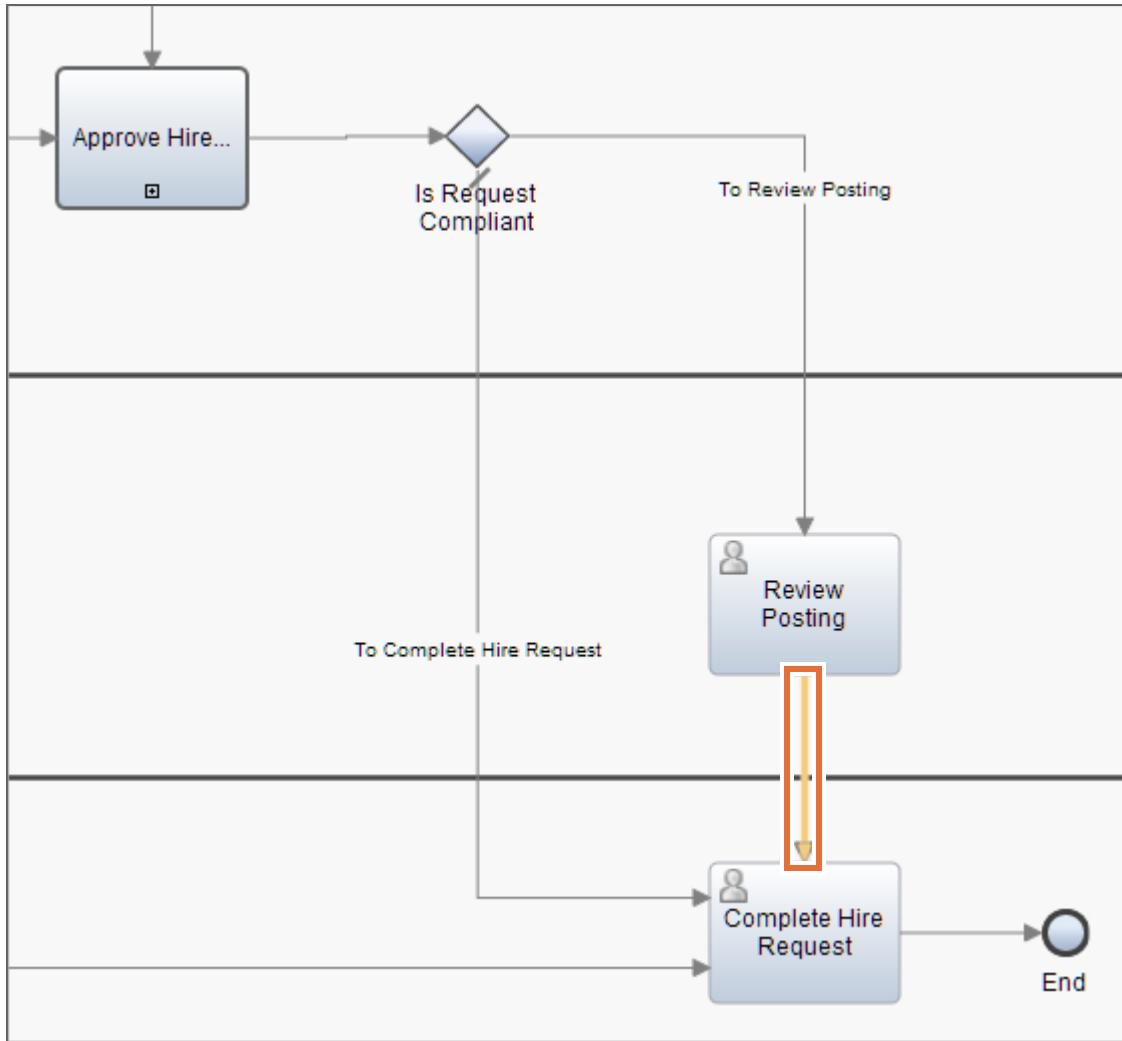


- ___ d. Drag the **Review Posting** activity above the **Complete Hire Request** activity.

- __ e. Connect the **Is Request Compliant** gateway to the **Review Posting** activity.



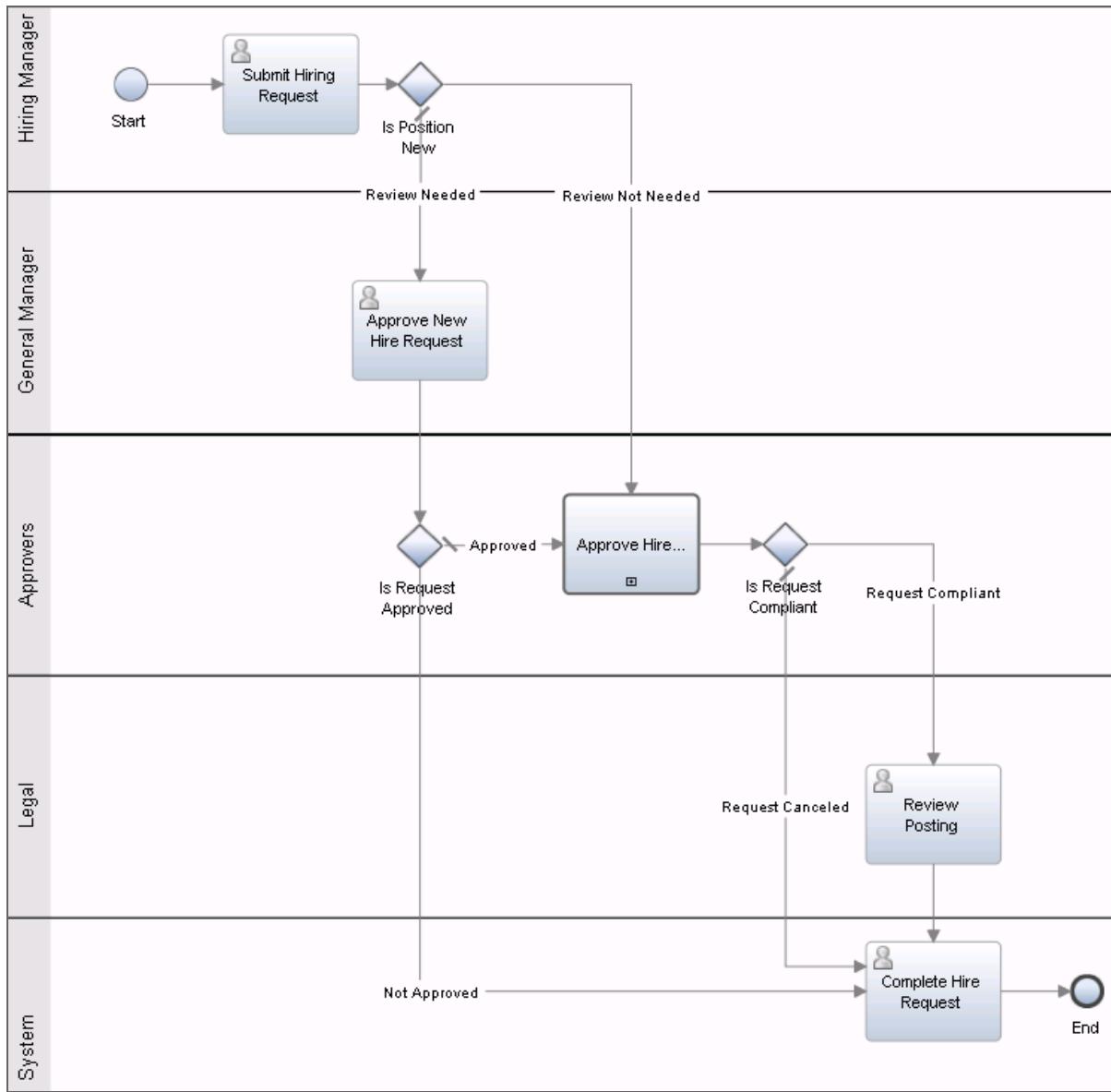
- ___ f. Create a flow from **Review Posting** to **Complete Hire Request**.



- ___ g. Rename the sequence flow between **Is Request Compliant** and **Complete Hire Request** as: Request Canceled
___ h. Rename the sequence flow between **Is Request Compliant** and **Review Posting** as: Request Compliant
___ i. Save your work.

___ 5. Verify the process model

- ___ a. Ensure that the new process looks similar to the diagram before moving on:



Important

After adding an element to your process, it is a good practice to verify the process a second time. When you encounter process problems while developing for the Playback 1 process phase, it is common to return to Playback 0 to correct the problems.

Playback 0 is complete.

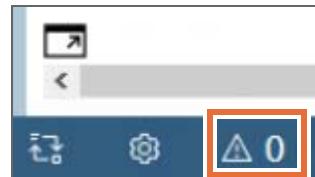
Part 5: Validate the process model and take a snapshot

Now that playback 0 is complete, it is a good time to validate the process application and take a snapshot.

- ___ 1. Validate the process application.

While you are developing, Process Designer keeps track of any validation errors and displays the number of errors in the status bar at the bottom.

- ___ a. Review the status bar validation icon for any errors.

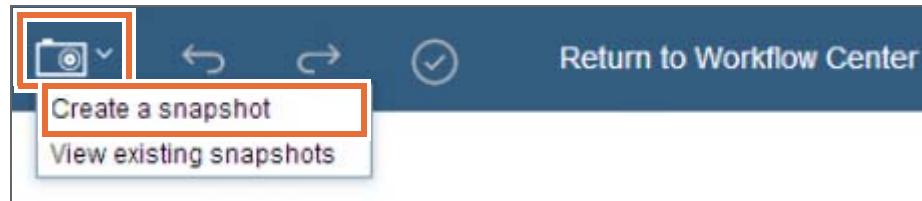


- ___ b. If you have validation errors, you should resolve them before moving onto the next playback. As you add coaches and services, this functionality becomes more important. For now, there should be no validation errors for the processes.

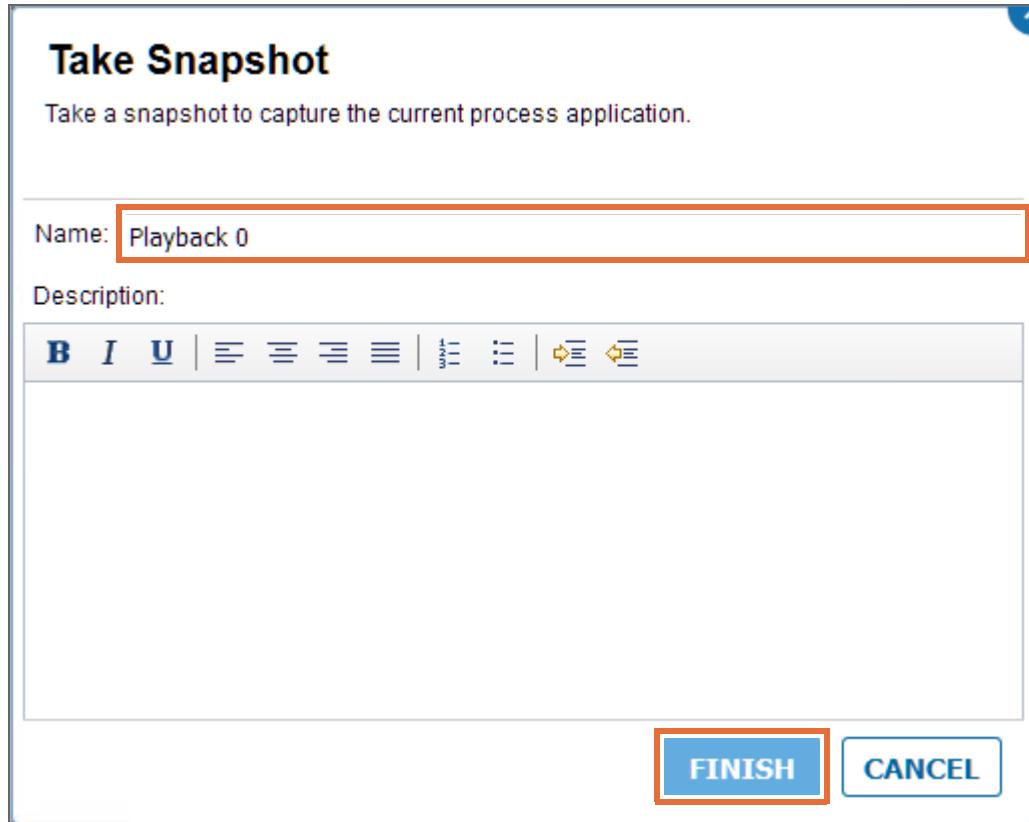
- ___ 2. Take a snapshot.

A snapshot captures the state of the library items within a process application at a specific point in time. Snapshots usually represent a milestone or are used for playbacks or for installation.

- ___ a. Click the **camera icon** in the upper right corner and select **Create a new snapshot**.



- ___ b. Enter **Playback 0** as the name and click **FINISH**.



- ___ c. Click **OK** when the snapshot is created.

You have completed this exercise.

The first part of this exercise involved adding gateways to the process in IBM Process Designer. Next, you modeled the appropriate sequence flows for each gateway. Then, you added a timer intermediate event for a process requirement. Next, you validated the business process, and you took the playback feedback and added a requirement to the process. All the stakeholders agreed to the revised version of the process, and the process owner along with the lead developer suggested that the team transition to Playback 1.

End of Exercise

Exercise 4. Playback 1: Controlling process flow with business data

Estimated time

01:30

Overview

In this exercise, you create assets that are required during Playback 1 controlling the process flow of the process lifecycle. You create variables, implement timer intermediate events, establish routing, and implement exclusive gateways.

Objectives

After completing this exercise, you should be able to:

- Create simple variables in a process
- Implement timer intermediate events in a process
- Implement gateways for a process
- Implement routing for an activity

Introduction

In this part of Playback 1, the process variables (also known as flow data) are built and implemented in the gateways. Flow data is used throughout the life of the process. It is used along with gateways to determine the flow of the process based on the values of the flow data. When a token is at a gateway, the value of the variable is evaluated to determine the next path to take. Flow data also includes the following circumstances:

- Data that is used to determine which activities to run
- Data that is used to determine who starts each activity
- Data that is used to determine when an activity is due or when an activity must be escalated

The flow data ensures that the business process gets the right activities to the right people at the right time.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Implement the intermediate timer event](#)

[Part 2: Create the process flow variables for the Hiring Request Process](#)

[Part 3: Create process flow variables for the Approve Hire Request linked process](#)

[Part 4: Implement gateways](#)

[Part 5: Implement routing for an activity](#)

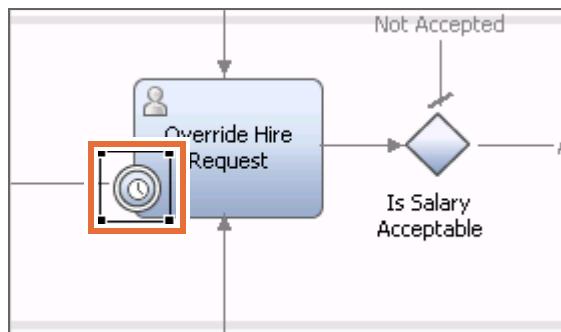
[Part 6: Review artifact references](#)

Exercise instructions

Part 1: Implement the intermediate timer event

In Playback 0 of the Hiring Request Process, an attached timer intermediate event was modeled. The HR Administrator has 4 hours to complete the review. If the review is not completed within 4 hours, an email is sent to the HR Administrator. The email notifies the HR Administrator of the missed deadline. Now your responsibility is to implement the timer event that was modeled.

- __ 1. Implement the attached timer intermediate event.
 - __ a. Open the **HR Recruitment Process** process application.
 - __ b. Open the **Approve Hire Request** process and select the **Definition** tab.
 - __ c. Click the **timer intermediate event** that is attached to the **Override Hire Request** activity.



- __ d. Below in the properties section, click **Implementation**.
- __ e. Change the **Event Properties** to match the following values:
 - **Trigger on:** After start of step
 - **Before or after difference:** 4 Hours
 - **Tolerance interval:** 0 Hours

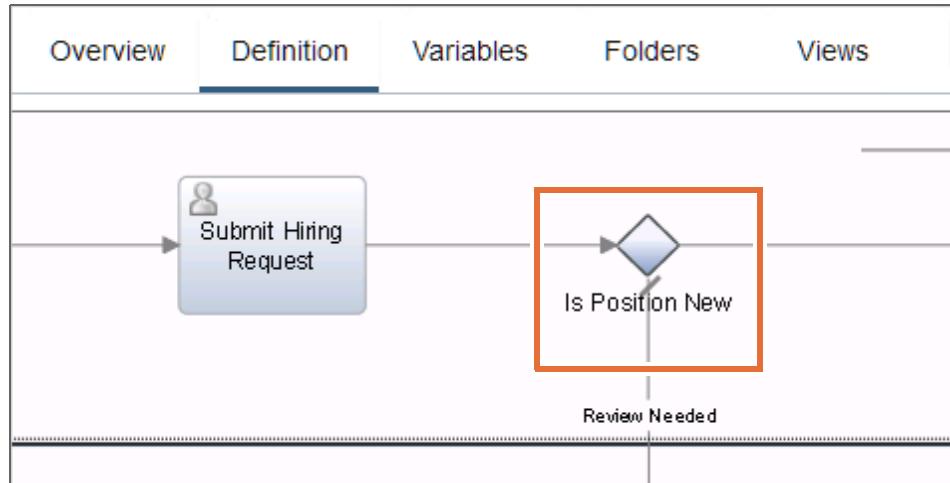
General	Implementation
Data Mapping	Boundary Event Type
Pre & Post	Timer Trigger on: After start of step Custom date: Before or after difference: 4 Hours Tolerance interval: 0 Hours Use the activity work schedule: <input type="checkbox"/>

- ___ f. Save your work.

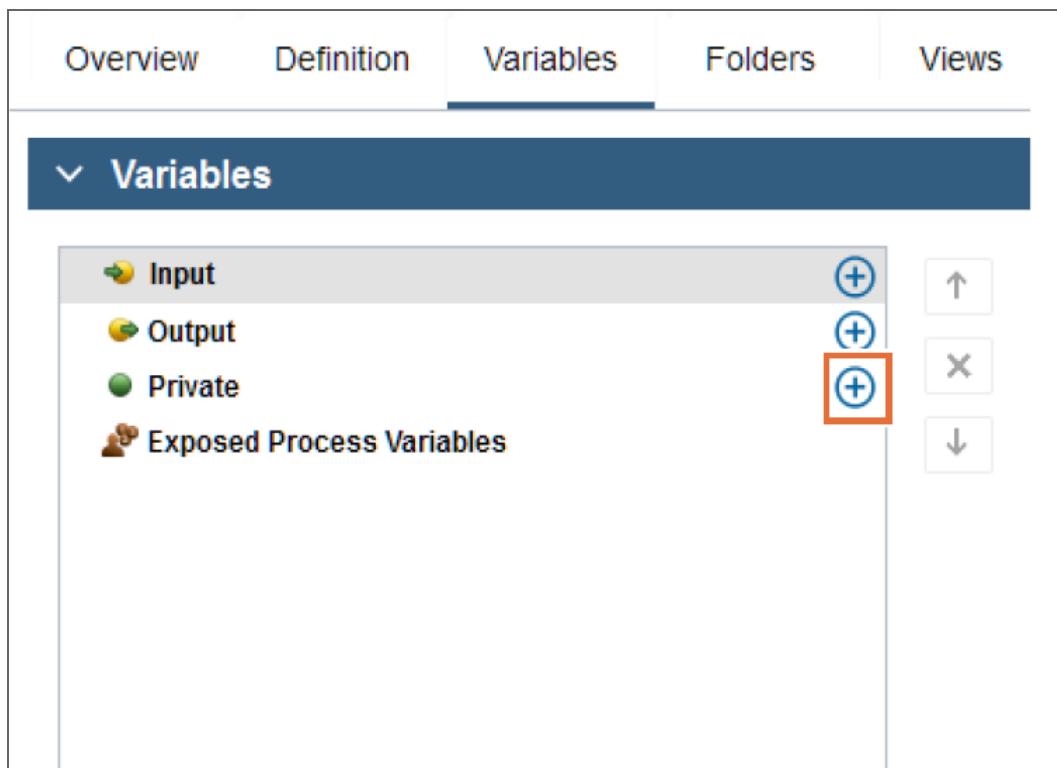
Part 2: Create the process flow variables for the Hiring Request Process

The process model for the Hiring Request Process is in place. Data flow implementation is one of the goals of Playback 1. Now you must start to implement the process, first by creating simple variables to implement the logic behind the decision gateways on the current process.

- ___ 1. Create a private variable to control the **Is Position New** exclusive gateway.



- ___ a. Open the **Hiring Request Process**.
- ___ b. Click the **Variables** tab.
- ___ c. Click the **(+)** plus sign next to Private.



- ___ d. Name the private variable: `isNewPosition`
 Confirm that the Variable Type is **String**.

The screenshot shows the SAP Fiori 'Details' dialog for a variable. The 'Name' field contains 'isNewPosition'. The 'Variable type' field is set to 'String' (highlighted with a red box), with options for 'System Data', 'Select...', and 'New...'. Below these fields are 'List' and 'Visible in Process Portal' checkboxes, both of which are unchecked.

String variables are used here instead of Boolean for flexibility of the implementation. If requirements change later, and a third outflow is added to the gateway, a String is easier to implement the change than a Boolean would be.

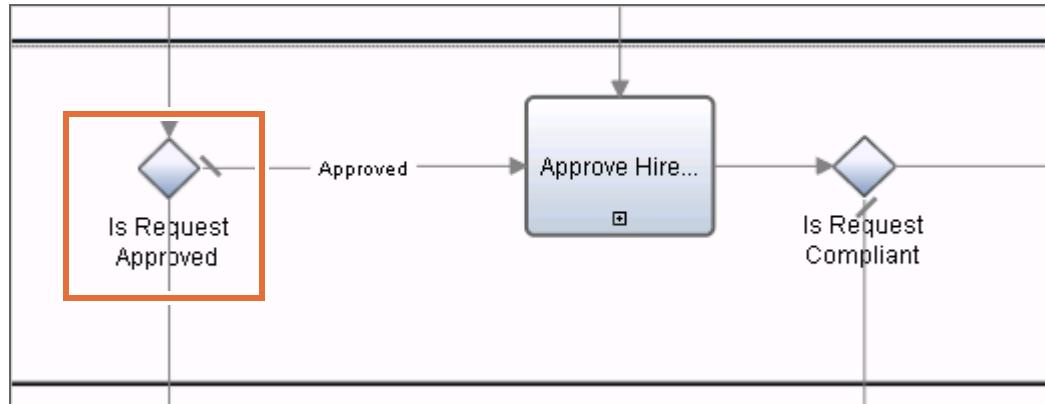
- ___ e. In the **Documentation** field, insert the following text:

0 = Not a new position
 1 = New position, requires approval

The screenshot shows the SAP Fiori 'Details' dialog for a variable. The 'Name' field is 'isNewPosition'. The 'Documentation' field contains the text '0 = Not a new position' and '1 = New position, requires approval' (highlighted with a red box).

- ___ f. Do not change the other options. Save your changes.

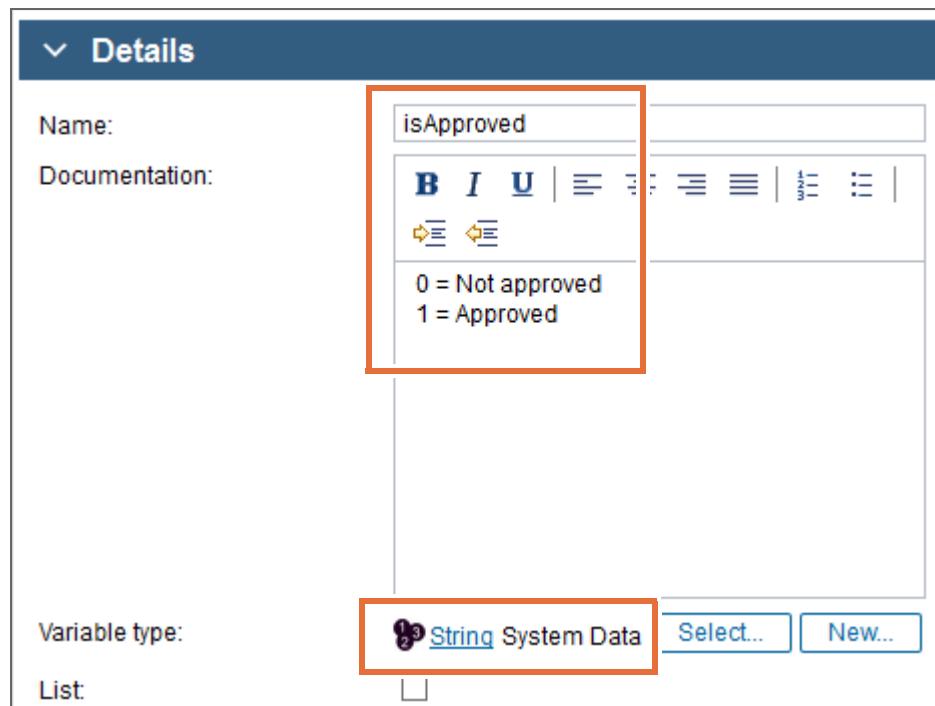
- __ 2. Create a variable to control the **Is Request Approved** exclusive gateway.



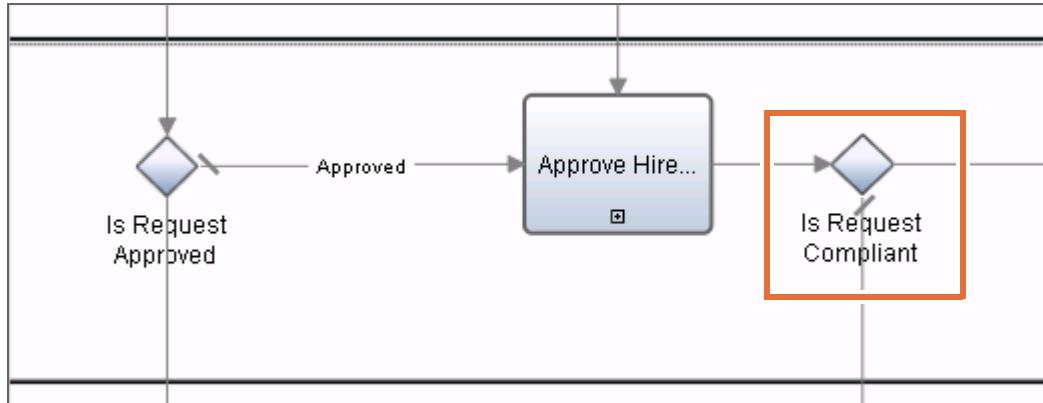
- __ a. On the **Variables** tab, create another Private variable: `isApproved` (String)
 __ b. In the **Documentation** field, insert the following text:

0 = Not approved

1 = Approved

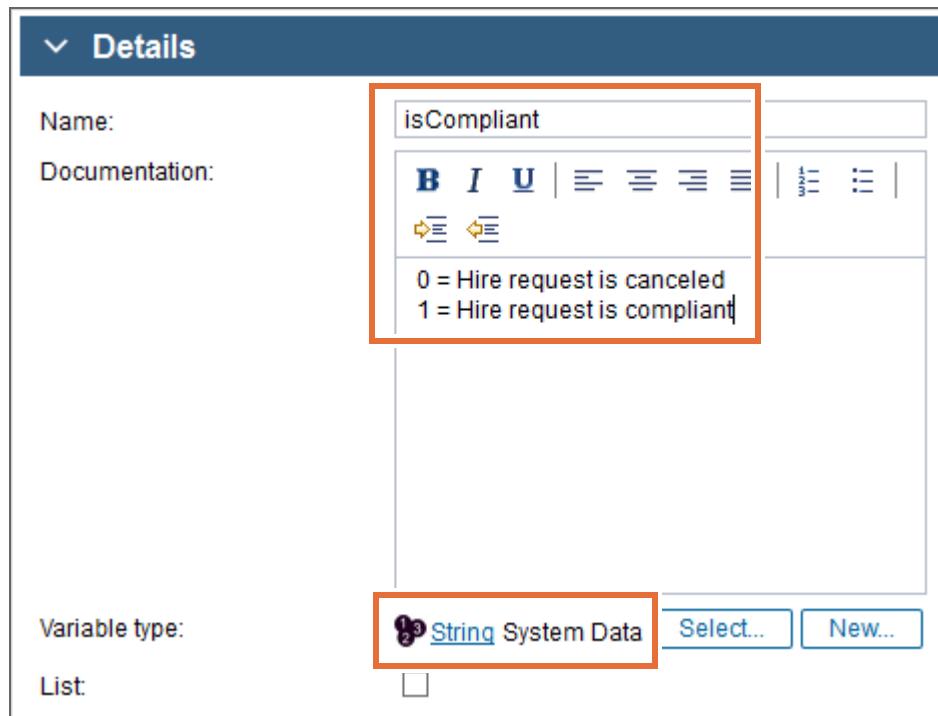


- ___ 3. Create a variable to control the **Is Request Compliant** exclusive gateway.



- ___ a. On the **Variables** tab, create another Private variable: `isCompliant` (String)
 The `isCompliant` process flow variable maps to a variable in the Approve Hire Request linked process.
- ___ b. In the **Documentation** field, insert the following text:

0 = Hire request is canceled
 1 = Hire request is compliant



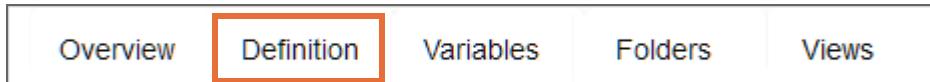
- ___ c. Save your changes.

You created three private variables in this exercise.

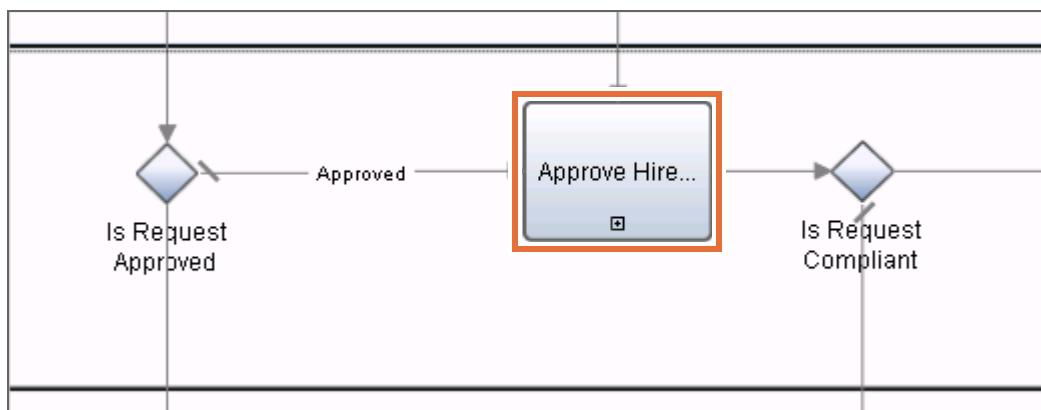
The screenshot shows a software interface for managing variables. At the top, there's a header with a dropdown arrow and the word "Variables". Below the header, there are three categories: "Input" (with a plus sign icon), "Output" (with a plus sign icon), and "Private" (with a plus sign icon). The "Private" category is expanded, revealing three variables listed in a box with a red border: "isNewPosition (String)", "isApproved (String)", and "isCompliant (String)". To the right of the variable list are four small icons: an upward arrow, a plus sign, a downward arrow, and a minus sign. At the bottom of the variable list is a section labeled "Exposed Process Variables" with a plus sign icon. The entire interface is contained within a light gray frame.

Part 3: Create process flow variables for the Approve Hire Request linked process

- 1. Create a variable for the **Is Salary Compliant** exclusive gateway in the **Approve Hire Request** process. Because the parent process uses this data, create a variable that passes data through the linked process as an input and output variable.
- a. Return to the **Hiring Request Process** model by clicking the **Definition** tab.

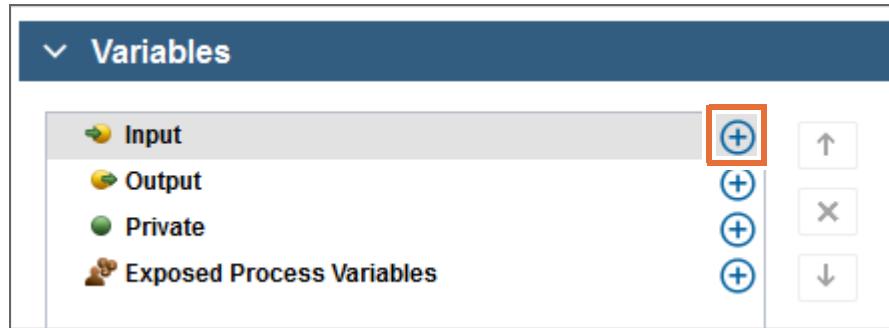


- b. Open the **Approve Hire Request** linked process by double-clicking the linked process in the diagram.



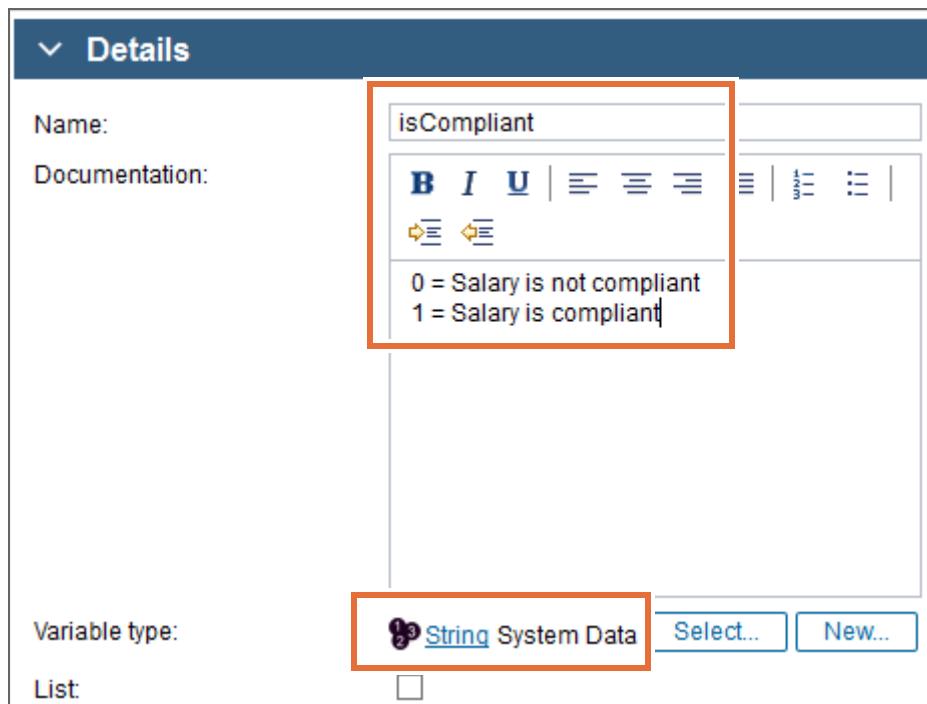
Be sure that you select the **Approve Hire Request** linked process, not the **Approve New Hire Request** activity. Look for the linked process marker on the bottom center.

- ___ c. Click the **Variables** tab for the **Approve Hire Request** linked process.
- ___ d. Click the **(+)** plus sign next to Input.



- ___ e. Name the input variable: `isCompliant`
Verify that the Variable Type is **String**.
- ___ f. Insert the following text in the **Documentation** field:

 0 = Salary is not compliant
 1 = Salary is compliant

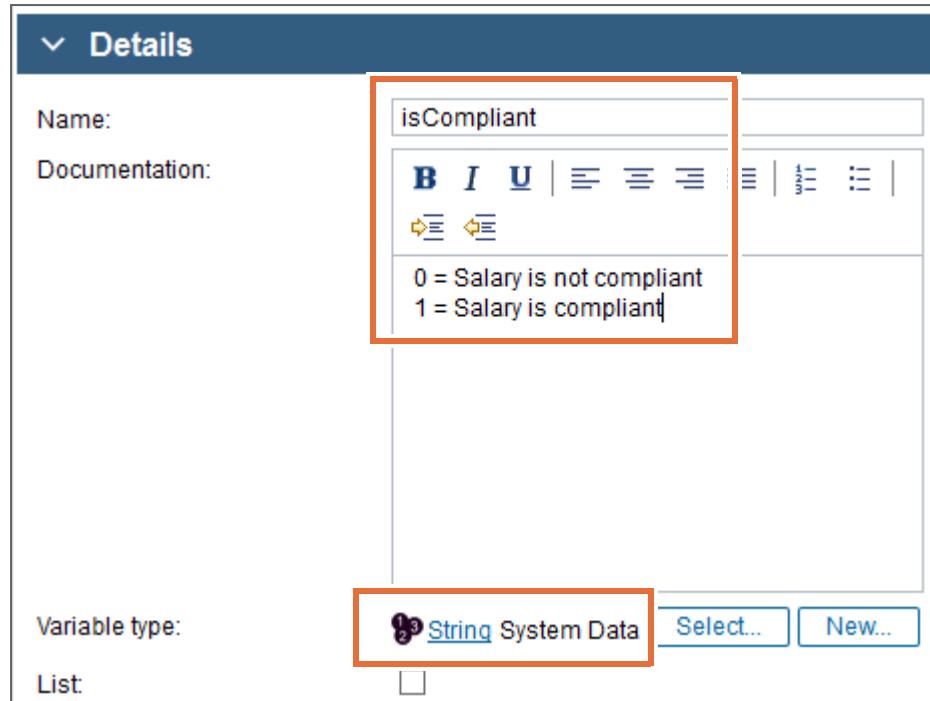


- ___ g. Create an output variable: `isCompliant` (`String`)

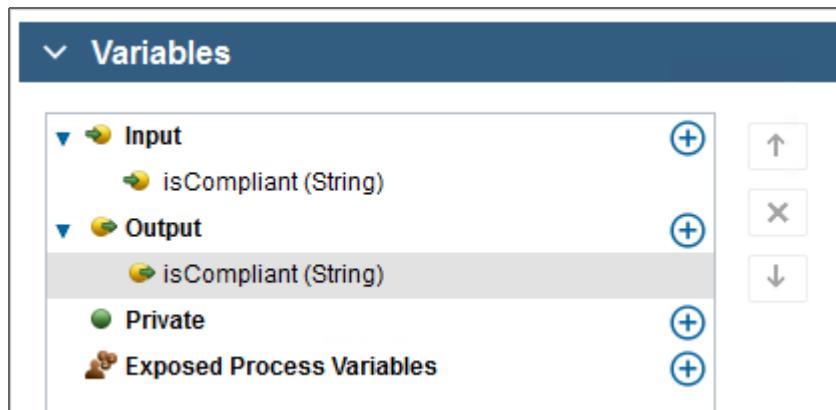
- __ h. Insert the following text in the **Documentation** field:

0 = Salary is not compliant

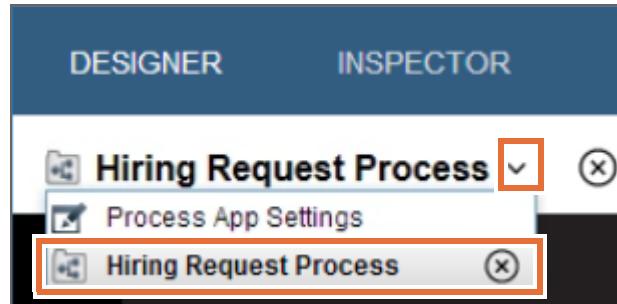
1 = Salary is compliant



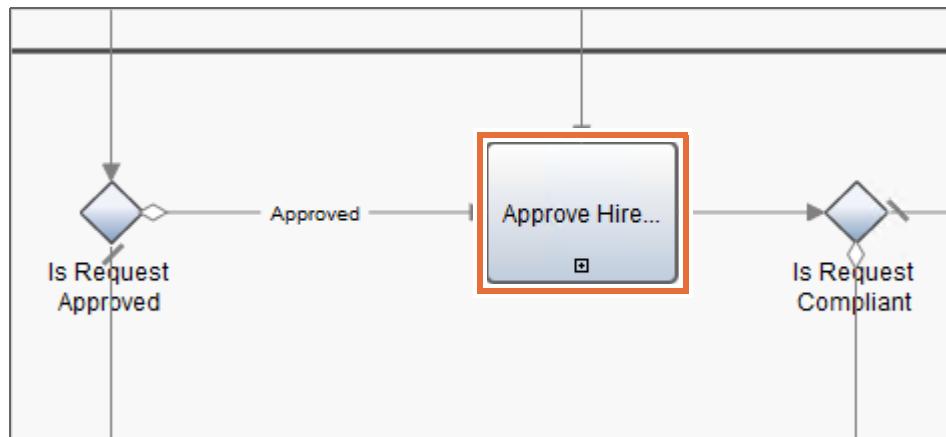
- __ i. Save your work.



- ___ 2. Map the process flow variable to the Approve Hire Request linked process.
- ___ a. Return to the Hiring Request Process by opening the **History** menu at the top and selecting **Hiring Request Process**.



- ___ a. On the **Hiring Request Process**, select the **Approve Hire Request** linked process.



- ___ b. Click the **Data Mapping** menu.
- ___ c. Click the **auto-map icon** on both the Input Mapping and Output Mapping sections.

The screenshot shows the 'Data Mapping' interface. At the top, there are two sections: 'Input Mapping' and 'Output Mapping', each with a blue header and a small orange square icon. Below each section is a mapping row. In the 'Input Mapping' section, the source variable is 'tw.local.isCompliant' and the target variable is 'isCompliant...'. In the 'Output Mapping' section, the source variable is 'isCompliant...' and the target variable is 'tw.local.isCompliant'. Both rows contain a small orange square icon with a double-headed arrow between them, indicating an auto-mapped relationship.

Because the variables that are mapped use the same name and variable type, the auto-mapping feature is appropriate to be used here. This feature is especially helpful when there are many variables that are mapped with the same names and types.

- ___ d. Save your work.

When you run the process, the value of the `isCompliant` variable is passed in and out of the Approve Hire Request linked process. Any changes to the variable while the Approve Hire Request process runs are reflected when the `isCompliant` variable is passed back to the parent process.

Part 4: Implement gateways

When you modeled the process during Playback 0, you added several gateways to the process. Now you must implement these gateways. In this section, you enhance the process by implementing the exclusive gateways in both the Hiring Request Process and the linked process. You use the simple variables that were created in the previous sections to implement the gateways.

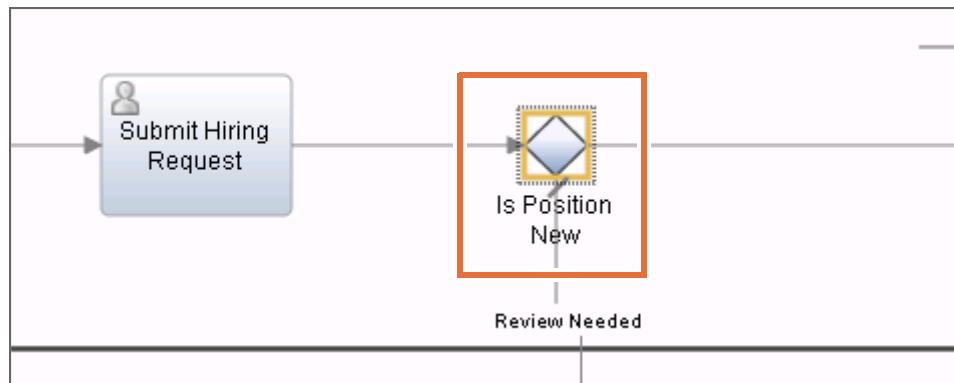
You implement the exclusive gateways in the Hiring Request Process process and the Approve Hire Request linked process to meet the following requirements:

- If the request is for a new position, the request is forwarded to the General Manager. After the General Manager receives the request, the General Manager approves or rejects the request.
- If the request is rejected, the system processes the request and the Hiring Request process is complete. If the request is approved, the process flows to the Approve Hire Request linked process.
- In the linked process, after the requisition is submitted, an automated system-level check for salary compliance starts. If the request meets salary compliance, the hiring request is forwarded to Legal for review. If the request is canceled, it is automatically sent to the HR Positions database and processed.

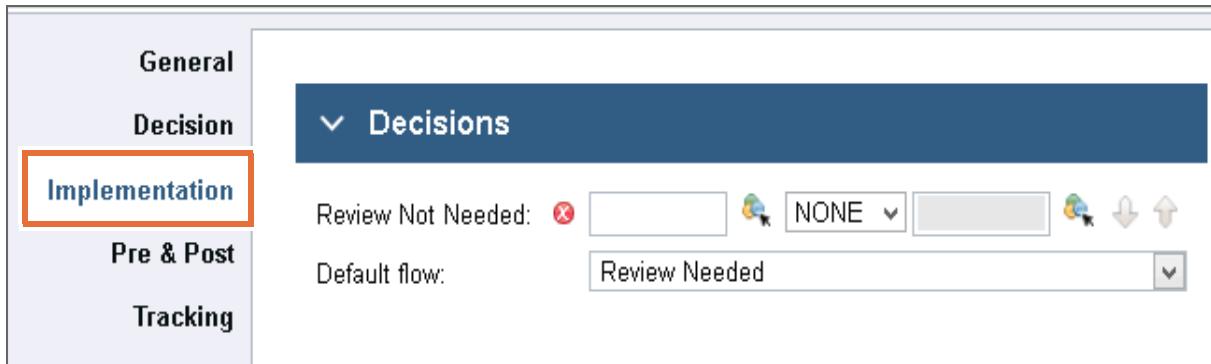
1. Implement the **Is Position New** exclusive gateway.

When the position is new, the process must flow to the Approve New Hire Request activity down the Review Needed flow.

- a. In the Hiring Request Process diagram, click the **Is Position New** exclusive gateway. Note where the default flow is. This changes when you implement the gateway later in the exercise.



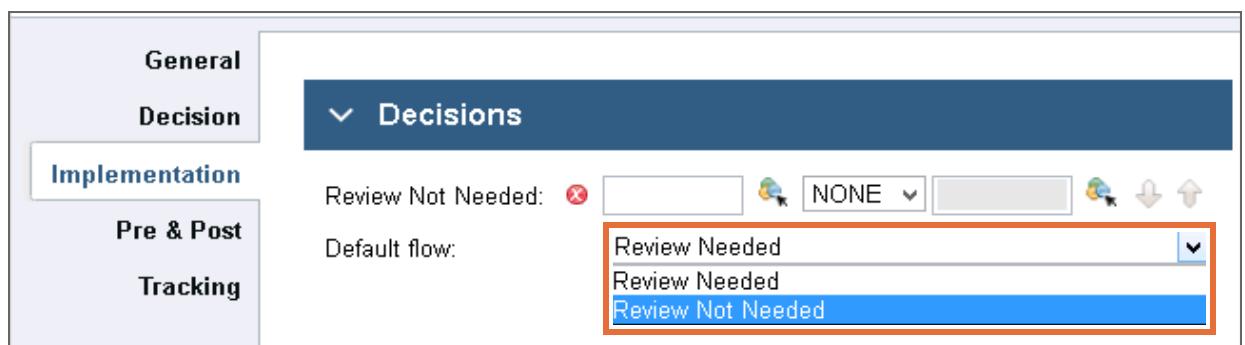
- __ b. Open the **Properties > Implementation** menu.



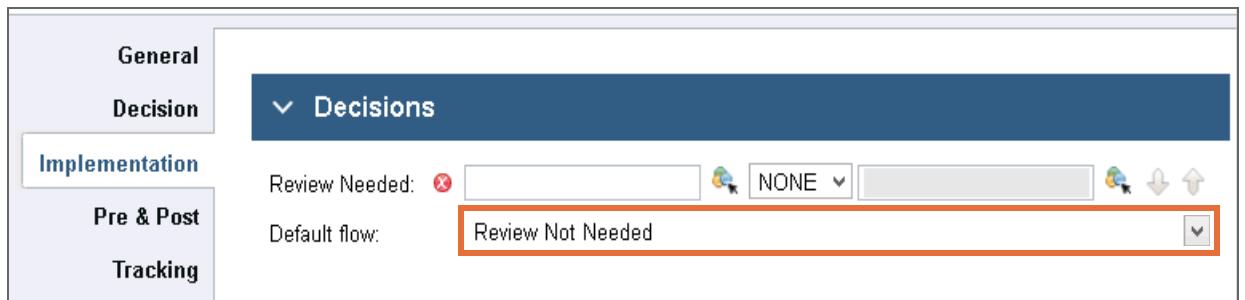
Information

You see a red “x” next to the field while the decision is being configured. It clears when the decision is defined.

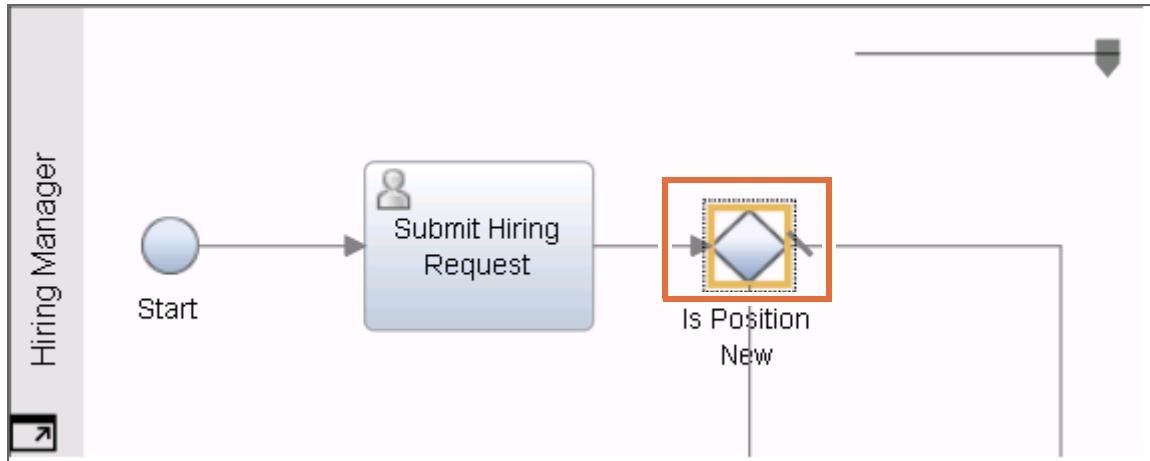
- __ c. In the **Decisions** section, click the arrow to expand the **Default flow** selection box.



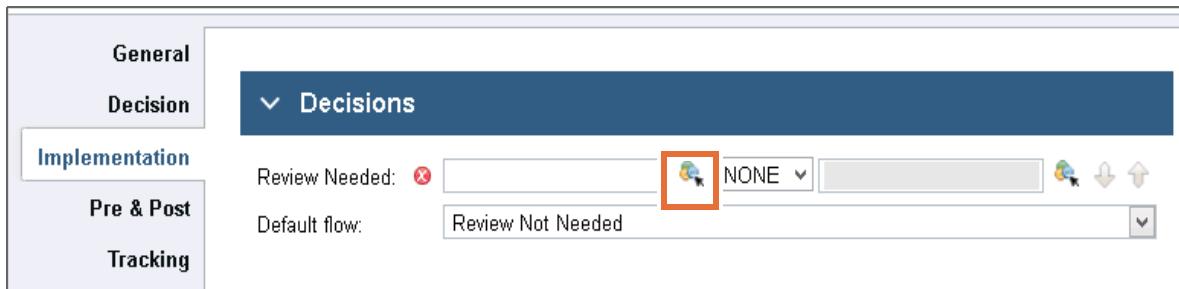
- __ d. Select **Review Not Needed**.



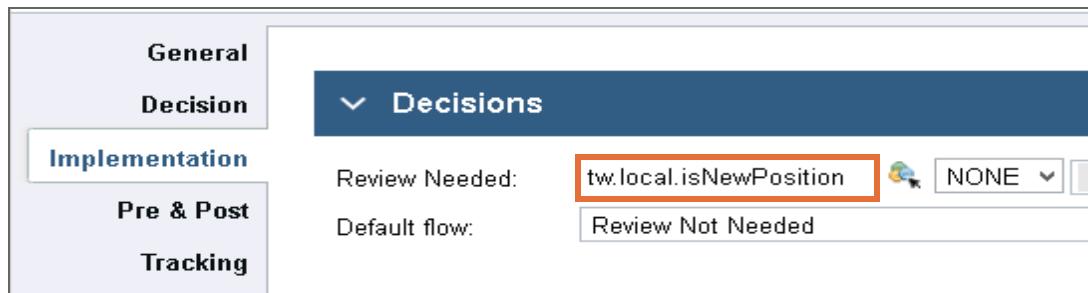
- __ e. Verify that the default flow is now changed for the **Is Position New** gateway.



- __ f. In the first field to the right of the **Review Needed** label, click the **Variable Picker** icon.



- __ g. Select **isNewPosition (String)**. The field is populated with the variable:
`tw.local.isNewPosition`



**Hint**

You can also type this variable and as you type and press **Ctrl+Spacebar**, the auto-complete feature suggests different options to you that match what you already typed. The options are filtered as you complete your entry.

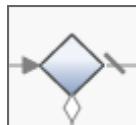
The auto-complete window provides other contextual information on the object or object type. Use the mouse or the up and down arrows on the keyboard to select the correct option, and press Enter to select the option.

- ___ h. Next, change the evaluation value to: ==
- ___ i. In the last field to the right of **Review Needed**, enter the value: "1"

The condition is set for the Is Position New gateway. If the first condition is not met, the default condition is Review Not Needed.

**Important**

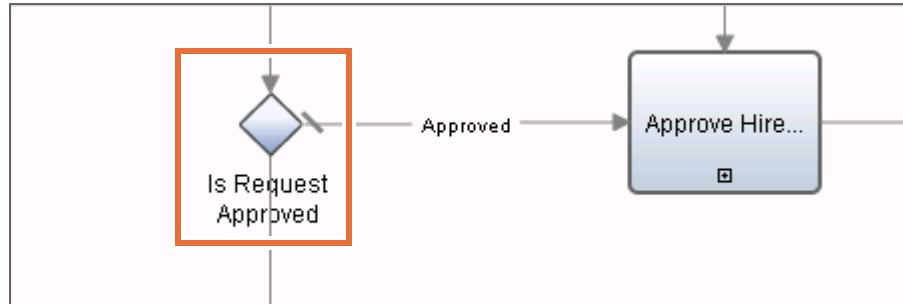
Once the gateway is implemented, the conditional flow is flagged with a diamond at the point at which it exits the gateway.



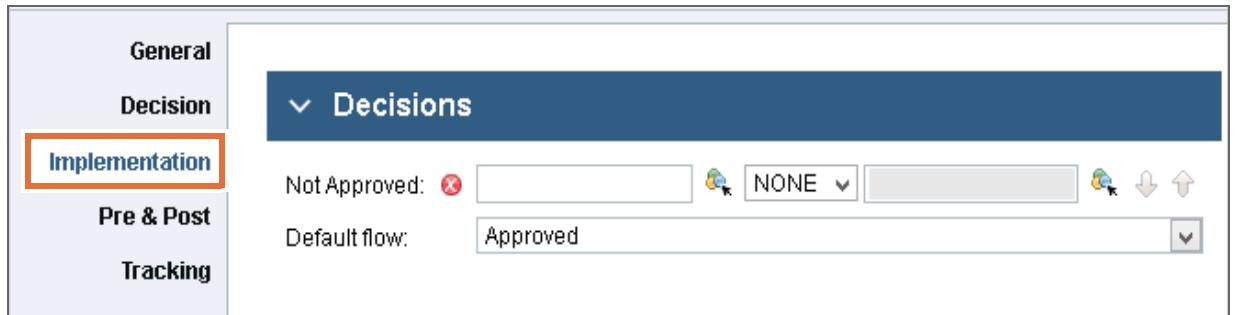
- ___ j. Save your work. The red "x" clears when the decision is saved
- ___ 2. Implement the **Is Request Approved** exclusive gateway.

When the request is approved, the process must flow to the Approve Hire Request linked process down the Approved flow.

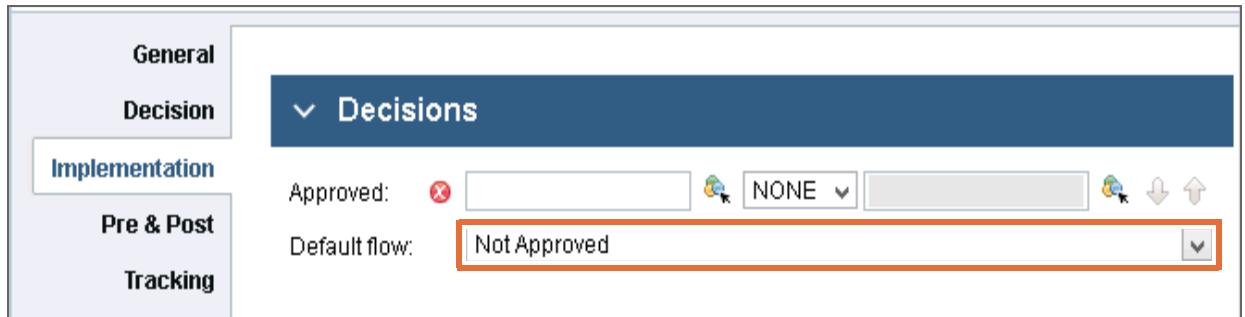
- ___ a. Select the **Is Request Approved** exclusive gateway on the process.



- ___ b. In the Properties pane, open the **Implementation** menu.



- ___ c. In the **Decisions** section, expand the Default flow selector.
- ___ d. Select **Not Approved**. This selection makes **Not Approved** the default sequence flow. The order of sequence flow starts with **Approved** at the top and **Not Approved** on the bottom.



- ___ e. In the first field to the right of **Approved**, select the following using the variable picker:
tw.local.isApproved
- ___ f. Next, change the evaluation value to: ==

- ___ g. In the last field, enter the value: "1"

General

Decision

Implementation

Pre & Post

Tracking

Decisions

Approved: tw.local.isApproved == "1"

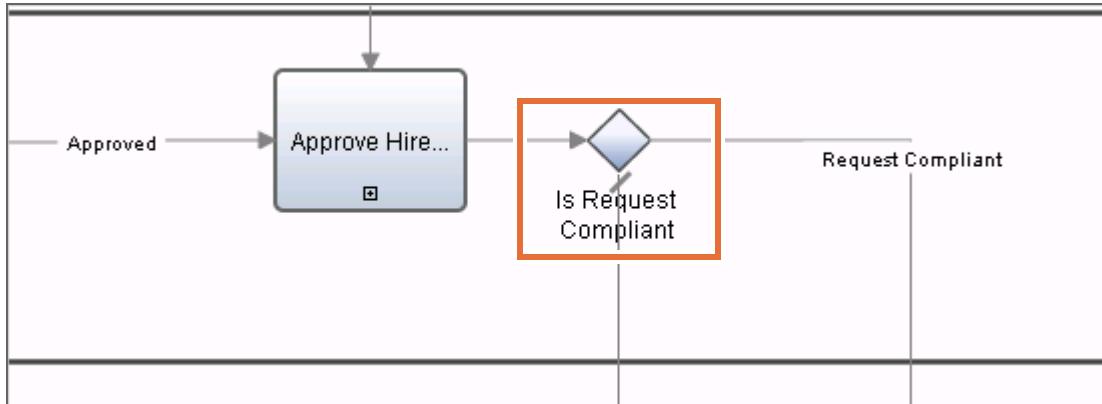
Default flow: Not Approved

- ___ 3. Implement the **Is Request Compliant** exclusive gateway.

When the request is compliant, the process must flow to the Review Posting activity down the Request Compliant flow.

The conditions for this gateway are set in a decision service in Playback 2. All the conditions that must be met are established as a business rule. The intent for this playback is merely to control the flow. For now, focus on implementing the gateway, not how the data is generated.

- ___ a. Click the **Is Request Compliant** gateway on the process.



- ___ b. Click the **Implementation** menu.

General

Decision

Implementation

Pre & Post

Tracking

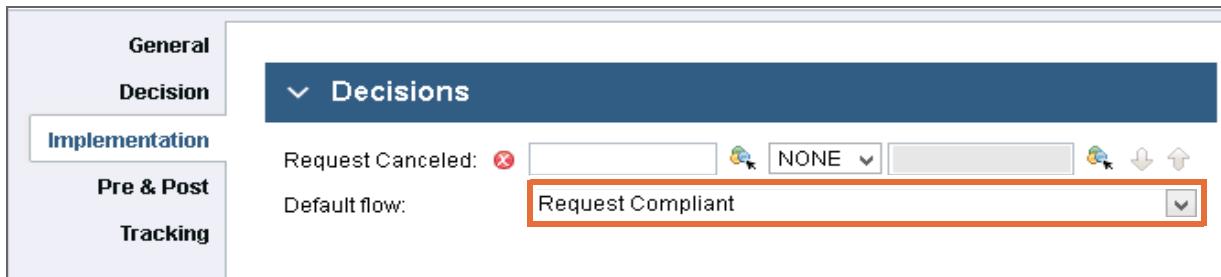
Decisions

Request Compliant: NONE

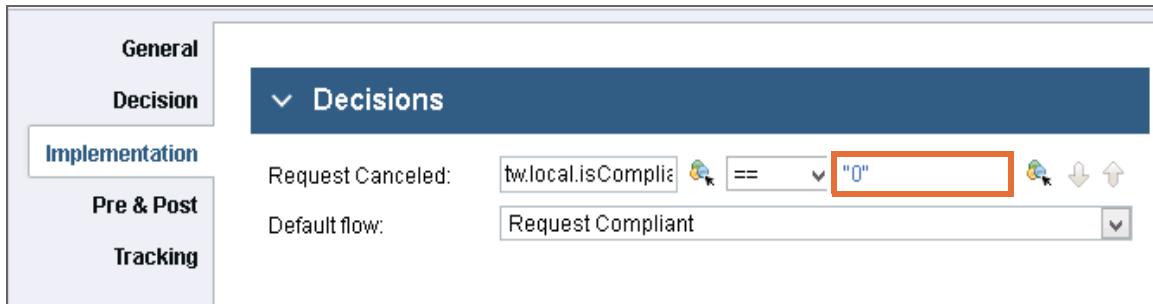
Default flow: Request Canceled

- ___ c. In the **Implementation** section, expand the **Default flow** selector.

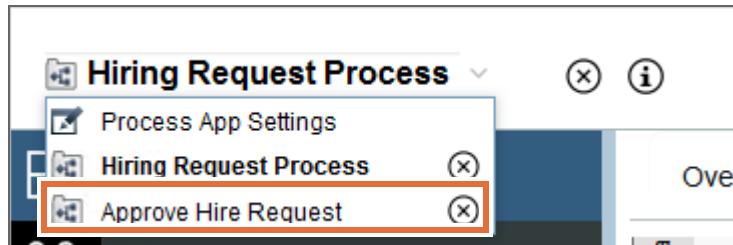
- ___ d. Select **Request Compliant**. This arrangement makes **Request Compliant** the default sequence flow. The order of sequence flow expresses that the **Request Canceled** option is first and that the **Request Compliant** option is second.



- ___ e. In the first field to the right of **Request Canceled**, enter: `tw.local.isCompliant`
 ___ f. Change the evaluation value to: `==`
 ___ g. In the last field to the right, enter the value: "0"

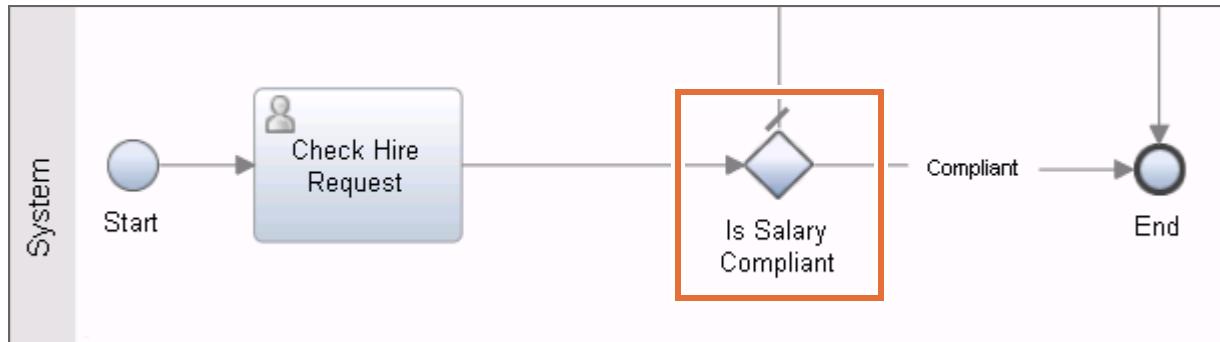


- ___ h. Save your work.
 ___ 4. Implement the **Is Salary Compliant** exclusive gateway in the **Approve Hire Request** linked process. When the salary is compliant, the flow must move to the end event.
 ___ a. Open the **Approve Hire Request** linked process from the History menu.



If you don't see the process in your history menu, open the Approve Hire Request from the library on the left.

- ___ b. On the **Definition** tab, select the **Is Salary Compliant** gateway.



Make sure that you select the **Is Salary Compliant** gateway, not the **Is Salary Acceptable** gateway.

- ___ c. In the **Decisions** section of the **Implementation** panel (not the Decisions panel) below, select **Not Compliant**, if not selected as the default flow.

This arrangement makes **Not Compliant** the default sequence flow. The order of sequence flow in the Implementation section ensures that the **Compliant** option is assessed first, and if the first assessment is false, the **Not Compliant** option is taken.

- ___ d. In the first field to the right of **Compliant**, enter: `tw.local.isCompliant`
- ___ e. Change the evaluation drop-down value to: `==`
- ___ f. In the last field to the right of **Compliant**, type: `"1"`

General	Decisions		
Decision	Compliant: <code>tw.local.isCompliant</code> <code>==</code> <input type="text" value="1"/> Default flow: <code>Not Compliant</code>		
Implementation			
Pre & Post			
Tracking			

- ___ g. Save your work.

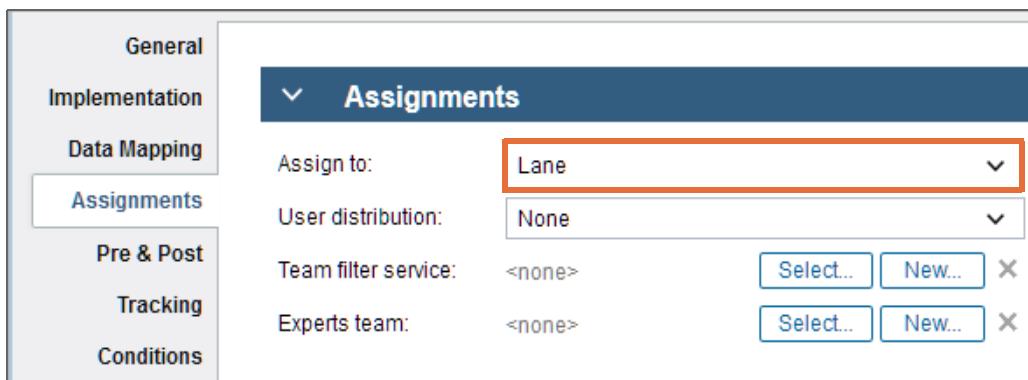
Part 5: Implement routing for an activity

Effective business process management is about routing the right task to the right teams at the right time. Using IBM Business Automation Workflow, authors use lanes (sometimes known as swimlanes for a pool) to identify the process teams. Process developers implement routing in the process for each activity to the right team in the lane. The goal of this section is to implement the teams for the lanes and implement assignment routing for all activities in the process.

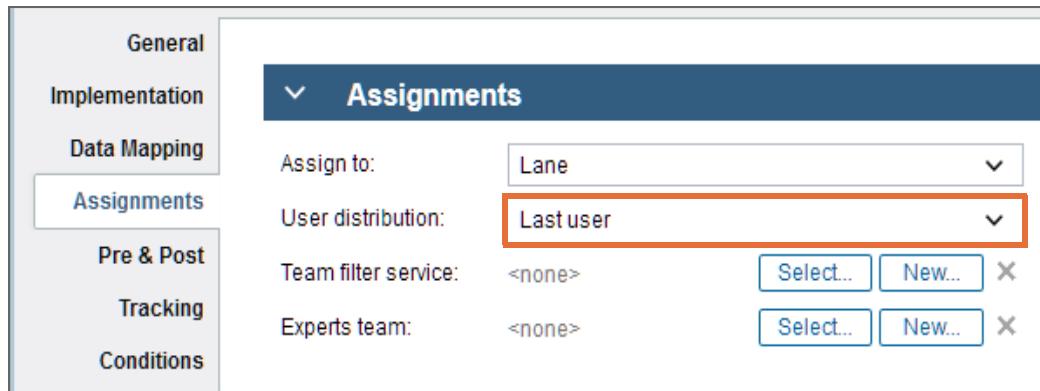
- ___ 1. Route the Submit Hiring Request activity in the Hiring Request Process to **Lane** and distribution to **Last User**.
 - ___ a. Open the **Hiring Request Process**.
 - ___ b. On the **Definition** tab, select the **Submit Hiring Request** activity.



- ___ c. Open the **Assignments > Assignments** section.
- ___ d. Verify that the default assignment for **Assign to** is: Lane



- __ e. Change the **User distribution** to: Last user



Information

The Last user option assigns the runtime task to a user who completed an earlier task.

- If the assignment option is set to Lane, the task is assigned to the user who completed the activity that immediately precedes the selected activity in the lane.
- If the assignment option is set to Team, the task is assigned to the user who completed the last task that was assigned to the same team.

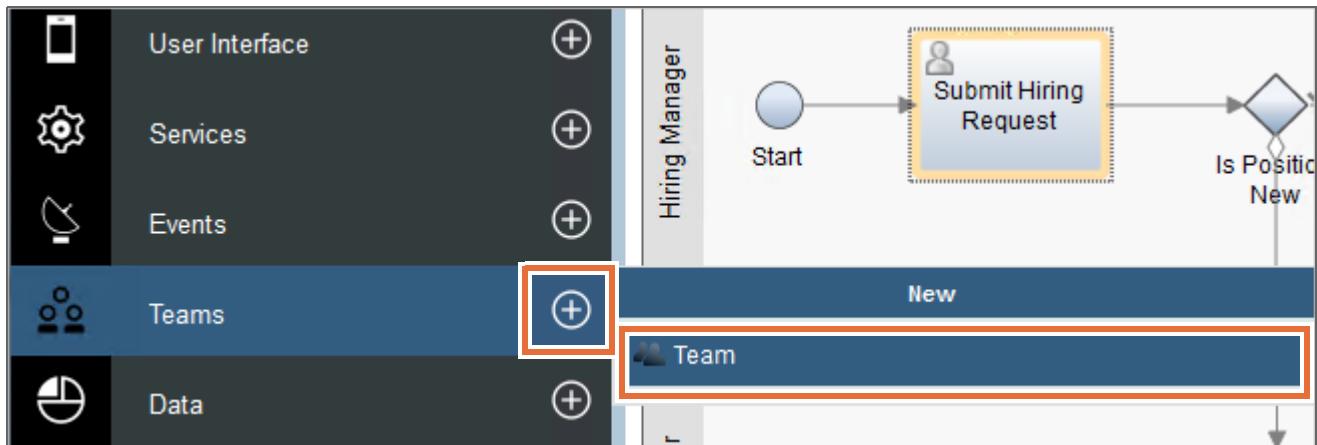
In both cases, the task is assigned only to an established last user, if the user is a member of the group that is associated with the task. An established last user is determined by using the policy valid for the Lane or Team assignment option. If the user is not a member of the group that is associated with the task, then the task is assigned to the task group.

Because this activity is the first in this lane for this process, the system automatically assigns the first task to the user that creates the instance. When a user creates an instance of this process, the server displays any screens that are configured for the task. You create those screens in the next playback phase.

-
- __ f. Save your changes.

__ 2. Create a team and add a member.

__ a. In the process library, click the (+) plus sign next to **Teams**. Click **Team**.



__ b. Name the team General Managers and click **FINISH**.

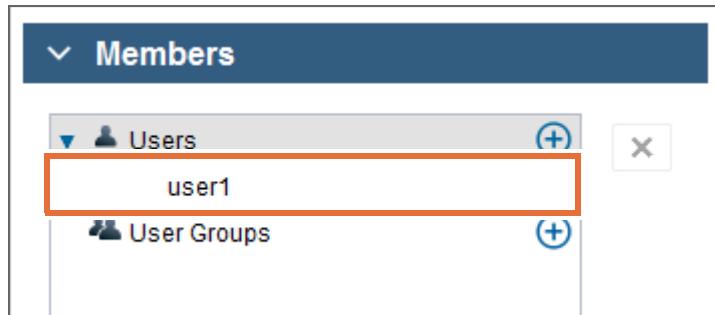
The screenshot shows the 'New Team' dialog box. It has a title 'New Team' and a description: 'Teams represent the teams of users in your enterprise that perform tasks. After you create a team, you can add users or teams as members, or define an expression to determine membership.' There is a 'Name:' field containing 'General Managers', which is highlighted with a red box. At the bottom are 'FINISH' and 'CANCEL' buttons, with 'FINISH' also highlighted with a red box.

__ c. In the **Members** section, click the (+) plus sign next to **Users**.

The screenshot shows the 'General Managers' team settings page. It has two main sections: 'Common' and 'Behavior'. The 'Common' section includes fields for 'Name' (General Managers), 'Modified' (author1 (Nov 15, 2017, 3:10:20 PM)), and 'Documentation'. The 'Behavior' section includes options to specify members by using 'Users or groups' (selected) or 'A service'. Below these sections is a 'Members' section with a table. The 'Users' row has a '+' button highlighted with a red box. The table also includes a 'User Groups' row with a '+' button.

- __ d. Add `user1` to the team from the list.

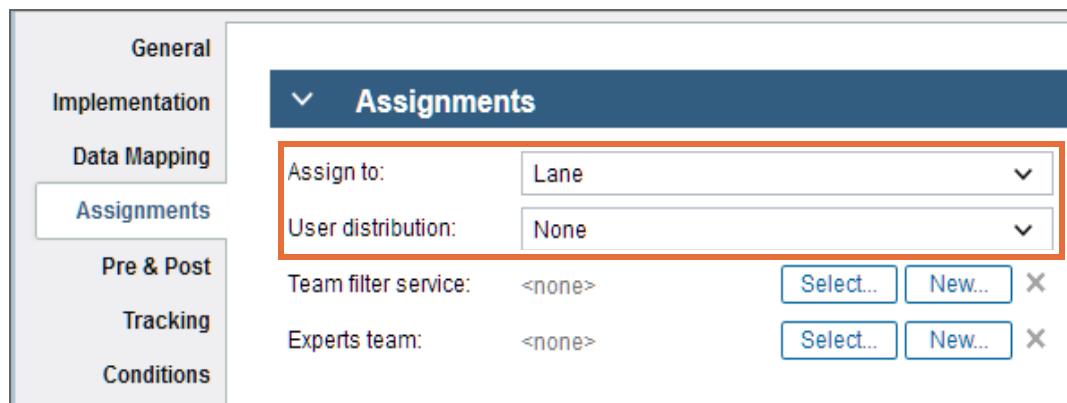
You now have a populated list in the **Members** section with `user1` for the General Managers team.



- __ e. Save your work.

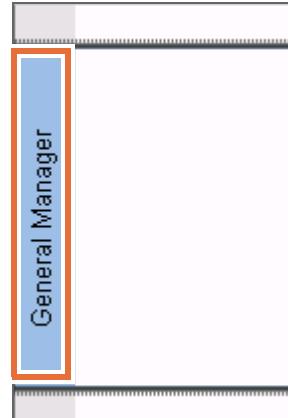
- __ 3. Implement the routing for **Approve New Hire Request**.

- __ a. Open the **Hiring Request Process**.
- __ b. On the **Definition** tab, select the **Approve New Hire Request** activity.
- __ c. In the panel below, in the **Assignments > Assignments** section, leave the value of **Assign to:** as `Lane` and the value of **User distribution** as `None`.

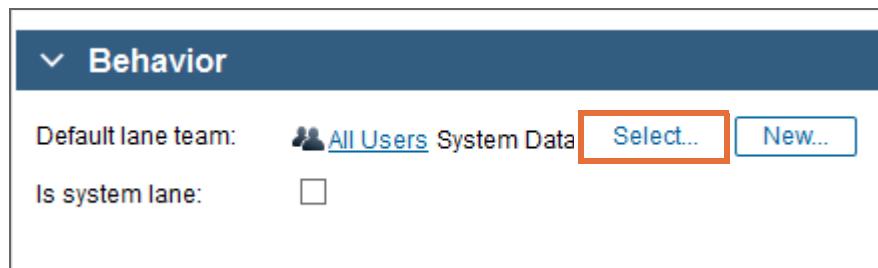


The default **User distribution** option is `None`. IBM Business Automation Workflow does not assign the task to any user when using this option. All the other options in this selection assign the task to a user of the system.

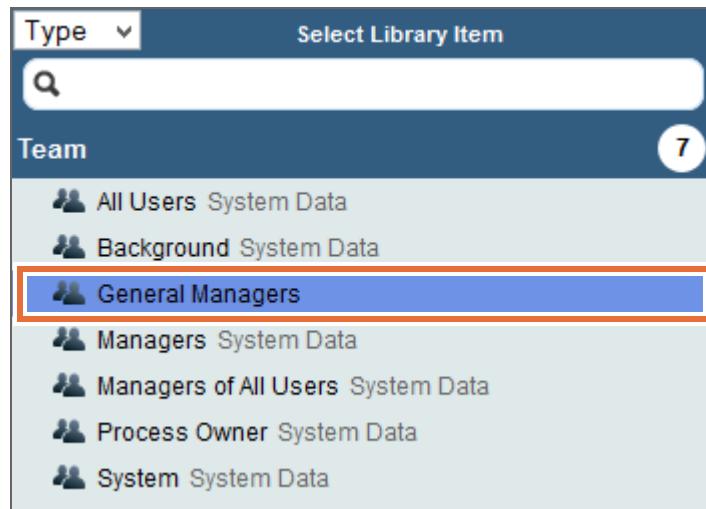
- __ d. On the General Manager lane, click the **label** on the left.



- __ e. In the panel below, go to the **General > Behavior** section, click **Select** next to Default lane team.



- __ f. Click the **General Managers** team to select it.



The **General Managers** team is assigned as the default team for the **General Managers** lane.

Behavior

Default lane team: **General Managers**

Is system lane:

- ___ 4. Assign the **General Managers** team as the **Experts Group** for the Approve New Hire Request activity.
- ___ a. On the **Hiring Request Process**, select the **Approve New Hire Request** activity.
- ___ b. Open the **Assignments > Assignments** section. Click **Select** next to the Experts team.

General

Implementation

Data Mapping

Assignments **▼** **Assignments**

Assign to: Lane

User distribution: None

Team filter service: <none>

Experts team: <none> **Select...**

- ___ c. Click **General Managers** to assign the General Managers team as the Experts team.

General

Implementation

Data Mapping

Assignments **▼** **Assignments**

Assign to: Lane

User distribution: None

Team filter service: <none>

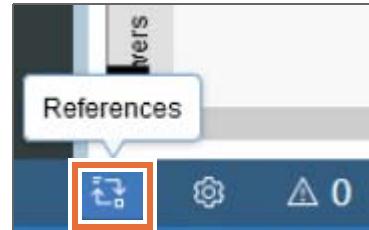
Experts team: **General Managers**

- ___ d. Save your work.

Part 6: Review artifact references

As you add more artifacts to your solution, it is a good time to review artifact references. When you work with an artifact in the Process Designer, it is useful to know where the artifact is used so that you can see the impact of potential changes that you want to make.

- ___ 1. Review the local scope references for the Hiring Request Process artifact.
- ___ a. With the Hiring Request process open, click the **references icon** in the status bar at the bottom.

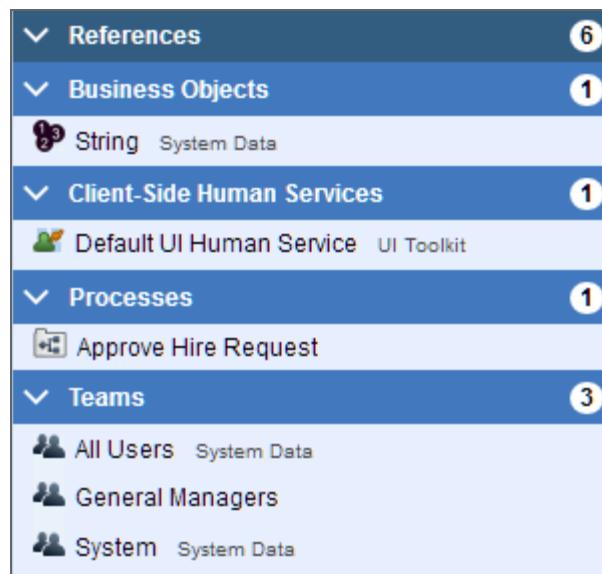


- ___ b. In the References pane, review the **local scope** references.



Local scope shows where artifacts are used only in the current process application and its dependencies. This is the default view. Currently, you can see that there are no artifacts in the process model with a local scope.

- ___ c. In the References pane, review the remaining **references**.

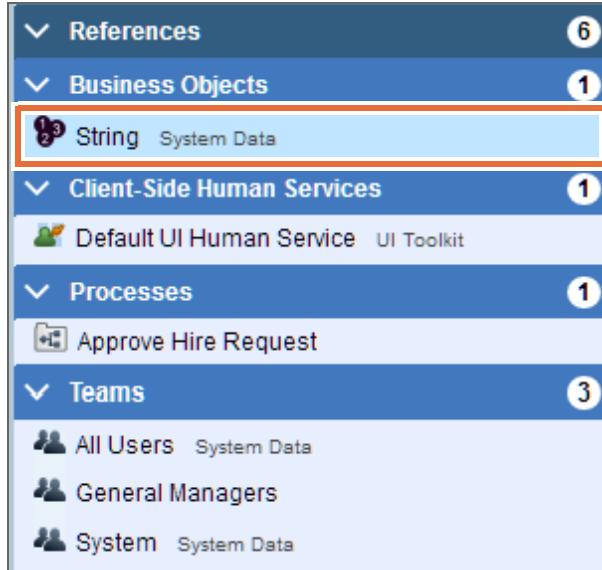


Depending on timing, it might take a minute or two for the General Managers team to display.

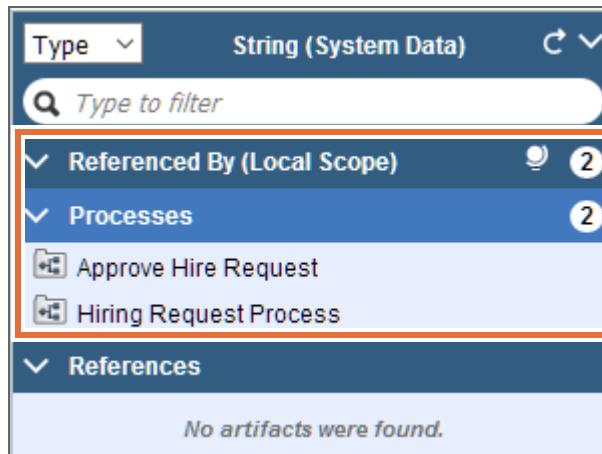
Specifically, the list shows the references to the system toolkit artifacts. For example, since you defined a variable to pass between the parent and nested processes, the String object is referenced.

The numbers on the right are the instances in the project. You should see the same number having followed the lab instructions. If not, then your are missing something in the lab.

- ___ d. Review local references of the String artifact by clicking the **String section**. You might need to refresh the References section to see the new team.

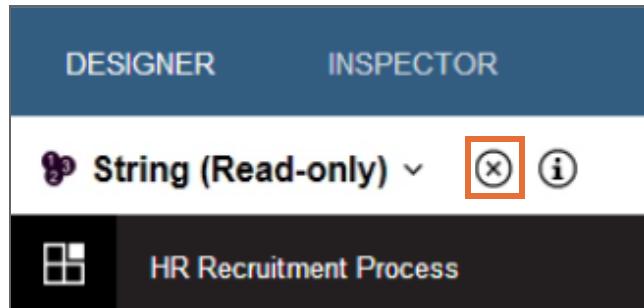


- ___ e. For the String references, there are two local references that point to the two process models that use the string variable.



- ___ f. Close the references panel.

- __ g. Close the String (Read-only) window.



The String object is read-only since it is a reference to the System Toolkit.

You have completed this exercise.

This exercise looked at setting variables in business processes, implementing an intermediate timer event on a process, implementing exclusive decision gateways, and establishing routing for an activity. You also evaluated artifact dependencies by reviewing the references for some of the artifacts.

The assets for Playback 1: Controlling process flow, are complete. You now can control gateways by assigning values to variables inside the process. The next exercise looks at building business objects and creating a reusable client-side human service.

End of exercise

Exercise 5. Playback 1: Business data, services, and coaches

Estimated time

01:30

Overview

In this exercise, by using the core requirements, you determine and create all of the necessary assets to support a coach in the Hiring Request Process. You use complex business objects (variable types) to organize your data, and pass data into and out of a linked process. You build a service and define guided user interactions with a coach. You also implement a service for an activity, and map variables between a nested service and an activity. You model the coach by using the concept of grids.

Objectives

After completing this exercise, you should be able to:

- Determine and organize data when provided with a written process
- Add business objects and object types
- Create a client-side human service
- Add variables and business objects to a process application
- Create and configure a coach to obtain process participant input
- Model a coach by using the concept of grids
- Add coach controls to control process flow
- Create a client-side human service and coach for the General Manager review activity
- Implement an activity by attaching a service and mapping data

Introduction

In Playback 1, you create human services. Users can use human services to access task assignments in a web-based user interface. At this stage of development, it is important that the functions of the human service are implemented. Users interact with web-based interfaces called coaches to complete their assigned tasks. Enhancements can be added later. This exercise is about making sure that users have what they need in terms of business data and task assignment information to complete the process activity.

The human services with coaches are implemented on two activities in the Hiring Request Process process. The coaches are used during playbacks to provide business data to the process and control the process flow.

This exercise and the next provide practice in building and moving controls on a coach. Repetitive steps are purposely provided to gain experience in this skill.

This exercise is long, so a break is included in the middle of the instructions.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Build business objects and variables](#)

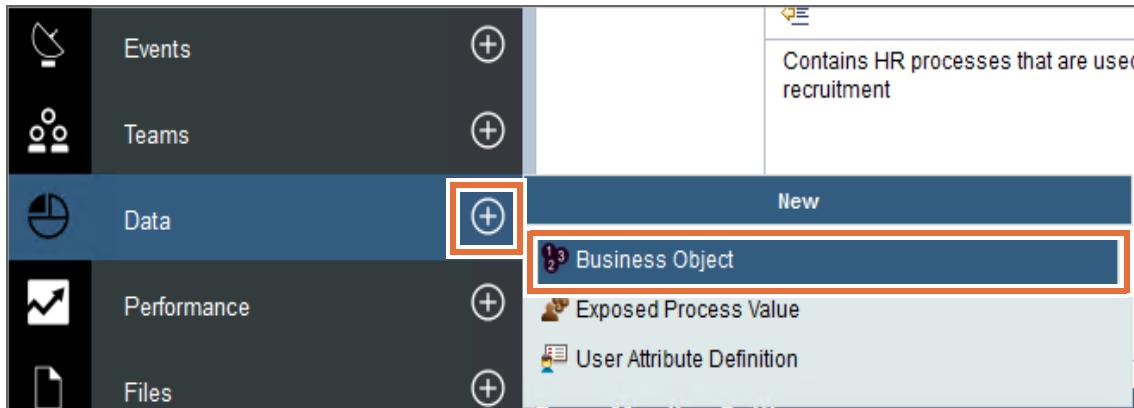
[Part 2: Create a reusable client-side human service and define input controls](#)

[Part 3: Reuse and debug the client-side human service](#)

Exercise instructions

Part 1: Build business objects and variables

- 1. In this section, you create a complex variable (business object) to hold the position data. Then, you use this variable on the coach to allow users to input the position data. Later in the lab, you map these variables to move the data through the process.
 - a. Open the **HR Recruitment Process** process application.
 - b. In the library, click the (+) plus sign next to the Data category, and select **Business Object**.



- c. Name the business object **Position** and click **FINISH**.

New Business Object

Business objects represent the data that is used in an application. You can create complex business objects with multiple parameters of different variable types to store the data, for example, that a user would enter in a form.

Name:

FINISH **CANCEL**

Always follow the naming convention for the variable names. Variable names are case-sensitive. Capitalize the first letter when creating business objects (for example, Employee), but use camel case for the parameters or instantiation of the variable (for example, employeeId). Camel case refers to a word or string of letters that has no space and has an uppercase letter in a position other than the first letter.

- ___ d. You now see the settings page for the business object. Add a parameter to the business object by clicking the **(+)** plus sign next to Parameters.



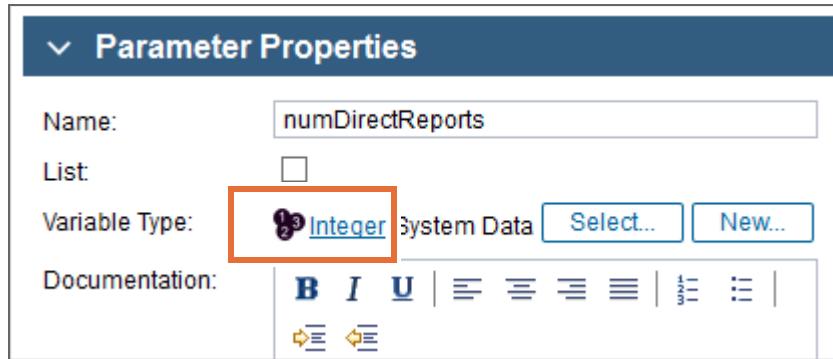
- ___ e. Change the name of the parameter to: jobTitle
___ f. Keep the variable type as **String**.

Parameters		Parameter Properties	
Parameters jobTitle (String)	+ ↑ X ↓	Name: jobTitle List: Variable Type: String Documentation:	System Data Select... New...

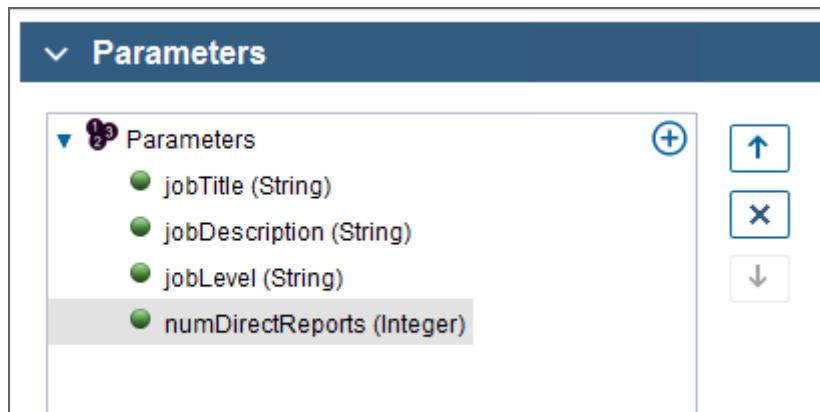
- ___ g. Repeat steps 1d – 1f to add the following parameters:
- jobDescription (String)
 - jobLevel (String)
- ___ h. Add another parameter that is named: numDirectReports
- ___ i. Change the type of the numDirectReports attribute to an Integer. Click **Select** to the right of Variable Type. In the filter, enter int. You see the Integer variable type. You can filter the variable types by typing the letters of the variable type you are looking for. Select the Integer variable type.

Type	Business Object
int	Integer System Data

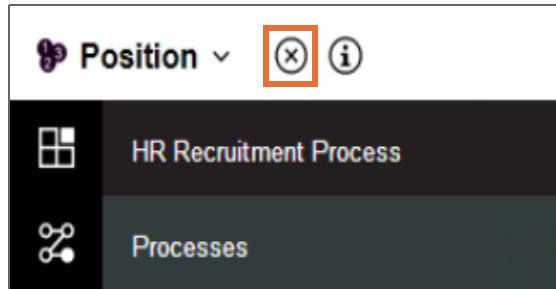
- j. Verify that the Integer variable type is selected.



- ___ k. Save your work. Your business object has four parameters.



- ___ l. Save your work and close the business object.



- ___ 2. Create a second business object that is named `CompensationDetails` to allow users to input compensation data on a coach.
- ___ a. Click the **(+)** plus sign next to the Data category in the library, and select **Business Object**.

- ___ b. Name the business object `CompensationDetails` and click **FINISH**.

New Business Object

Business objects represent the data that is used in an application. You can create complex business objects with multiple parameters of different variable types to store the data, for example, that a user would enter in a form.

Name: `CompensationDetails`

FINISH **CANCEL**

- ___ c. Add the following parameters for the complex variable `CompensationDetails`. Remember to change the variable type by clicking **Select** to the right of **Variable Type** to change it to Decimal rather than the default (String):
- `salaryToOffer` (Decimal)
 - `bonusAmount` (Decimal)

Parameters

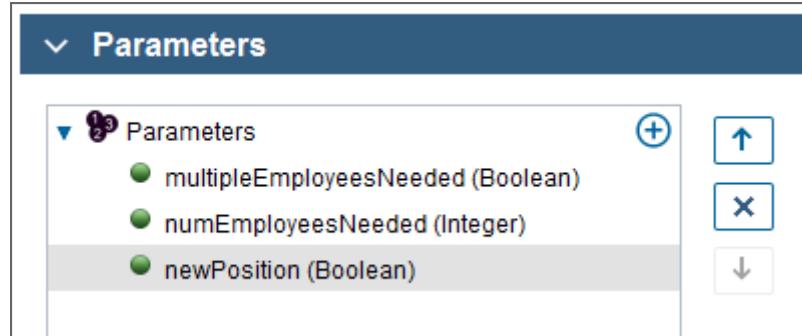
Parameters

- `salaryToOffer` (Decimal)
- `bonusAmount` (Decimal)

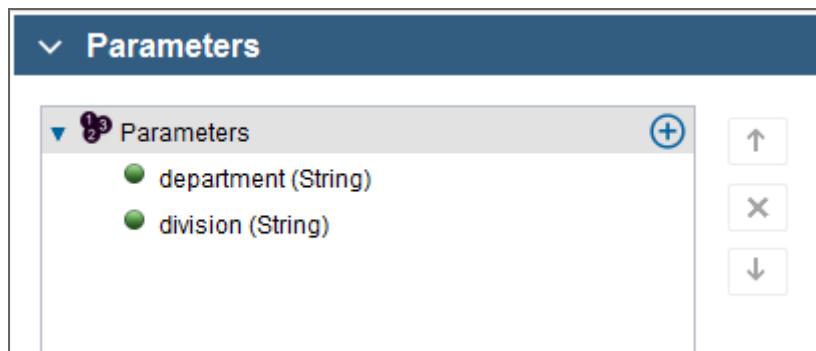
FINISH **CANCEL**

- ___ d. Save your work and close the business object.
- ___ 3. Create a business object that is named `RecruitingDetails` to allow users to input recruiting information on a coach.
- ___ a. Create a business object: `RecruitingDetails`
- ___ b. Include the following parameters in the complex variable:
- `multipleEmployeesNeeded` (Boolean)
 - `numEmployeesNeeded` (Integer)

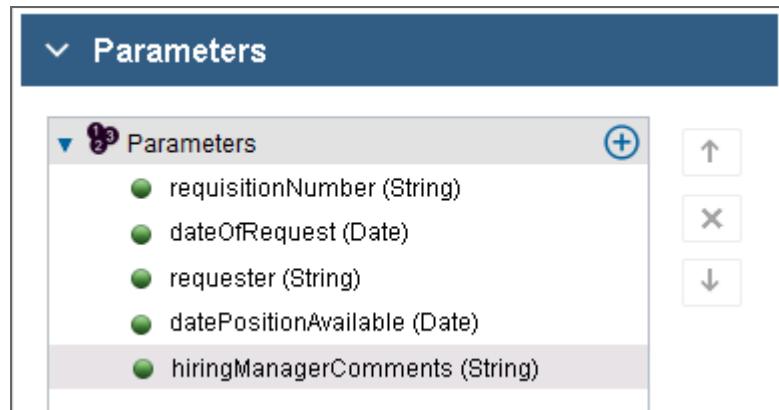
- newPosition (Boolean)



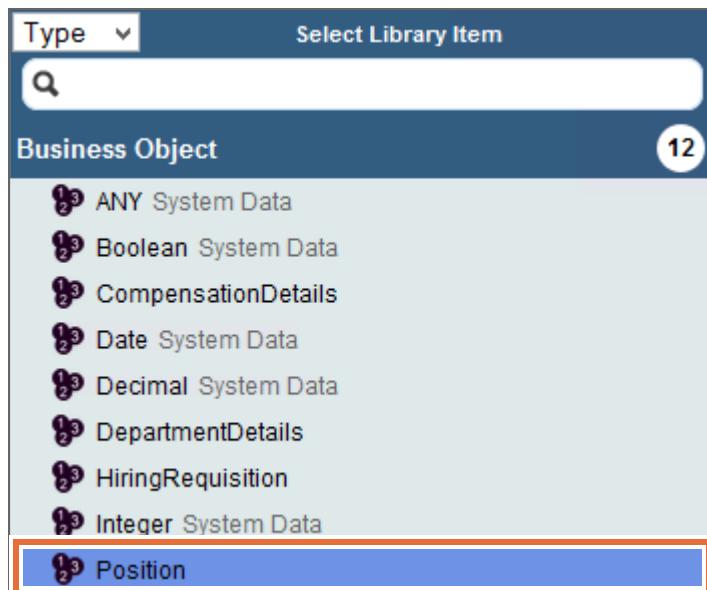
- __ c. Save your work and close the business object.
- __ 4. Create a fourth business object that is named `DepartmentDetails` to allow users to input department data on a coach.
 - __ a. Create a business object: `DepartmentDetails`
 - __ b. Include the following parameters in the complex variable:
 - department (String)
 - division (String)



- __ c. Save your work and close the business object.
- __ 5. Create a business object that is named `HiringRequisition` to save data about the requisition. Model this variable to hold the other objects you created. Use this variable as the main parent business object to hold all the data for the process.
 - __ a. Create a business object: `HiringRequisition`
 - __ b. Add the following parameters:
 - requisitionNumber (String)
 - dateOfRequest (Date)
 - requester (String)
 - datePositionAvailable (Date)
 - hiringManagerComments (String)



- ___ c. Add a parameter that is called `position`. This time, click **Select** to the right of **Variable Type** and use the **Position** business object that you created.



- d. Verify that the variable type is capitalized and the variable name starts with a lowercase letter. Recall that business objects are capitalized and variables are not.



- ___ e. Continue to add the other business objects as parameters:

- compensationDetails (CompensationDetails)
- departmentDetails (DepartmentDetails)
- recruitingDetails (RecruitingDetails)

The screenshot shows a 'Parameters' dialog box with a list of parameters. The parameters are listed in a tree view:

- Parameters
 - requisitionNumber (String)
 - dateOfRequest (Date)
 - requester (String)
 - datePositionAvailable (Date)
 - hiringManagerComments (String)
 - position (Position)
 - compensationDetails (CompensationDetails)
 - departmentDetails (DepartmentDetails)
 - recruitingDetails (RecruitingDetails)

On the right side of the dialog box, there are four buttons: a plus sign (+) button, an up arrow button, a delete (X) button, and a down arrow button.



Information

Use the up and down arrows to move parameters. The order of the parameters is important. As you add the business objects to your coach, the parameters are sequenced according to where they are in the list.



- ___ f. Save your work and close the business object.
- ___ 6. Add the `requisitionDetails` (`HiringRequisition`) variable to the Approve Hire Request linked process as an input and output variable.

To pass data from the parent process (`Hiring Request Process`) to your linked process (`Approve Hire Request`), you must create and map variables. Because the process is a linked process, you must pass the data object to the linked process by way of the input and output variable settings in the linked process. Variable mapping is required for linked processes because they are a separate process from the parent process.

- ___ a. Open the **Approve Hire Request** linked process.
- ___ b. Click the **Variables** tab.

The screenshot shows a process details interface with several tabs at the bottom:

- Overview
- Definition
- Variables** (highlighted with a red border)
- Folders

- ___ c. Add an Input variable `requisitionDetails` (`HiringRequisition`) to the process. After adding the variable, you can expand the `requisitionDetails` to view its contents.

The screenshot shows the 'Variables' dialog box with the 'Input' section expanded. A variable named 'requisitionDetails' is selected and highlighted with a red border. This variable is of type 'HiringRequisition' and contains the following sub-fields:

- requisitionNumber (String)
- dateOfRequest (Date)
- requester (String)
- datePositionAvailable (Date)
- hiringManagerComments (String)
- position (Position)
- compensationDetails (CompensationDetails)
- departmentDetails (DepartmentDetails)
- recruitingDetails (RecruitingDetails)

- ___ d. Add an output variable `requisitionDetails` (`HiringRequisition`) to the process.

Important

When you type in the output name, the variable is immediately flagged because you have an input and an output with the same name, but different variable types. The system defaults the variable to a String. When you set the variable type to match the type of the input variable, the error disappears. This can occur anytime an incomplete definition is made including data mapping.

The screenshot shows the 'Details' dialog box. In the 'Name:' field, the value 'requisitionDetails' is entered and highlighted with a red border, accompanied by a red X icon. Below the name field is a 'Documentation:' field and a rich text editor toolbar with buttons for bold (B), italic (I), underline (U), and alignment.

- ___ e. You added the variables to the subprocess. Save your work.

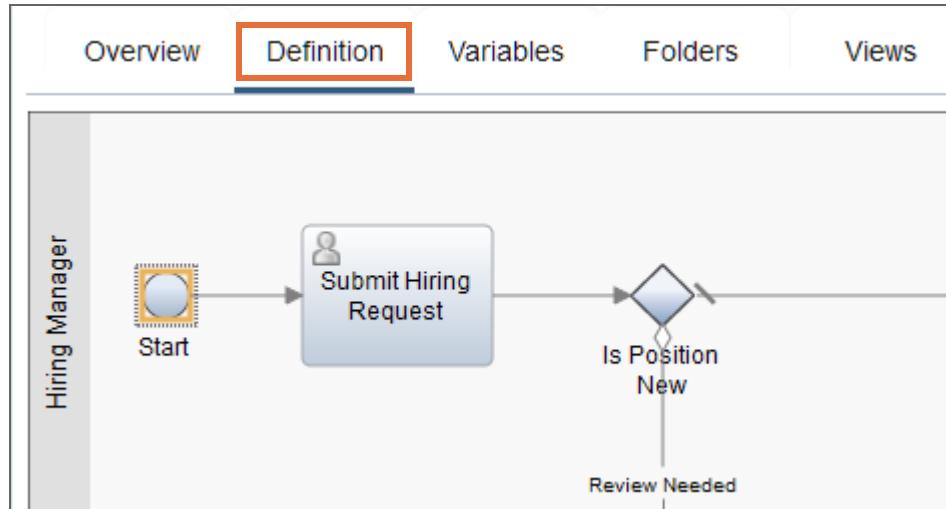


Make sure that the input and output variable names exactly match. If the names do not match, then you have two different variables in the linked process. The input data to the process is not sent as output to the parent process unless you copy the contents of one variable into the other. Copying and pasting the variable name is a good practice.

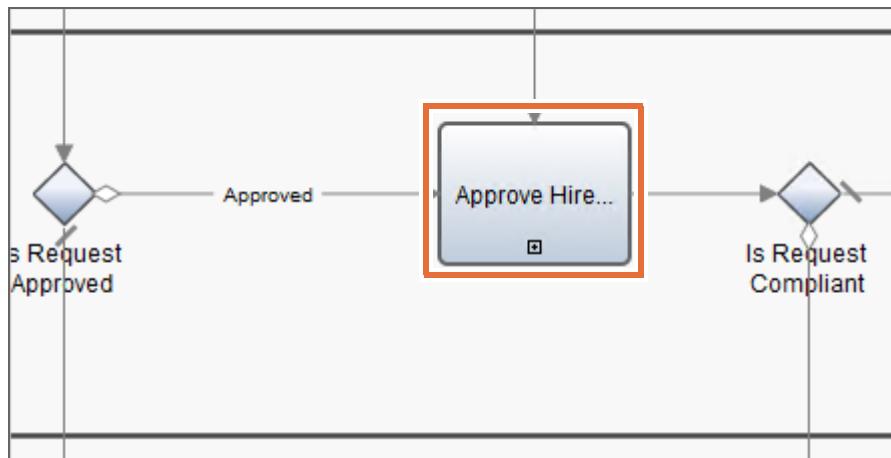
- ___ 7. Create and map the requisitionDetails variable from the higher-level process to the linked process.
- ___ a. Open the **Hiring Request Process** and click the **Variables** tab.
 - ___ b. Create a private variable: requisitionDetails (HiringRequisition)



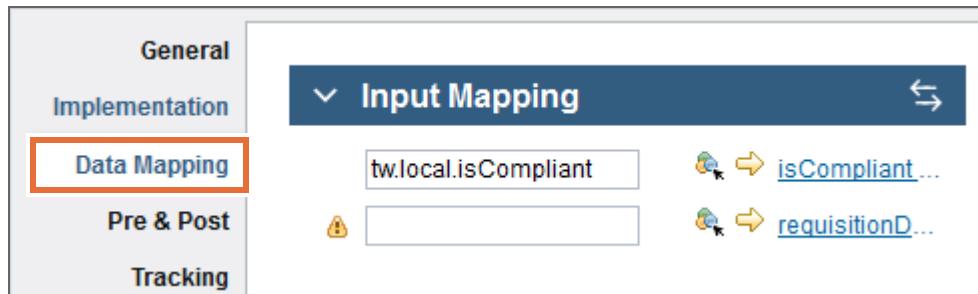
- ___ c. Click the **Definition** tab.



- ___ d. Click the **Approve Hire Request** linked process.



- ___ e. Click the **Data Mapping** menu in the Properties panel below.



- ___ f. Map `tw.local.requisitionDetails` to both the **Input Mapping** and **Output Mapping** sections. Place the cursor in the box next to `requisitionDetails` (`HiringRequest`) and press the **Ctrl+Space** to view the auto complete options. Select **requisitionDetails - HiringRequisition**.

- ___ g. Verify your data mapping and save your work.



Optional

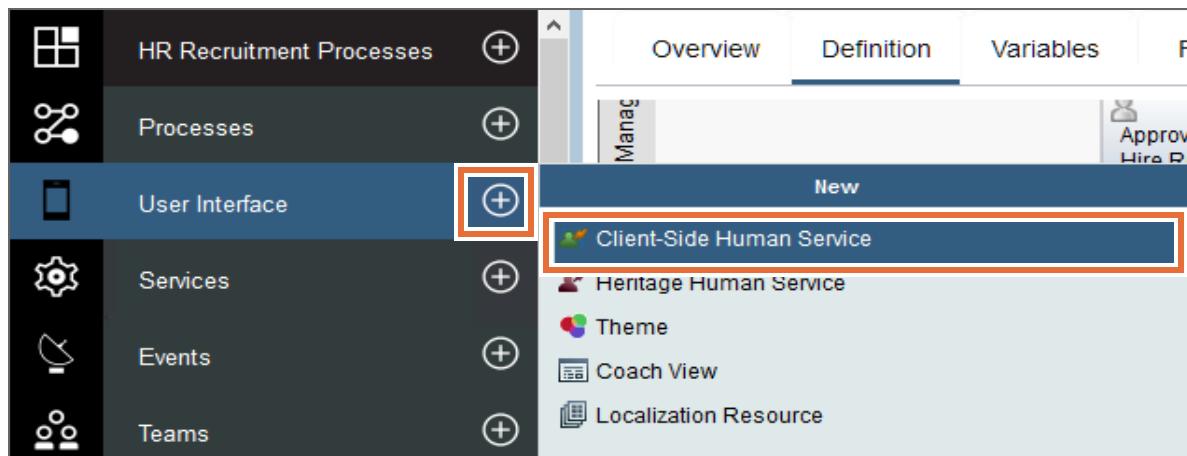
You have completed part 1 of 3, so you are now approximately a third done with this exercise. The next section is large, so now would be a good time to take a 10-minute break.

Part 2: Create a reusable client-side human service and define input controls

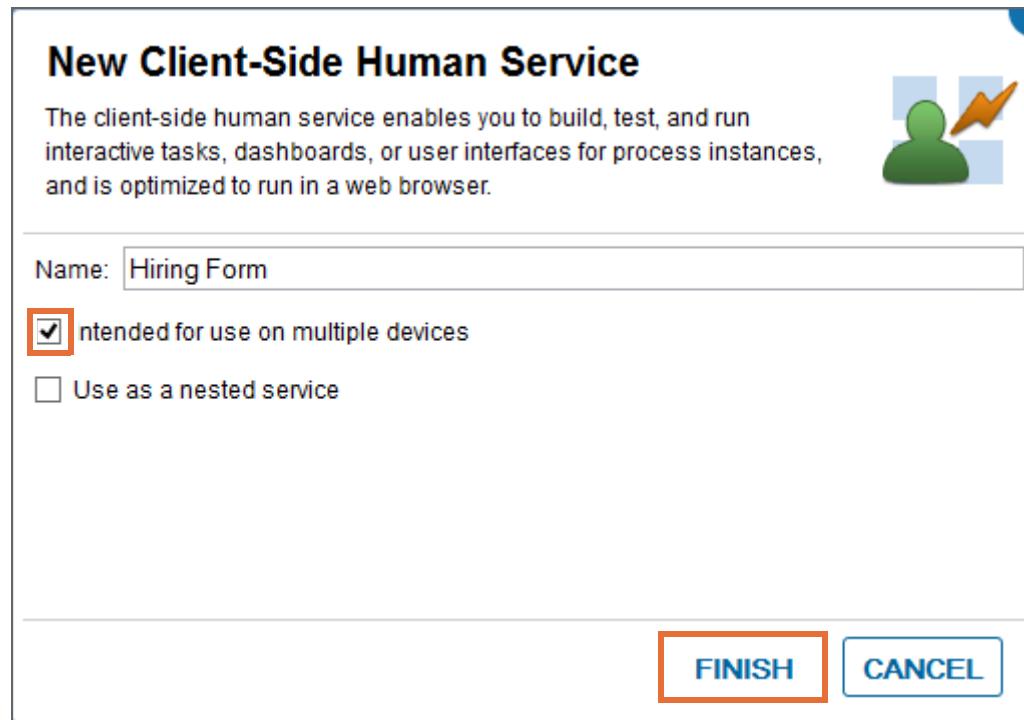
Up to this point, each activity is implemented by using the Default UI Human Service. In this section, you define and implement custom services and coaches for activities in the process. In the next section, you create a coach to collect information from process teams. The custom service passes the collected information in the coach to the process.

Two activities on the process need a coach that has the same fields: the Submit Hiring Request and the Approve Hiring Request. One requires the ability to input data, and the other needs the data to be read-only with an approval mechanism. In this part of the exercise, you create a single coach for both activities. Because the same data can be shown for multiple activities on the process, you create a coach that can be reused for data input and viewing data (read-only).

- ___ 1. Create a client-side human service.
- ___ a. In the Process Designer library, click the (+) plus sign next to User Interface, and select **Client-Side Human Service**.

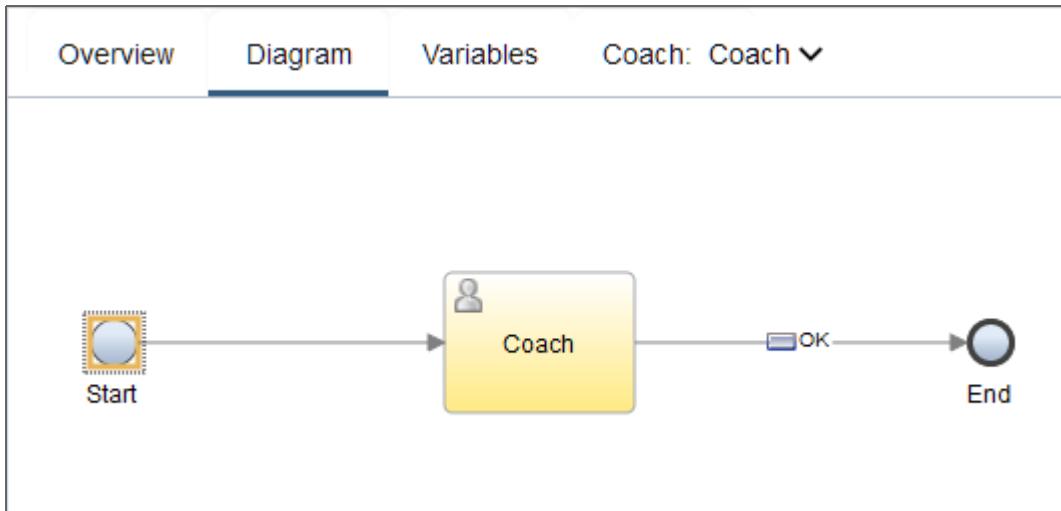


- ___ b. Name the client-side human service: **Hiring Form**. Select the **Intended for use on multiple devices** checkbox and click **FINISH**.



Intended for use on multiple devices enables the new service to be used on multiple device types. With this option, you can use responsive controls for all the new coaches and coach content that you add to the service.

- ___ c. The new human service opens in the Process Designer editor.

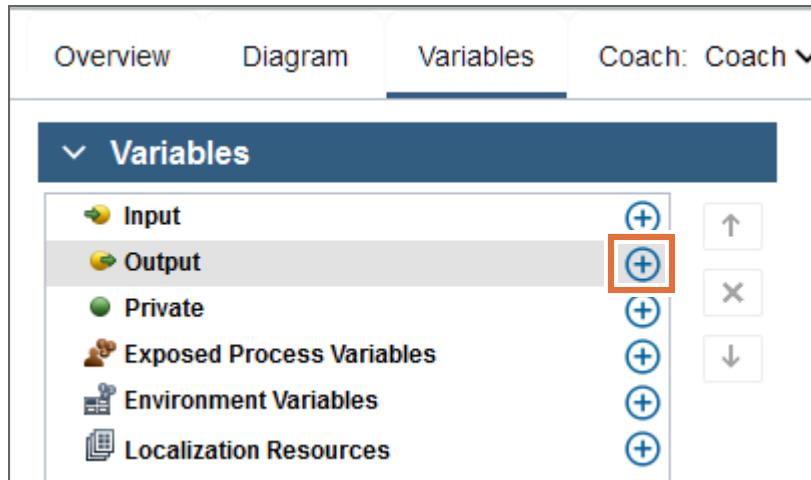


- ___ 2. Add the `requisitionDetails` output variable to the human service.

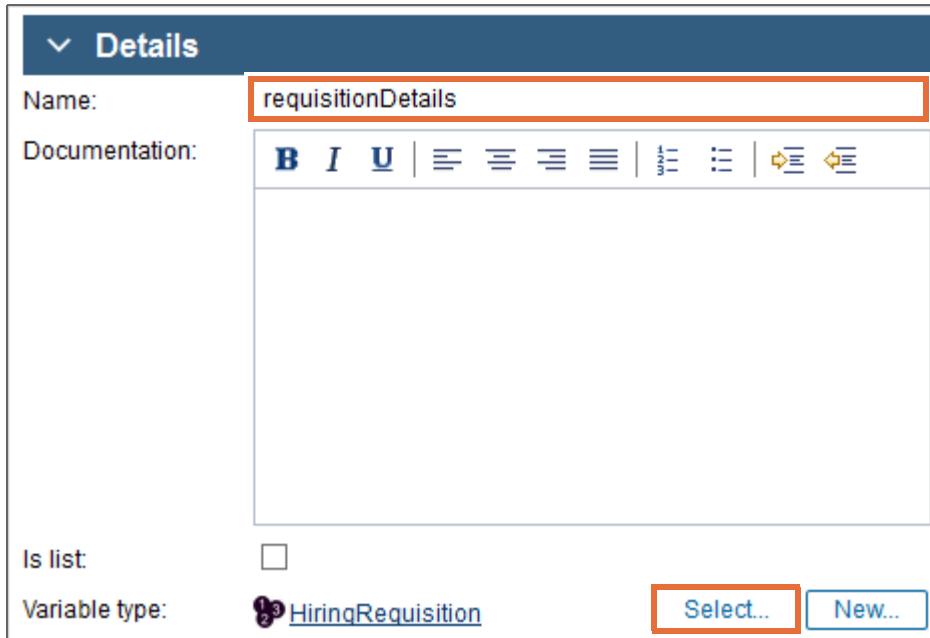
- ___ a. In the **Hiring Form** service, click the **Variables** tab.



- ___ b. Click the (+) plus sign next to Output.



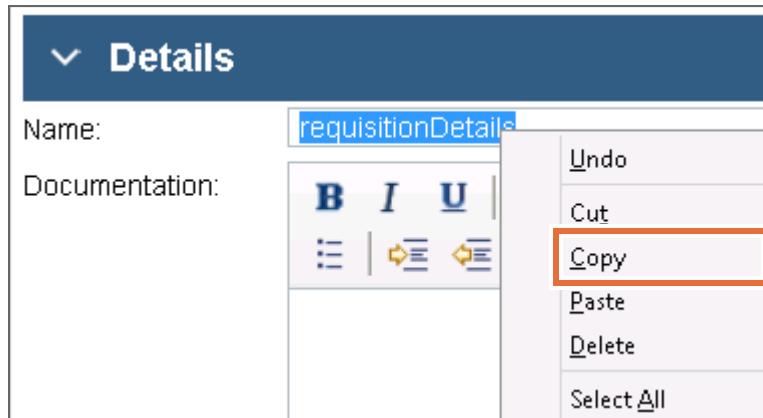
- ___ c. Name the variable: requisitionDetails. Click Select to the right of the Variable type option to select HiringRequisition.



Because you need all of the parameters of the **HiringRequisition** object, you use the full object in the coach. If you needed only part of the object, include that parameter and map the parameter from the parent object when mapping inputs and outputs.

- ___ d. Save your work.

- ___ 3. Because you reuse this coach for multiple activities, add matching input variables to the human service. If any data is entered as part of the process instance from the first activity, you can present it on the coach.
 - ___ a. Click the output variable `requisitionDetails`, highlight the name, and copy it.



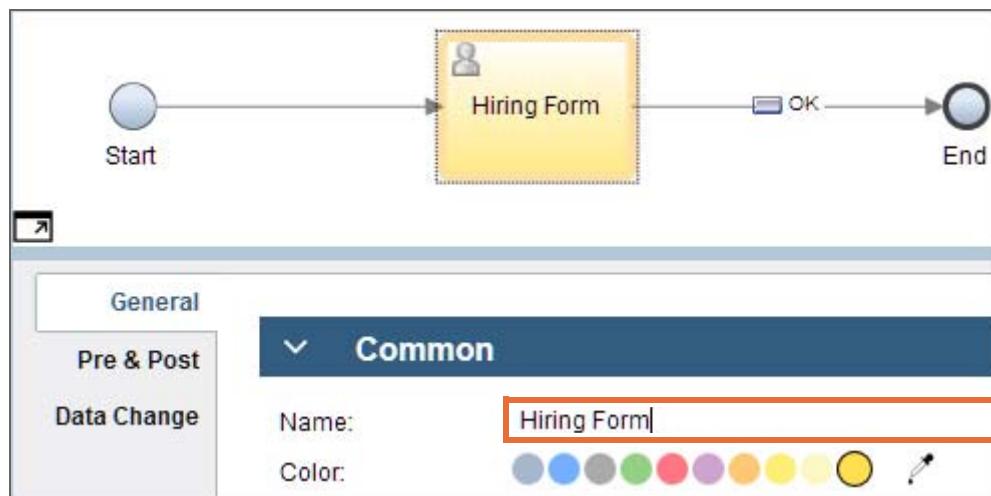
- ___ b. Click the **(+)** plus sign next to Input and paste `requisitionDetails` into the **Name** field.
- ___ c. Click **Select** to the right of the Variable type option.
- ___ d. Change the type to **HiringRequisition**.



- ___ e. Save your work.
- ___ 4. Now that the variables are created, add the inputs that are bound to the variables in the coach.
 - ___ a. In the Hiring Form service, click the **Diagram** tab.



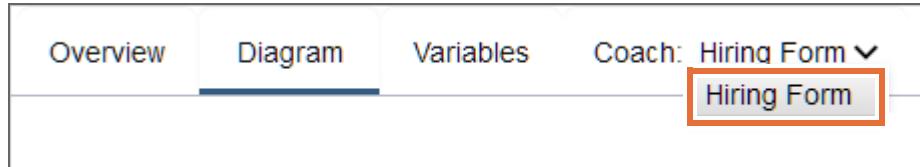
- ___ b. Click the **Coach** node on the canvas, and in the **General > Common** section, change the name of the coach to: **Hiring Form**



- ___ c. Save your work.
- ___ 5. Build the coach by using a two-column grid and add the Requisition Number and Requester variables.

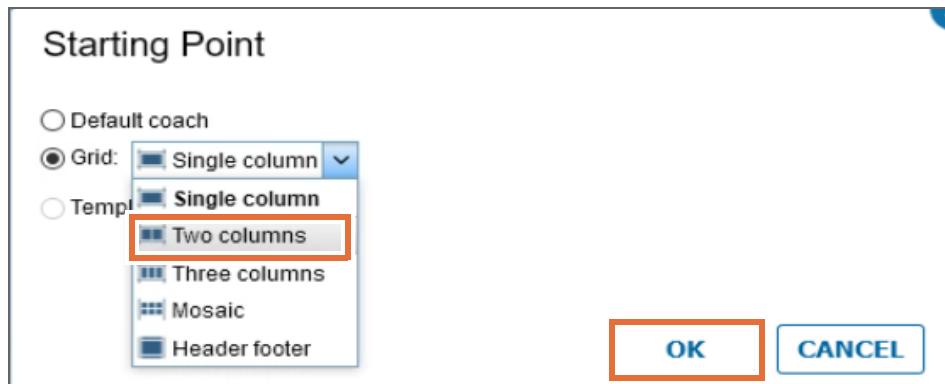
Build the first section that contains all the simple variables from the Requisition Details object. Consider how items are laid out. This consideration determines whether the coach is laid out vertically or horizontally.

- ___ a. For the Hiring Form service, click the down arrow next to **Coach: Hiring Form** and select **Hiring Form**.

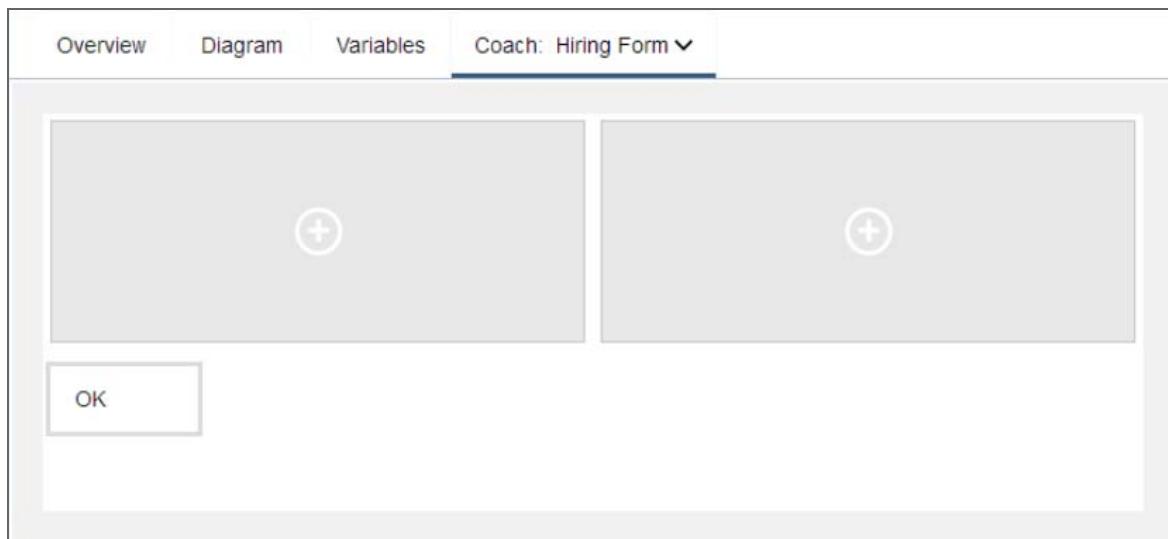


When you renamed the Coach in the diagram, the name changed at the top from Coach to Hiring Form

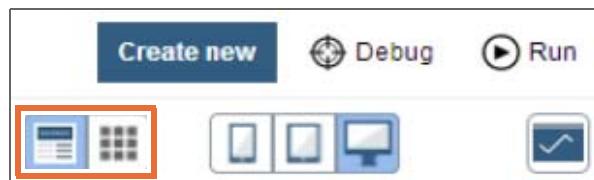
- ___ b. A message box for setting a Starting Point appears. Click **Grid** and in the drop-down, select **Two Columns** and click **OK**.



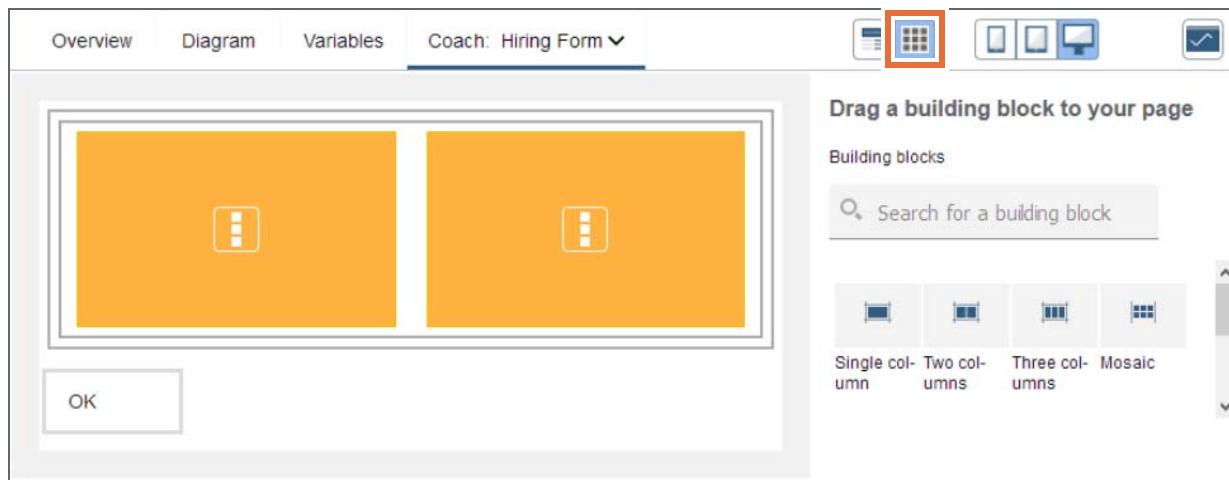
A two-column grid appears for the **Hiring Form** coach layout. This layout is the layout that you start from to build your coach.



The grid layout and content layout are mutually exclusive. When in grid view, you can arrange grids. When in content view, you cannot arrange grids but you can add content to each grid cell. The Content / Grid toggle at the top highlights which is displayed. The color of the grids might vary slightly depending on your monitor settings.

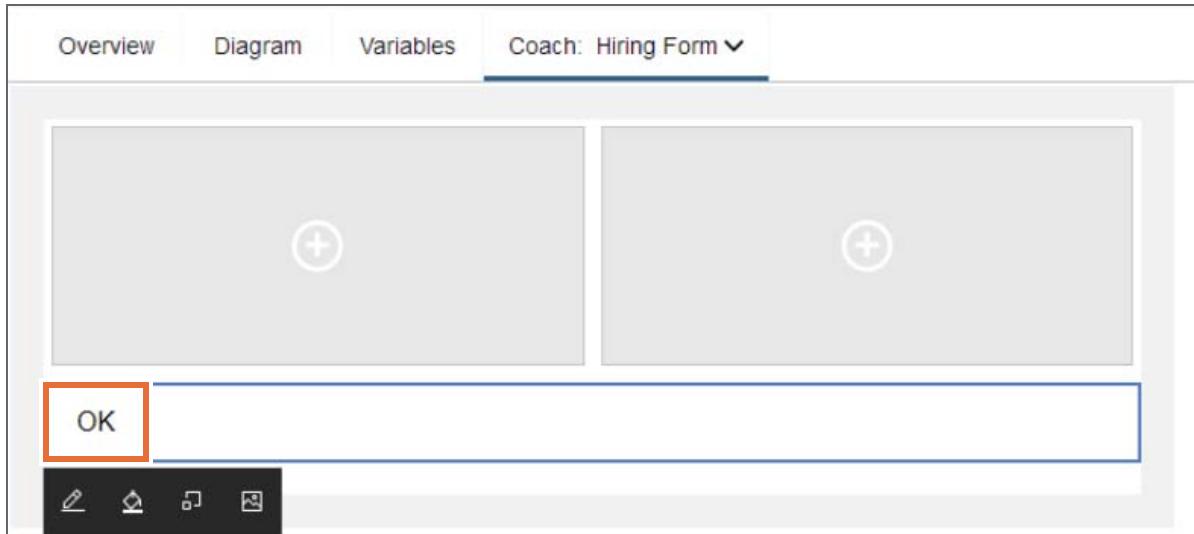


- ___ c. Click the **Grid toggle** to view the grid layout.

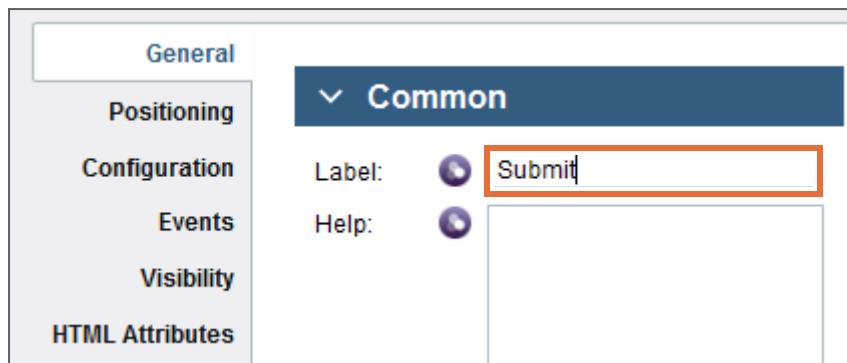


- ___ d. Click the **Content toggle** to view the content layout.

- __ e. Click **OK** on the coach canvas to select the button.



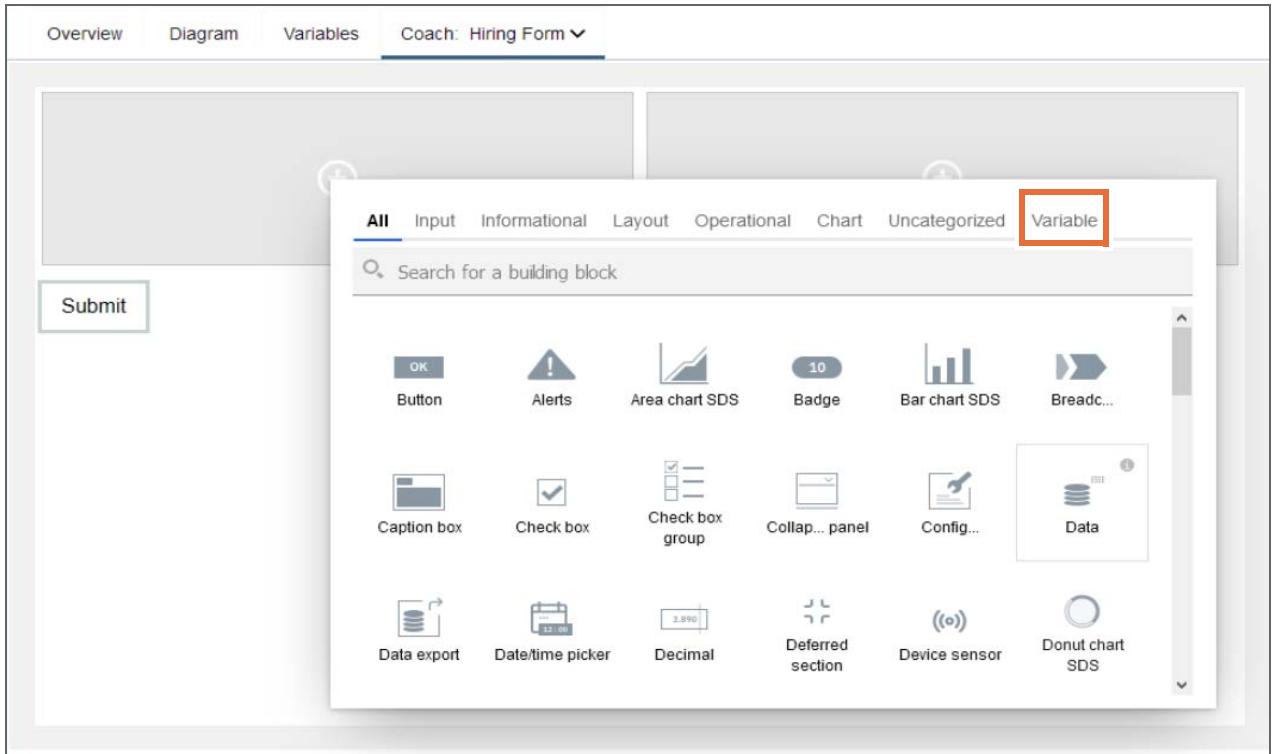
- __ f. In the **General > Common** section, change the label of the control button to: **Submit**



- __ g. The first inputs that you add to the coach are the parameters of the `requisitionDetails` variable. In the left column of the grid, click the white (+) plus sign in the middle.



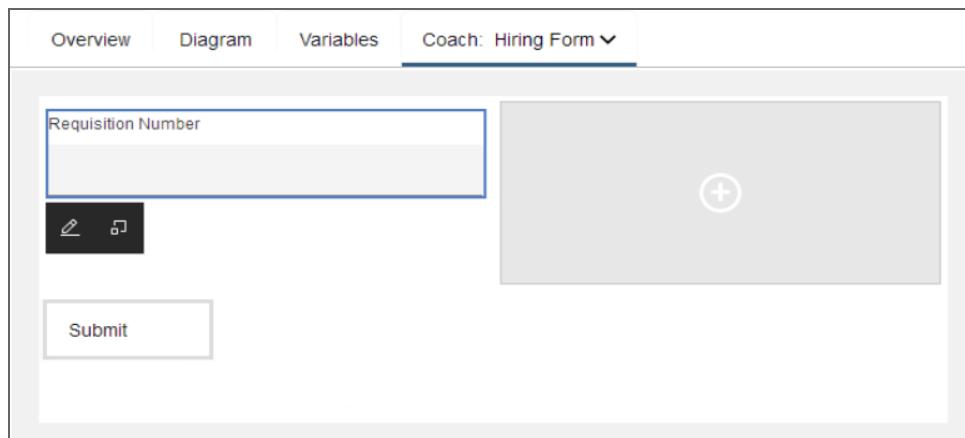
__ h. In the newly opened window, click the **Variable** tab.



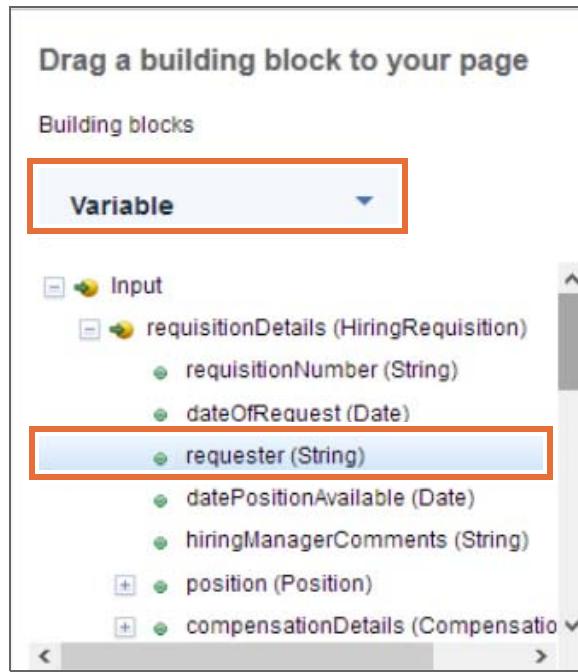
__ i. Select **Input > requisitionDetails > requisitionNumber**



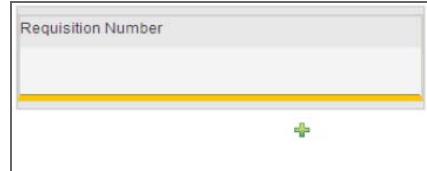
The result is that the requisition number is placed in the left column of the grid and is highlighted.



- __ j. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > requester**



- __ k. While holding down the left mouse button, drag the `requester` variable underneath the **Requisition Number** input control. Let go of the mouse button when the bottom border displays a gold highlight.

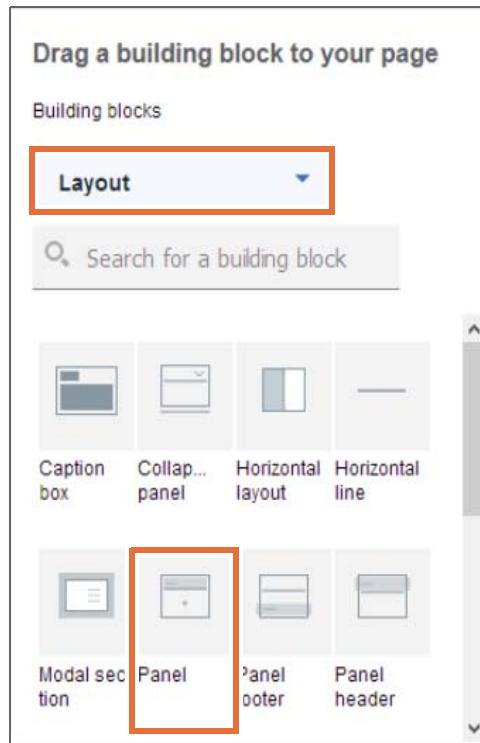


A **green plus (+)** is displayed when you are in an area that the item can be dropped. A **red circle with an (x)** appears if the mouse is hovering in an area where it cannot be dropped.

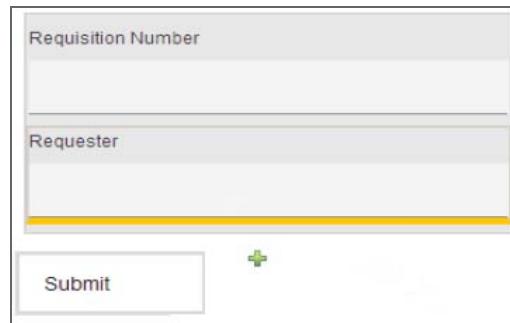
- __ l. Verify the **Requester** input control is placed below **Requisition Number**.

6. Create a panel and populate it with the date details

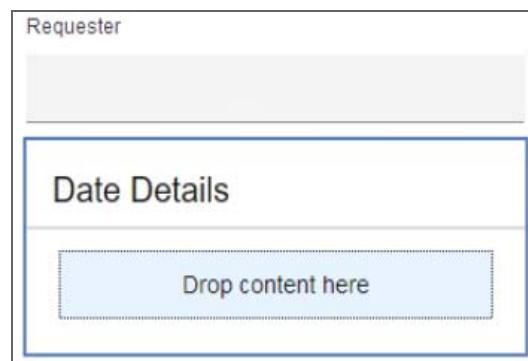
- ___ a. Select **Layout** in the **Building blocks** panel and select **Panel**



- ___ b. While holding down the left mouse button, drag the **Panel** underneath the **Requester** input control. Let go of the mouse button when the bottom border displays a gold highlight.



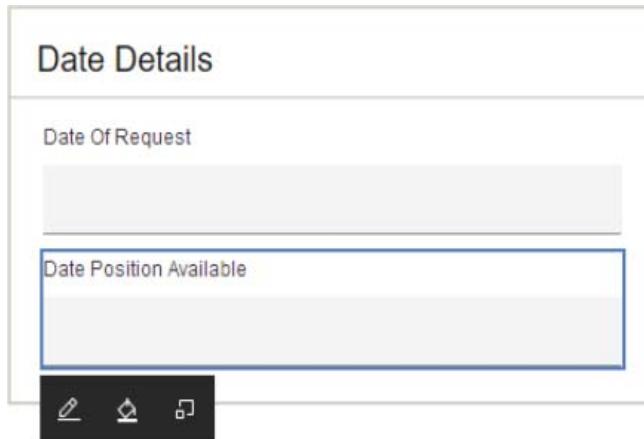
- ___ c. Since the panel is put in place with the name already highlighted, change the name to Date Details and press the **Enter** key.



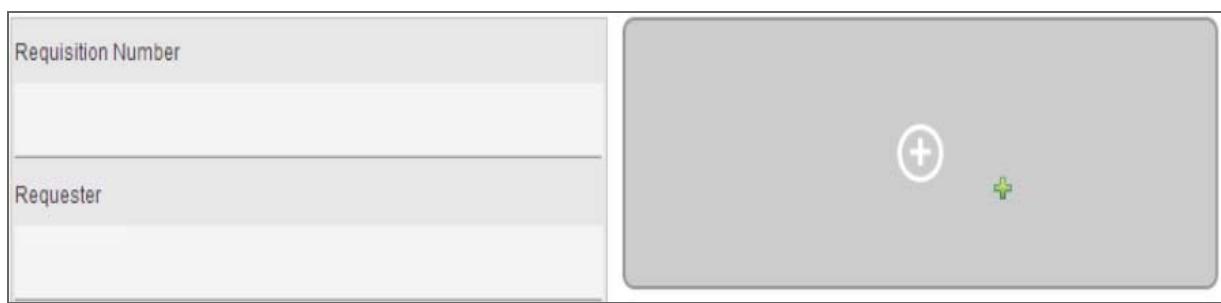
- __ d. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > dateOfRequest**



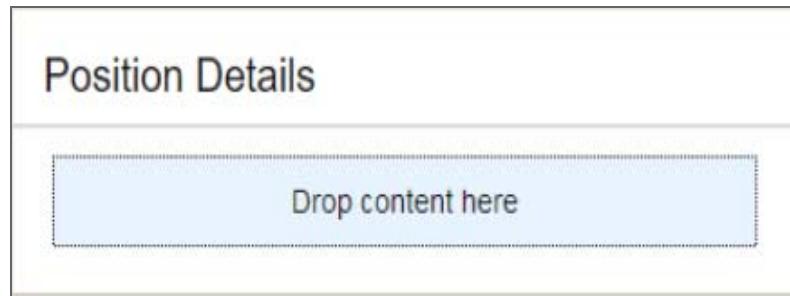
- __ e. While holding down the left mouse button, drag the **dateOfRequest** to the section of the **Date Details** panel labeled **Drop content here**. Let go of the mouse button when the **Drop content here** section is highlighted and the cursor displays a green plus.
- __ f. Repeat the last two steps to drag **datePositionAvailable** underneath the **Date Of Request** input control.



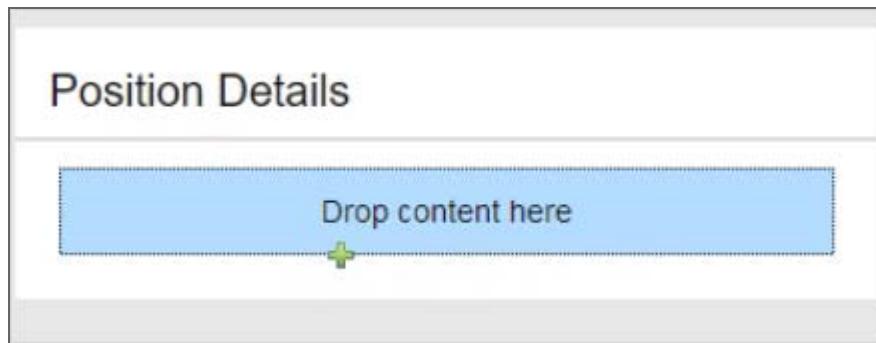
- __ g. Save your work.
7. Create a panel and populate it with the Position business object.
- __ a. Select **Layout** in the **Building blocks** panel and select **Panel**
- __ b. While holding down the left mouse button, drag the **Panel** to the grid on the right. Let go of the mouse button when the section is highlighted and the cursor displays a green plus.



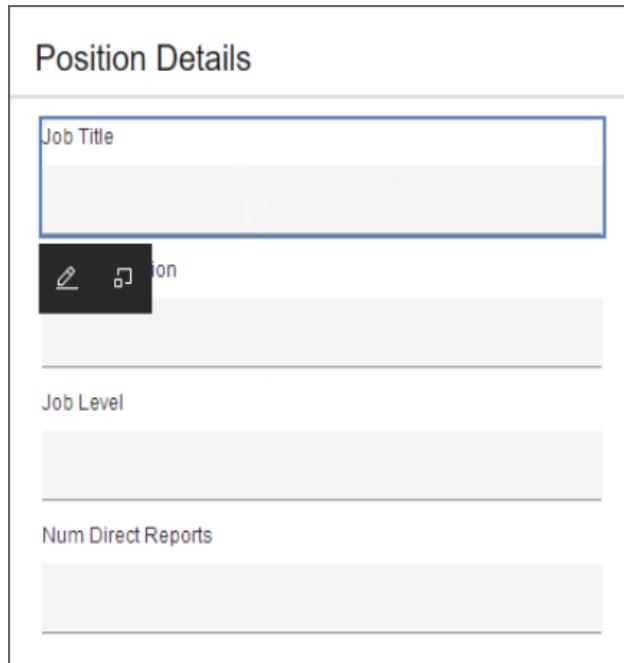
- ___ c. With the name highlighted, label the panel **Position Details**.



- ___ d. Click the empty space below the panel on the canvas to refocus the mouse.
- ___ e. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > position**
- ___ f. While holding down the left mouse button, drag **position** to the section of the **Position Details** panel labeled **Drop content here**. Let go of the mouse button when the **Drop content here** section is highlighted and the cursor displays a green plus.



- ___ g. The section populates with the parameters of the **position** variable.



- ___ 8. Create a panel and populate it with the Recruiting Details business object.
 - ___ a. Select **Layout** in the **Building blocks** panel and select **Panel**
 - ___ b. While holding down the left mouse button, drag a **Panel** underneath the **Date Details** panel. Let go of the mouse button when the bottom border displays a gold highlight.

The screenshot shows a 'Requester' panel with a 'Date Details' sub-panel. The 'Date Details' panel contains two input fields: 'Date Of Request' and 'Date Position Available'. A yellow horizontal bar highlights the bottom edge of the 'Date Details' panel. A green plus sign icon is located at the bottom center of the 'Date Details' panel. At the very bottom of the screen is a 'Submit' button.

- ___ c. With the name highlighted, label the panel **Recruiting Details** and press Enter
- ___ d. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > recruitingDetails**
- ___ e. While holding down the left mouse button, drag `recruitingDetails` to the section of the **Recruiting Details** panel labeled **Drop content here**. Let go of the mouse button when the **Drop content here** section is highlighted and the cursor displays a green plus.
- ___ f. The section populates with the parameters of the `recruitingDetails` variable.

The screenshot shows a 'Recruiting Details' panel. It contains three checkboxes: 'Multiple Employees Needed' (selected), 'Employees Needed' (unchecked), and 'New Position' (unchecked).

- ___ g. Save your work.

- ___ 9. Create a panel underneath Position Details and to the right of Recruiting Details and populate it with the Compensation Details business object.
 - ___ a. Select **Layout** in the **Building blocks** panel and select **Panel**
 - ___ b. While holding down the left mouse button, drag the **Panel** underneath the **Position Details** panel. Let go of the mouse button when the bottom border displays a gold highlight.



- ___ c. With the name highlighted, label the panel **Compensation Details** and press Enter
- ___ d. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > compensationDetails**
- ___ e. While holding down the left mouse button, drag `compensationDetails` to the section of the **Compensation Details** panel labeled **Drop content here**. Let go of the mouse button when the **Drop content here** section is highlighted and the cursor displays a green plus.
- ___ f. Click the empty space below the panel on the canvas to refocus the mouse.

- ___ g. The section populates with the parameters of the compensationDetails variable.

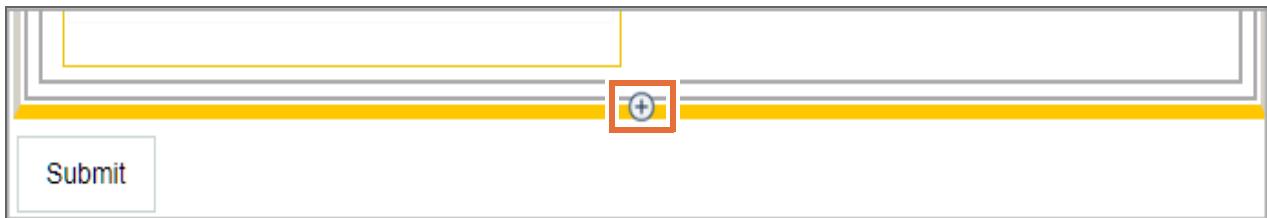
The screenshot shows a user interface element titled "Compensation Details". Inside the panel, there are two sections: "Salary To Offer" and "Bonus Amount", each with a corresponding input field below it. The input fields are represented by large, light-grey rectangular boxes.

- ___ 10. Create a panel and populate it with the Department Details business object.
- ___ a. Select **Layout** in the **Building blocks** panel and select **Panel**
 - ___ b. While holding down the left mouse button, drag the **Panel** underneath the **Recruiting Details** panel. Let go of the mouse button when the bottom border displays a gold highlight.
 - ___ c. With the name highlighted, label the panel **Department Details** and press Enter
 - ___ d. Select **Variable** in the **Building blocks** panel and select **Input > requisitionDetails > departmentDetails**
 - ___ e. While holding down the left mouse button, drag **departmentDetails** to the section of the **Department Details** panel labeled **Drop content here**. Let go of the mouse button when the **Drop content here** section is highlighted and the cursor displays a green plus.
 - ___ f. Click the empty space below the panel on the canvas to refocus the mouse.

- ___ g. The section populates with the parameters of the `departmentDetails` variable and is located underneath the Recruiting Details panel.

The image shows two panels side-by-side. The top panel is titled "Recruiting Details" and contains a checkbox labeled "Multiple Employees Needed" and a text input field labeled "Num Employees Needed". The bottom panel is titled "Department Details" and contains two text input fields, one labeled "Department" and another labeled "Division".

- ___ h. Save your work.
- ___ 11. Place a grid above the Submit button and add the Hiring Manager Comments input control.
- ___ a. Switch to the grid view by clicking the **Grid toggle**. This setting shows the grids on the canvas.
- ___ b. Hover your mouse and click the **(+)** plus sign on the bottom of the outer-most grid to add a grid above the Submit button.



- ___ c. Click the empty space below the panel on the canvas to refocus the mouse.

Your grid should have only a single line between the orange space and the submit button.



- ___ d. Switch to the Content view by clicking the **Content toggle**.
- ___ e. Under the Building Blocks panel, select **Input > requisitionDetails > hiringManagerComments**
- ___ f. While holding down the left mouse button, drag `hiringManagerComments` to the grid. Let go of the mouse button when the grid is highlighted and the cursor displays a green plus.
- ___ g. Click the empty space below the panel on the canvas to refocus the mouse.

Department Details

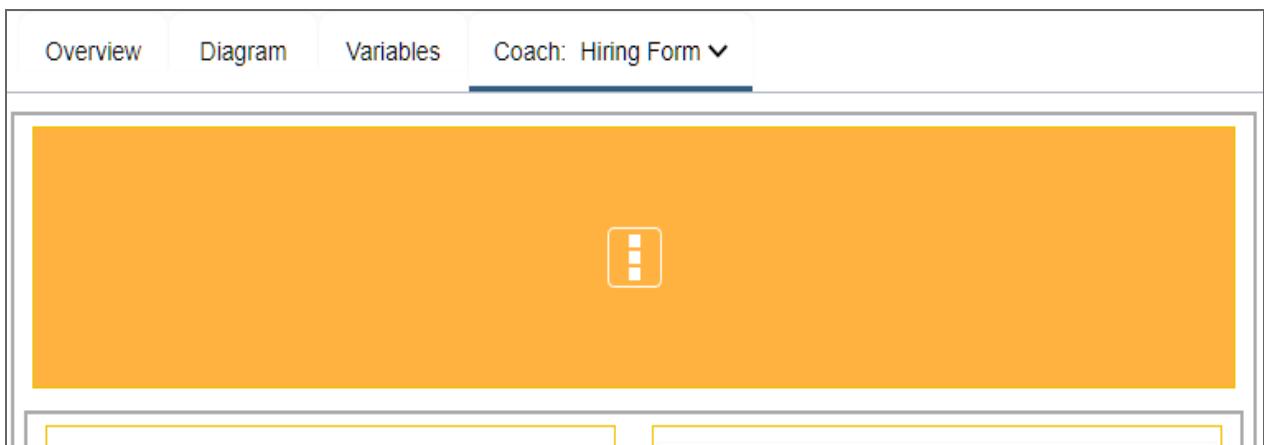
Department	
<hr/>	
Division	
<hr/>	
Hiring Manager Comments	
<hr/>	
<input type="button" value="Submit"/>	

- ___ h. Save your work.
- ___ 12. Place a grid at the top of the coach and group the Requisition Number and Requester variables together, setting them apart from the other data on the screen in a section.
 - ___ a. Switch to the grid view by clicking the **Grid toggle**.

- __ b. Hover your mouse and click the **(+)** plus sign on the top of the outer-most grid to add a grid above the two columns.



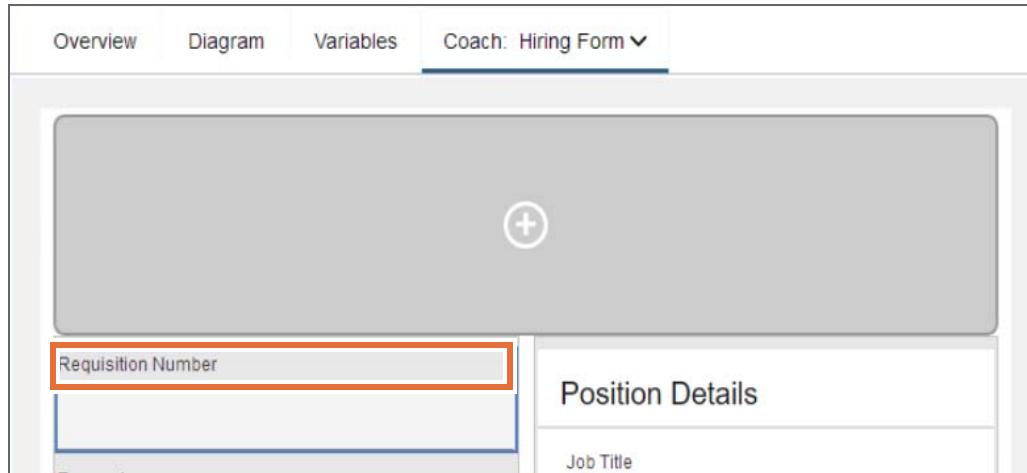
- __ c. Verify your grid is above the grid containing the two columns.



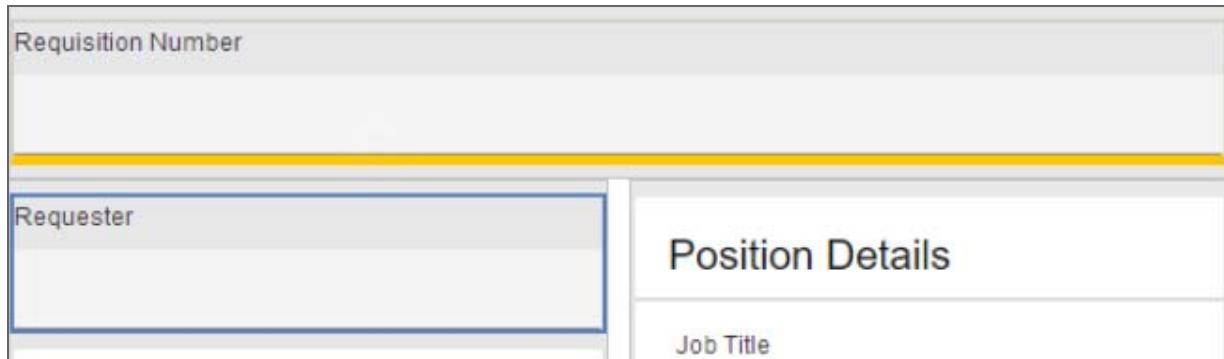
- __ d. Switch to the Content view by clicking the **Content toggle**.



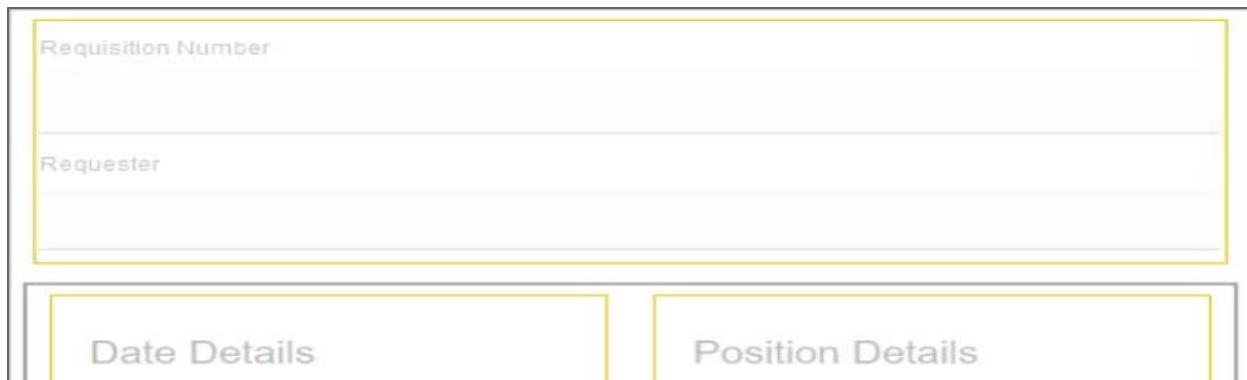
- ___ e. Place the cursor over the **Requisition Number** label, hold down the left mouse button, and drag the **Requisition Number** input control to the new grid at the top. When the grid is highlighted, let go of the mouse button. The mouse turns to the four-arrows icon when the control can be dropped.



- ___ f. Perform the same action for the **Requester** input control, placing it under the Requisition Number control. Use the yellow highlight as a guide to the placement of the control.



- ___ g. Switch to the Grid view.
___ h. Verify that your grid was placed above the two columns.

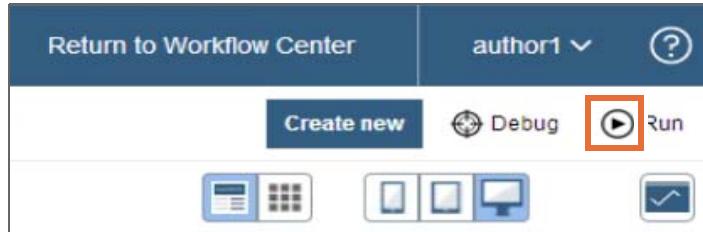


- ___ i. Switch back to the Content view and save your work..

__ j. Verify your coach

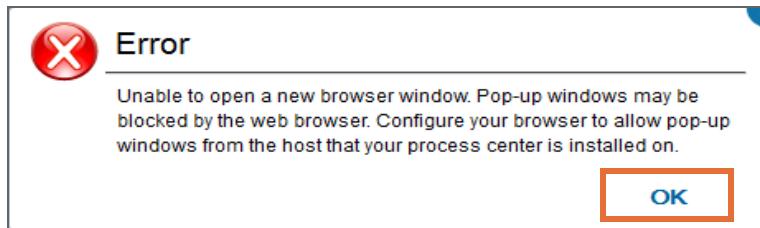
Requisition Number 	
Requester 	
Date Details	
Date Of Request 	Job Title
Date Position Available 	Job Description
Position Details	
Job Level 	Num Direct Reports
Recruiting Details	
<input type="checkbox"/> Multiple Employees Needed	
Num Employees Needed 	
<input type="checkbox"/> New Position	
Compensation Details	
Department 	Salary To Offer
Division 	Bonus Amount
Hiring Manager Comments 	
Submit	

- __ k. Drag any panels out of place to their correct location. Save your work.
- __ 13. Run the coach.
- __ a. Now that a working coach has variables that are bound to the fields, run the coach and view it in a browser. Click **Run** at the upper-right corner of the window.

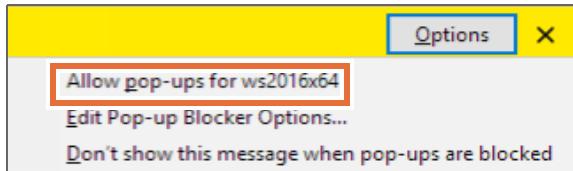


Troubleshooting

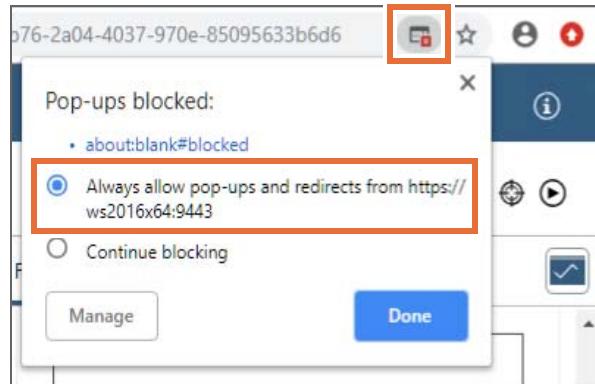
If pop-up windows are disabled in the browser, the browser prevents the site from opening the pop-up window. Click **OK**.



To correct the problem in Firefox, in the yellow bar at the top of the browser, click **Options > Allow pop-ups for ws2016x64**.



To correct the problem in Chrome, click pop-up blocker icon in the address bar and select **Always allow pop-ups and redirects from https://ws2016x64:9443**.



The pop-up window that appears shows the coach designer. Close this window and rerun the coach.

- __ b. Another browser window opens and displays the coach.

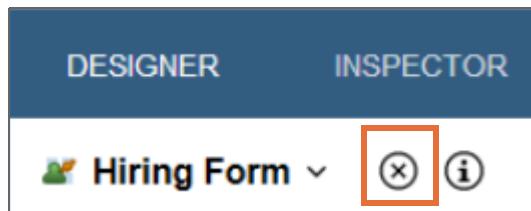
Requisition Number			
Requester			
Date Details Date Of Request <input type="text"/> Date Position Available <input type="text"/>		Position Details Job Title <input type="text"/> Job Description <input type="text"/> Job Level <input type="text"/>	
Recruiting Details <input type="checkbox"/> Multiple Employees Needed Num Employees Needed <input type="text" value="0"/> <input type="checkbox"/> New Position		Num Direct Reports <input type="text" value="0"/>	
Compensation Details Salary To Offer <input type="text" value="0"/> Bonus Amount <input type="text"/>			
Department Details Department <input type="text"/> Division <input type="text"/> Hiring Manager Comments <input type="text"/>			
<input type="button" value="Submit"/>			

The coach looks exactly like the coach that you created in the designer window. Most of the design and functional elements of a coach work the same in the designer as they do after you run the coach. The test run gives you an idea of the final page for the business users, but it usually looks the same as the coach in the designer view. Keep in mind, testing for a customer production situation would entail thoroughly testing the solution on the company supported browser. This coach is not meant to be an attractive, fully functioning front end; you add more functional and visual features later on in this playback. The intent of this playback is to start data flow.

- ___ c. Click inside the **Date Of Request** input control to select a date, insert some data in other fields, and when you are ready, scroll down to click **Submit**.

The screenshot shows a 'Date Details' pop-up window. At the top is a section labeled 'Date Of Request'. Below it is a calendar for October 2020. The days of the week are labeled S (Sunday) through S (Saturday). The dates are arranged in rows: 27, 28, 29, 30, 1, 2, 3; 4, 5, 6, 7, 8, 9, 10; 11, 12, 13, 14, 15, 16, 17; 18, 19, 20, 21, 22, 23, 24; 25, 26, 27, 28, 29, 30, 31. The number 30 is highlighted with a blue border. Navigation arrows are located at the top left and top right of the calendar grid.

- ___ d. The pop-up window displays **Service completed**
 ___ e. Close the pop-up browser window that you ran the coach in.
 ___ f. Close the **Hiring Form** client-side human service window.



Part 3: Reuse and debug the client-side human service

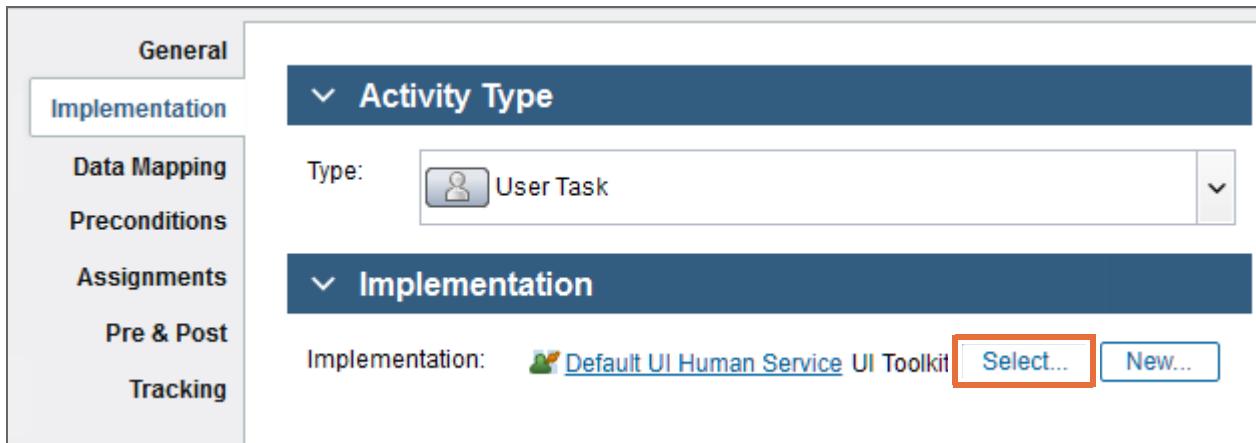
Because the **Approve New Hire Request** activity in the **Hiring Request Process** reviews the same data that is entered in the last human service, you can reuse the Hiring Form human service to add a service to an activity.

In the **Hiring Request Process** modeling effort, many of the activities are placed on the diagram canvas with default human services. It is time to implement the services that were created to the appropriate activities in the process. In this section, you replace the human services in the **Submit Hiring Request** and **Approve New Hire Request** activities in the Hiring Request Process business process with the appropriate human services.

- 1. Implement the **Submit Hiring Request** activity service.
 - a. In the Process Designer, open the **Hiring Request Process** process.
 - b. Click the **Submit Hiring Request** activity.



- c. Click Select in the **Implementation > Implementation** properties section and select the service that implements the user task.



- d. Select **Hiring Form** from the list.



- ___ e. The Hiring Form is now the implementation for the activity.

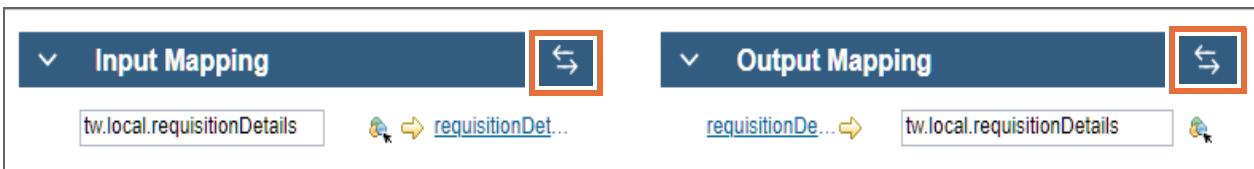


- ___ 2. Map the Submit Hiring Request input and output.
- ___ a. Open the **Data Mapping** menu for the **Submit Hiring Request** activity.

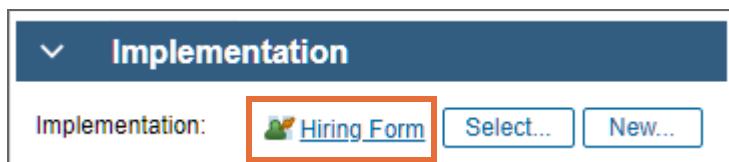


You will see a warning until the input mapping is complete.

- ___ b. Click the **auto-map icon** on both the Input Mapping and Output Mapping sections.



- ___ c. Save your work.
- ___ 3. Implement the **Approve New Hire Request** activity.
- ___ a. Click the **Approve New Hire Request** activity in the **Hiring Request Process** process.
- ___ b. Click **Select** in the **Implementation > Implementation** properties section.
- ___ c. Select the **Hiring Form** client-side human service from the list.
- ___ d. The **Hiring Form** is now the implementation for the activity.



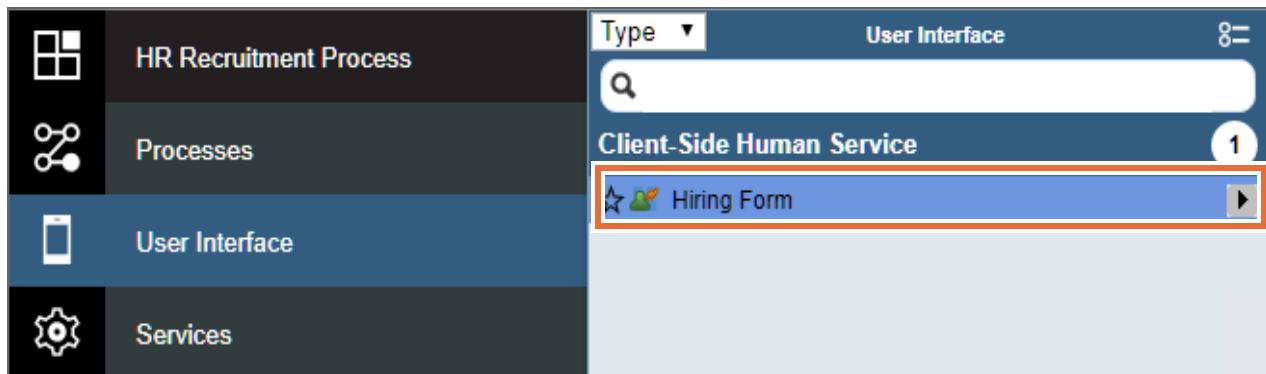
- ___ 4. Map the **Approve New Hire Request** inputs and outputs.
- ___ a. Open the **Data Mapping** menu for the **Approve New Hire Request** activity.
- ___ b. Map `tw.local.requisitionDetails` to both **Input Mapping** and **Output Mapping**.



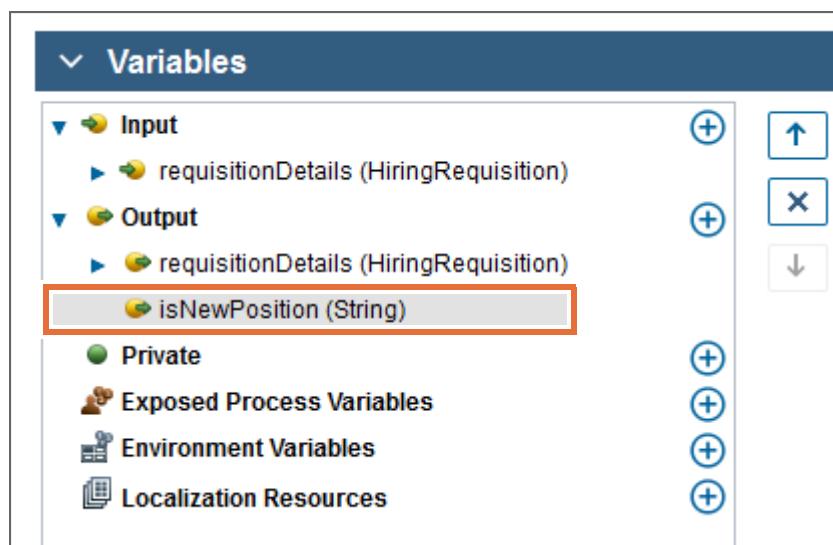
- ___ c. Save your work.
- ___ 5. Add a Script activity to the Hiring Form to evaluate whether the request is for a new position or not.

This script activity determines whether parameter **newPosition** is set to true. If it is, it updates the value of the **isNewPosition** variable to 1, otherwise it sets the value to 0.

- ___ a. In the process library, click **User Interface > Hiring Form**.



- ___ b. Click the **Variables** tab.
- ___ c. Click the (+) plus sign next to Output.
- ___ d. Add the variable: `isNewPosition (String)`



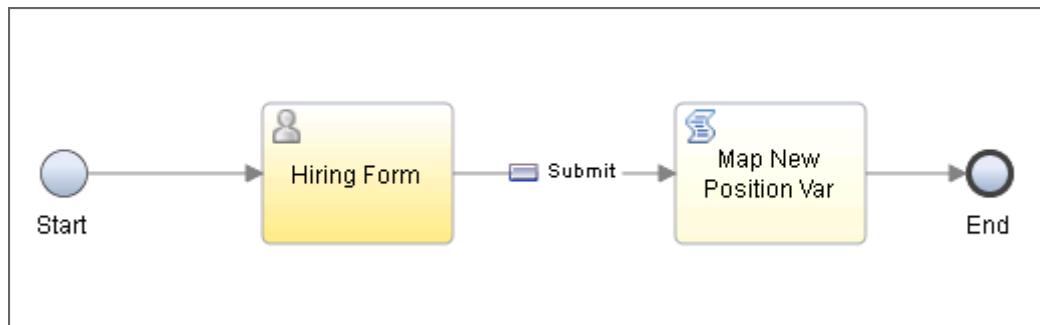
- ___ e. Click the **Diagram** tab.

- ___ f. Drag a **Client-Side script** between the **Hiring Form** coach and the **End** event. Place the script directly on the sequence flow line.



The script is inserted into the sequence flow.

- ___ g. Name the new client-side-script: Map New Position Var



- ___ h. Click the **Map New Position Var** script activity that you placed on the canvas.
 ___ i. Open the **Script** section in the properties below.

Script

This editor uses standard **JavaScript syntax**. Press Ctrl-space while typing to receive assistance with the script syntax and contents.

```

1 // This JavaScript will run within the browser and must use the client-side JavaScript syntax
2 // - To instantiate a complex object:
3 //   tw.local.customer = {};
4 //   tw.local.customer.name = "Jane";
5
  
```

The client-side script box has some example code commented out. You can erase the code, or you can add the code below the commented code. This code provides examples of how to instantiate different types of variables.

- ___ j. Copy and paste the following code in the **Script** field. You can copy the code from the file `Script1.txt` at the location: C:\labfiles\Support files\Ex5

```
if (tw.local.requisitionDetails.recruitingDetails newPosition == true)
    tw.local.isNewPosition = "1";
else
    tw.local.isNewPosition = "0";
```

Script

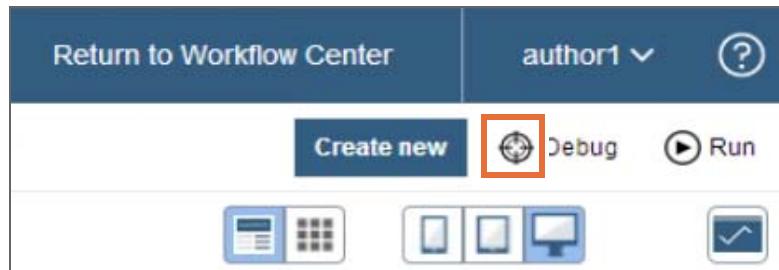
This editor uses standard JavaScript syntax. Press Ctrl-space while typing to receive assistance with the script syntax and contents.

```
1// This JavaScript will run within the browser and must use the client-side JavaScript syntax supported by the browser. For example,
2// - To instantiate a complex object:                                - To instantiate a list:
3//     tw.local.customer = {};                                     tw.local.addresses = [];
4//     tw.local.customer.name = "Jane";                            tw.local.addresses[0] = {};
5if (tw.local.requisitionDetails.recruitingDetails newPosition == true)
6    tw.local.isNewPosition = "1";
7else
8    tw.local.isNewPosition = "0";
9    tw.local.requisitionDetails.departmentDetails.department = tw.local.department.name;
```

- ___ k. Save your work.

- ___ 6. Debug the **Hiring Form** client-side human service and verify that data is flowing.

- ___ a. Click the **Debug** icon.



- ___ b. A new browser window is displayed, but this time it is in debug mode.

This window is hosting your debug session, and must remain open while you are debugging.

The running client-side human service is paused at the first activity. To proceed with the debugging of the client-side human service, use the actions on the sidebar of the Inspector.

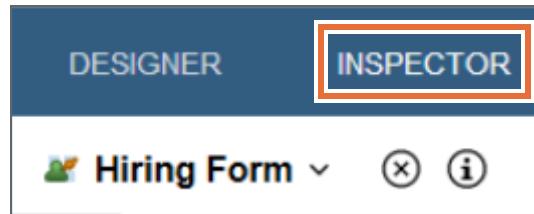
If this window is closed, use the 'Show Playback Window' action to reopen it.

- ___ c. Minimize this new browser window to view the client-side human service designer window.

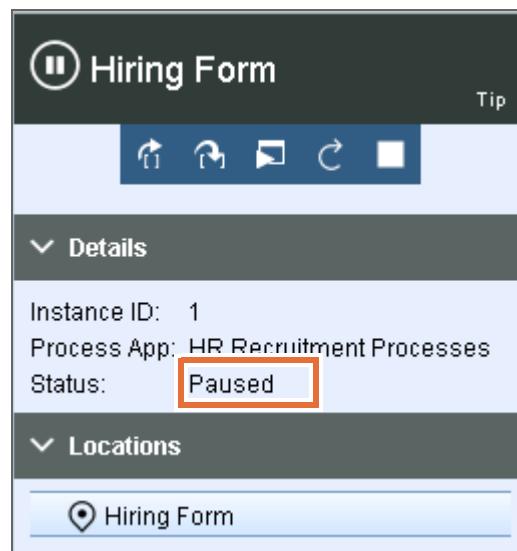


Information

The coach designer switches to the **Inspector** tab.



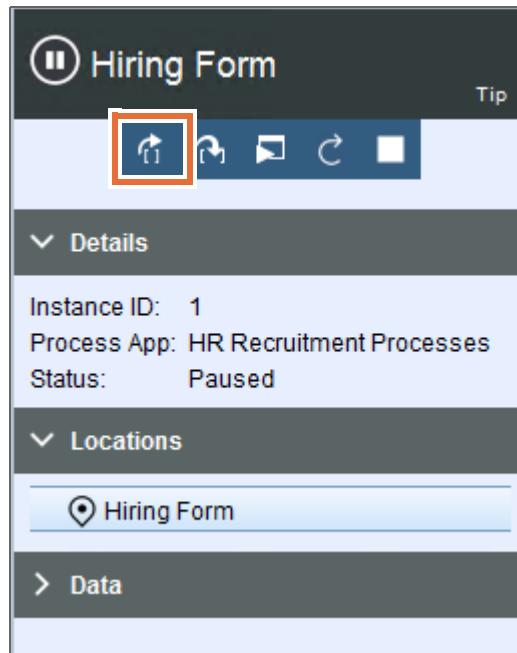
- ___ d. The inspector section is displayed on the right side. The status shows that the execution is **Paused**.



- ___ e. If not already selected, click the **Diagram** tab. The inspector shows a token that is waiting to execute this step in the process.



- ___ f. Click the **Step over** icon for the Hiring Form coach.



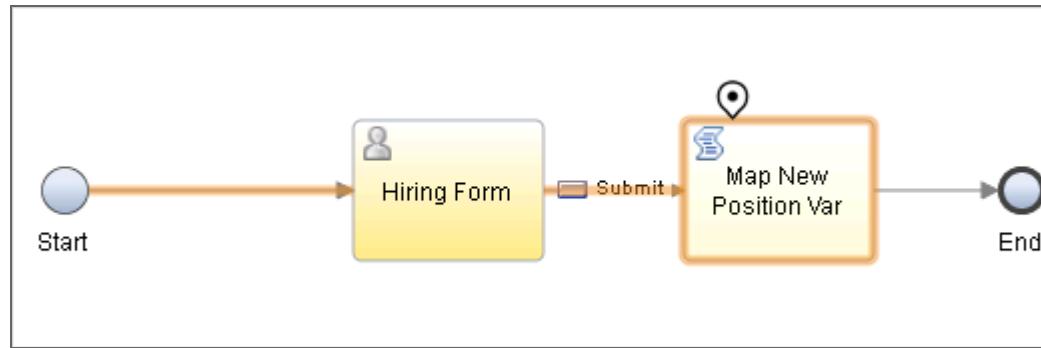
- ___ g. The second browser window renders the coach. Enter 12345 as the Requisition Number. Ensure that the **NewPosition** checkbox is not selected.

Requisition Number	
12345	
Requester	
Date Details	
Date Of Request	
Date Position Available	
Position Details	
Job Title	
Job Description	
Job Level	
Recruiting Details	
<input type="checkbox"/> Multiple Employees Needed	
Num Employees Needed	0
0	
<input type="checkbox"/> New Position	
Compensation Details	
Num Direct Reports	0

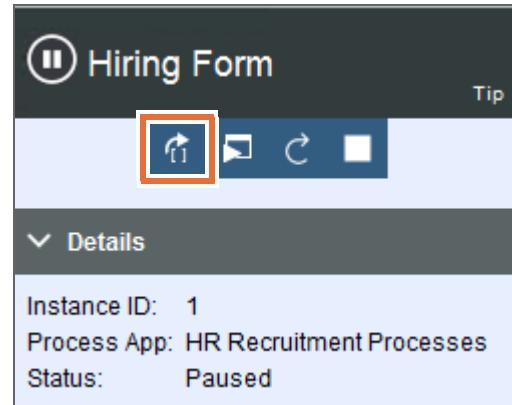
- __ h. Click **Submit**.

The screenshot shows a form titled "Department Details". It contains fields for "Department" and "Division", both of which are currently empty. To the right of these fields are two input fields: "Salary To Offer" (containing "0") and "Bonus Amount" (empty). Below these is a large text area labeled "Hiring Manager Comments". At the bottom left is a button labeled "Submit", which is highlighted with a red rectangular border.

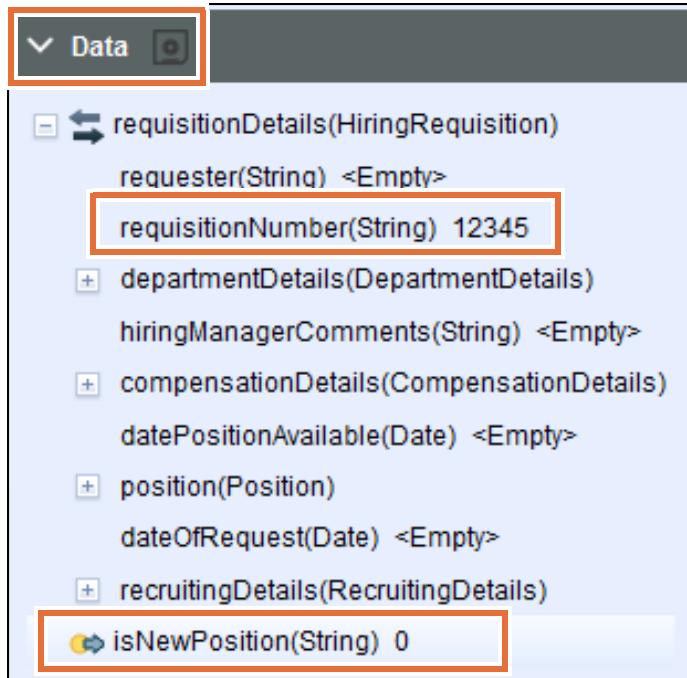
- __ i. Minimize the coach browser window and return to the designer browser window.
The Hiring Form coach is complete, and the token moves to **Map New Position Var**.



- __ j. The status is **Paused**, waiting for the next step in the service. Click **Step over** to run the Map New Position Var server script.



- ___ k. The status changes to **Finished** and the token disappears because it reached the end event. On the right, expand the **Data** section and click the (+) plus sign next to **requisitionDetails** to see the updated value for the `requisitionDetails.requisitionNumber` variable. Verify that the updated value for the `requisitionDetails.requisitionNumber` variable is 12345. The `tw.local.isNewPosition` variable is set to the String "0".



- ___ l. Run the service again, and this time select the **New Position** checkbox on the coach. The `isNewPosition` value should be set to 1.

Recruiting Details

Multiple Employees Needed

Num Employees Needed

0

New Position

- ___ m. Return to the Debug Hiring Form and verify that the service is completed. Close this browser window.
- ___ n. The Status is **Finished**, and you are currently on the **INSPECTOR** tab. Return to the **DESIGNER** tab. The activity and flow highlights disappear in the Designer mode.
- ___ 7. Map the new variable in the process.
- ___ a. Open the **Hiring Request Process** process.
- ___ b. Click the **Submit Hiring Request** activity.

- ___ c. In the **Data Mapping** properties section, map the `isNewPosition` output variable to `tw.local.isNewPosition`.



- ___ d. Click the **Approve New Hire Request** activity.
___ e. In the **Data Mapping** properties section, map the `isNewPosition` output variable to `tw.local.isNewPosition`.
___ f. Save your work.

You have completed this exercise.

Playback 1: Business data and services is complete. In this exercise, by using the core requirements, you created all of the necessary assets to support a coach in the Hiring Request Process. You modeled the coach using grids and common controls. You used complex business objects (variable types) to organize your data, and pass data into and out of a linked process. You built a human service and defined guided user interactions with a coach. You also implemented a service for an activity, and mapped variables between a linked service and an activity.

End of exercise

Exercise 6. Playback 1: User interface design and implementation

Estimated time

01:30

Overview

In this exercise, you group controls into tabs on a coach and change the appearance of the coach by applying a theme.

Objectives

After completing this exercise, you should be able to:

- Create tabs on a coach
- Change the appearance of a coach by applying a custom theme
- Change the coach layout for a mobile format
- Configure controls to respond to different screen sizes
- Debug the coach by using a responsive sensor

Introduction

In Playback 1, the enhancements that the business wants for process application user interfaces are implemented. Often, the initial development efforts are marred with requests to begin with full fidelity user interfaces. It is not uncommon to have this request because it is driven from a desire to impress executive level stakeholders with prototypical user interfaces that can be manipulated. It is better to reserve this type of development for this playback stage. You ensure that the data model is in place, the functions of the user interface are approved, and all the remaining process application interactions and integrations are complete. Now the enhancements can be done without fear that rework would be needed later because of a change in requirements to the items in the prior playbacks. Changes can happen, but the likelihood is that those functional requirement changes are handled in Optimization and not within the development cycle because consensus is reached to move to this stage of development.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Group controls into tabs on a coach](#)

[Part 2: Change the appearance of the coach by applying a custom theme](#)

[Part 3: Customize controls](#)

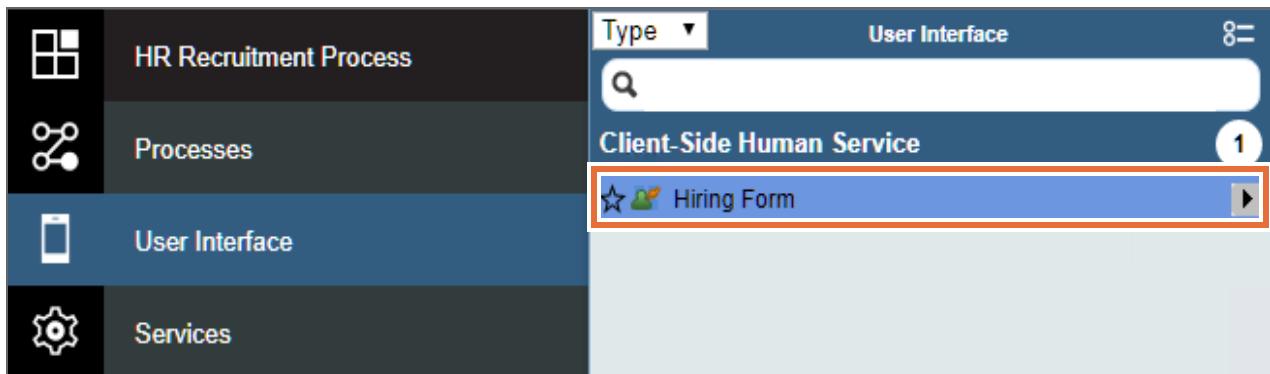
[Part 4: Modify the mobile layout](#)

Exercise instructions

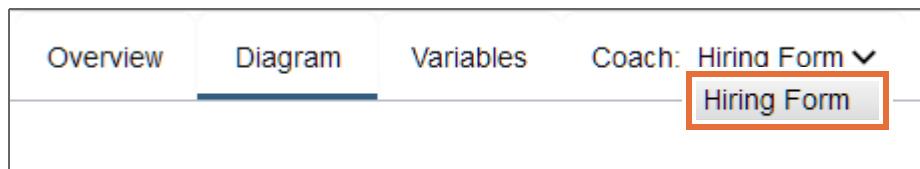
Part 1: Group controls into tabs on a coach

In one of the previous playbacks, you provided a functional look at the coaches in the Hiring Request Process. Enhance the coaches with more features and functions. Start with creating a polished look to the coach by grouping similar data into tabs.

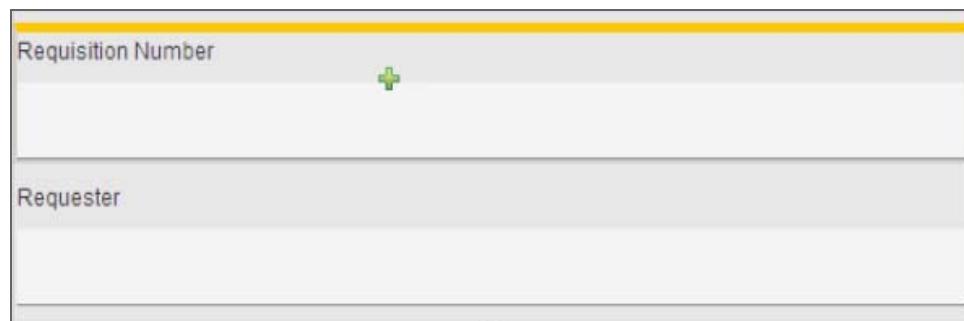
- ___ 1. Create a panel and populate it with two parameters from the Requisition Details business object.
 - ___ a. Open the **HR Recruitment Process** process application.
 - ___ b. In the Process Designer library, open the **Hiring Form**.



- ___ c. Click the down arrow next to **Coach: Hiring Form** and select **Hiring Form**.



- ___ d. Select **Layout** in the **Building blocks** panel and select **Panel**
- ___ e. While holding down the left mouse button, drag the **Panel** above the **Requisition Number** input control. Let go of the mouse button when the top border displays a gold highlight.



- __ f. With the name highlighted, label the panel **Requisition Details** and press Enter

The screenshot shows a window titled "Requisition Details". Inside the window, there is a blue rectangular area with a dotted border containing the text "Drop content here". Below this area, the text "Requisition Number" is visible.

- __ g. Place the cursor over the **Requisition Number** label, hold down the left mouse button, and drag the **Requisition Number** input control to the section of the new panel labeled **Drop content here**. Let go of the mouse button when the panel is highlighted.

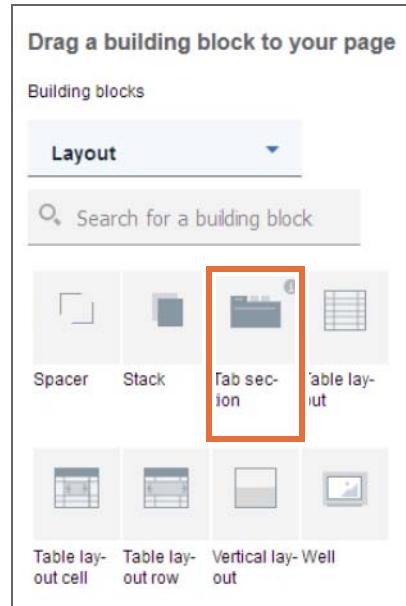
- __ h. The result is the Requisition Number is in the new Panel.

The screenshot shows the same window titled "Requisition Details". The "Requisition Number" label is now positioned above a large, empty text input field. The "Requester" label is still present below it.

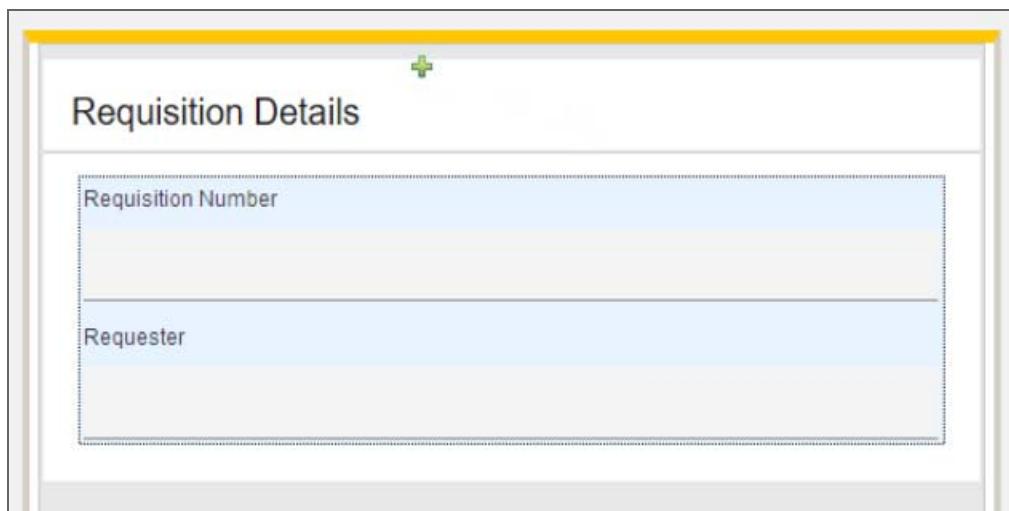
- __ i. Perform the same steps to place the **Requester** input control under the Requisition Number of the same panel. Use the yellow highlight as a guide to the placement of the control.

The screenshot shows the same window titled "Requisition Details". The "Requisition Number" label is above the first text input field, and the "Requester" label is above the second text input field. Both fields are large and empty.

- ___ j. Save your work.
- ___ 2. Create a tab section to hold the grouped sections on the coach.
 - ___ a. While holding down the left mouse button, drag a **Tab section** above the **Requisition Details** panel.



- ___ b. Let go of the mouse button when the top border displays a gold highlight.



- ___ 3. Build the Requisition Details tab.
- ___ a. Select the **Date Details** label for the panel that contains the **Date of Request** and the **Date Position Available** input controls and drag it into the editable area of the **Requisition Details** panel under **Requester**.

The screenshot shows a user interface design tool with a central workspace. At the top, there's a toolbar with a green plus icon and a placeholder box labeled "Drop content here". Below this is a header bar with the title "Requisition Details". The main content area is divided into several panels:

- A large blue panel on the left labeled "Requisition Number" containing two empty text input fields.
- A blue panel below it labeled "Requester" containing one empty text input field.
- A panel titled "Date Details" with a red border, containing two input fields: "Date Of Request" and "Date Position Available".
- A panel titled "Position Details" containing three input fields: "Job Title", "Job Description", and "Job Level".

The "Date Details" panel is highlighted with a red border, indicating it is selected or being manipulated. A yellow horizontal bar is visible at the bottom of the "Requester" panel.

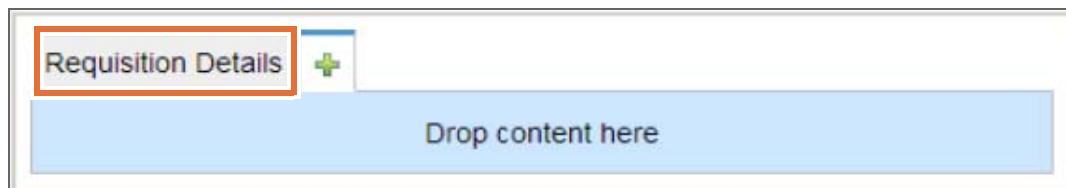
- ___ b. The **Requisition Details** panel should look like this screen capture after you add the **Date Details** panel.

The screenshot shows a user interface with a main panel titled "Requisition Details". Inside this panel, there are two input fields: "Requisition Number" and "Requester". Below the "Requisition Details" panel, there is a separate panel titled "Date Details", which also contains two input fields: "Date Of Request" and "Date Position Available". The "Date Details" panel is enclosed in a blue border, indicating it is a nested or modal component.

- ___ c. Drag the **Hiring Manager Comments** control underneath the **Date Details** panel in the **Requisition Details** panel.

The screenshot shows a user interface with a main title "Requisition Details". Below it are two stacked panels. The top panel is titled "Requester" and contains two input fields: "Requisition Number" and "Requester". The bottom panel is titled "Date Details" and contains two input fields: "Date Of Request" and "Date Position Available". The entire "Date Details" panel is highlighted with a yellow border.

- ___ d. Drag the **Requisition Details** panel into the tabs area that is labeled **Drop content here**. The label on the tab changes to match the label that is applied to the panel, and a second tab is added.



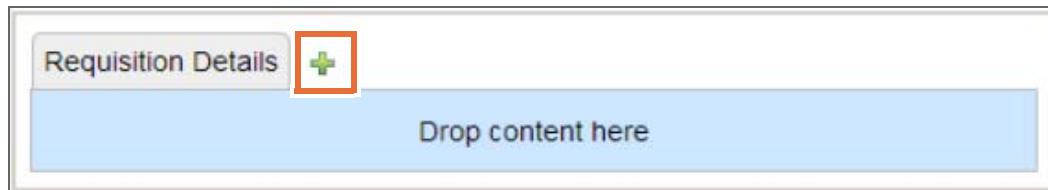
- ___ e. Click the tab to view the **Requisition Details**.

- ___ f. Your Requisition Details tab should look like the screen capture.

The screenshot shows a user interface for a 'Requisition Details' tab. At the top left is the tab name 'Requisition Details' and a small green plus sign icon. Below the tab is a section titled 'Requisition Details'. Inside this section are two input fields: 'Requisition Number' and 'Requester'. Below this is another section titled 'Date Details' containing two input fields: 'Date Of Request' and 'Date Position Available'. At the bottom is a section titled 'Hiring Manager Comments' with a single input field.

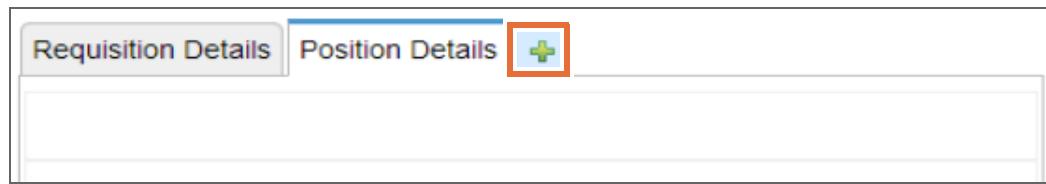
- ___ 4. Build the remaining four tabs.

- ___ a. Click the (+) plus sign on the tab next to the **Requisition Details** tab.



- ___ b. Drag the **Position Details** section onto the section titled "Drop content here". This one is easier because all the controls are already in a single section.

- ___ c. Create a tab for the **Recruiting Details**. This time, drag the **Recruiting Details** panel over the plus sign to the right of the **Position Details** tab.

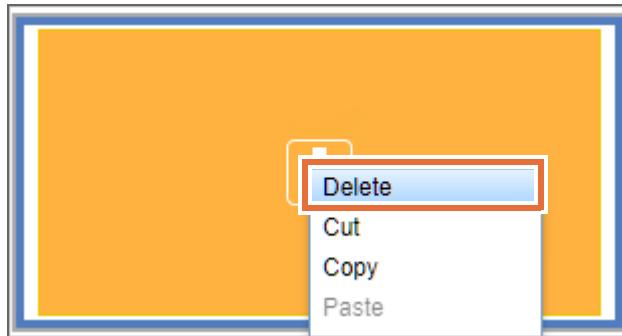


- ___ d. Release the mouse button, and the section becomes the **Recruiting Details** tab next to the **Position Details** tab.

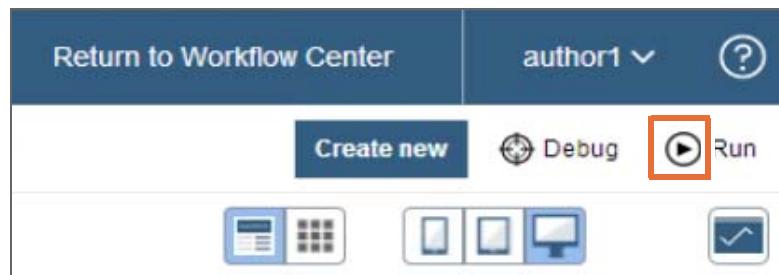
- ___ e. Repeat the last two steps for the **Compensation Details** and **Department Details**. Drag each of the respective sections onto their tabs.

- ___ 5. Delete the empty grids.
___ a. Return to the **Grid** view.

- ___ b. The empty grids are displayed. Right-click each of the four empty grids and click **Delete**.



- ___ c. Return to **Content** view.
___ d. Save your work.
___ 6. View the coach.
___ a. Click **Run** at the top of the coach.



- ___ b. The coach now has multiple tabs. Click the tabs to see what is contained on each tab.

Requisition Details	Position Details	Recruiting Details	Compensation Details	Department Details
<h3>Requisition Details</h3> <p>Requisition Number</p> <input type="text"/> <p>Requester</p> <input type="text"/>				
<h3>Date Details</h3> <p>Date Of Request</p> <input type="text"/> <p>Date Position Available</p> <input type="text"/>				
<p>Hiring Manager Comments</p> <input type="text"/>				
<input type="button" value="Submit"/>				

- ___ c. Close the browser that displayed the form.

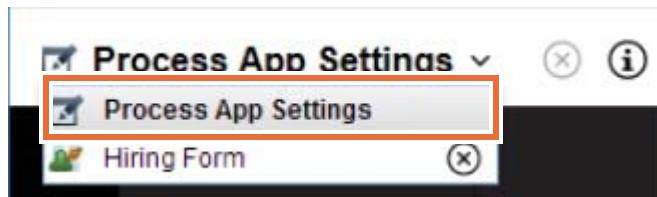
Part 2: Change the appearance of the coach by applying a custom theme

Creating a theme consists of assigning values to a set of theme variables. By default, a new process application uses the Carbon Theme because it contains all the variables that the controls in the UI Toolkit use. To create a custom theme and use any of these settings in your process application, you use the Classic Theme as your starting point. You can then extend your theme by adding custom variables and modifying the variables.

- ___ 1. View the solution using the Spark Theme.

Three themes are provided out-of-the-box. You can use themes to quickly change the look and feel of a user interface.

- ___ a. View the Process App Settings by selecting the page from the menu.



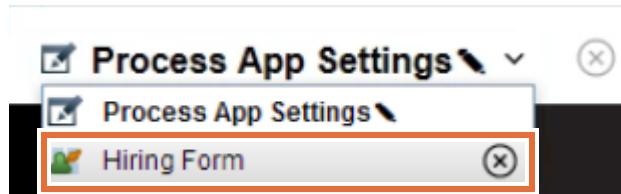
- ___ b. Under the Coach Designer Settings section, click Select next to the Theme.

Coach Designer Settings			
Theme:	Carbon	System Data	Select... (highlighted)
Heritage Coach XSL:	CoachDesigner.xsl	System Data	Select... New... Clear
Heritage Coach CSS:	coach_designer.css	System Data	Select... New... Clear

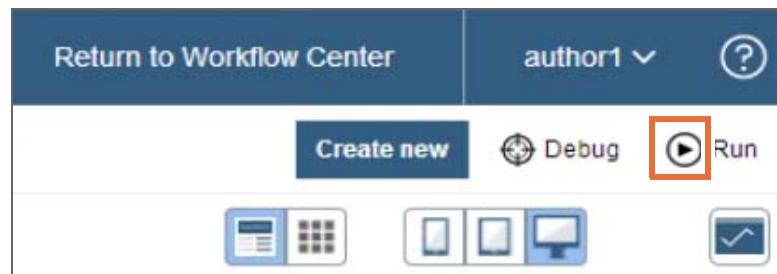
- ___ c. You are presented with the list of out-of-the-box themes. Select the SPARK UI Theme.



- ___ d. Return to the Hiring Form.



- ___ e. Click the **Run** icon at the upper-right side to display the coach with the new theme. Initially when you view the coach, the server must generate the CSS and render the new coach on the screen, so a delay might occur before you see the effect.



- ___ f. The coach is displayed using the Spark UI Theme.

The screenshot shows a requisition form with the following fields:

- Requisition Details** tab is selected.
- Position Details**, **Recruiting Details**, and **Compensation Details** tabs are visible at the top.
- Requisition Number**: An empty input field.
- Requester**: An empty input field.
- Date Details**: A section containing two input fields: **Date Of Request** and **Date Position Available**.
- Hiring Manager Comments**: An empty input field.
- Submit**: A button at the bottom left.

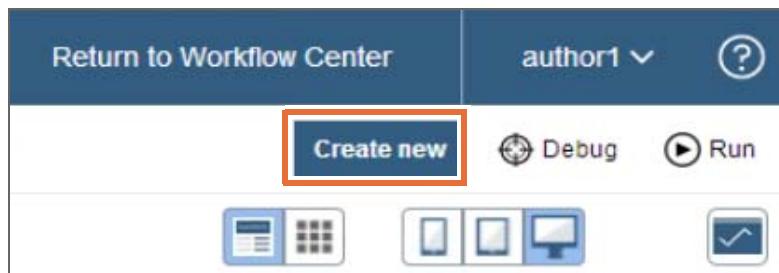
- ___ g. Close the browser that displayed the coach.

- ___ h. Optional: Repeat the process to display the Classic theme.

When you create a custom theme in the next step, you base it off of this theme.

- ___ 2. Create a custom theme for the HR Recruitment Processes process application.

- ___ a. Click **Create New** on the upper-right corner of the Process Designer.



- __ b. Click **Theme**.



- __ c. Enter the name: Hiring Request Theme

Change the option to **Copy an existing theme: Classic (System Data)** and click **FINISH**.

New Theme

A theme sets the visual style of the coaches within a process application or toolkit. A theme can also apply to a portal in Process Portal.

Name:

Copy an existing theme
Classic (System Data)

Import a theme file
Browse... No file selected.

FINISH **CANCEL**

The screenshot shows a 'New Theme' dialog box. At the top is the title 'New Theme'. Below it is a descriptive text: 'A theme sets the visual style of the coaches within a process application or toolkit. A theme can also apply to a portal in Process Portal.' To the right of this text is a graphic of four overlapping circles in red, green, blue, and yellow. The main area contains a 'Name:' label followed by an input field containing 'Hiring Request Theme', which is highlighted with a red border. Below this is a radio button group for 'Copy an existing theme', with 'Classic (System Data)' selected. There is also an 'Import a theme file' section with a 'Browse...' button and a message 'No file selected.'. At the bottom of the dialog are two buttons: 'FINISH' and 'CANCEL', with 'FINISH' also highlighted with a red border.

- __ d. The editor opens the new theme and defaults to the Design tab.

The screenshot shows the BPMN Studio UI Designer interface. The top navigation bar has tabs for Overview, Design (which is selected), and Source. The main area is divided into two sections. On the left, under 'Base Settings', there is a list of CSS-like rules with their corresponding color hex codes. On the right, there are examples of UI components: 'Controls' (Input, Input Small, Input Large), 'Buttons' (Default, Primary, Info, Success, Warning, Danger, Large, Small, Extra-Small), and date/time pickers. A 'Radio Button Group' and a 'Check Box Group' are also shown.

- __ 3. Change the primary color of the theme

Management wants to change the color scheme to match the corporate color scheme. Change the settings of the default Classic Theme and apply the new theme to the process.



Hint

You can use the browser **Find in This Page** feature to locate text on the page.

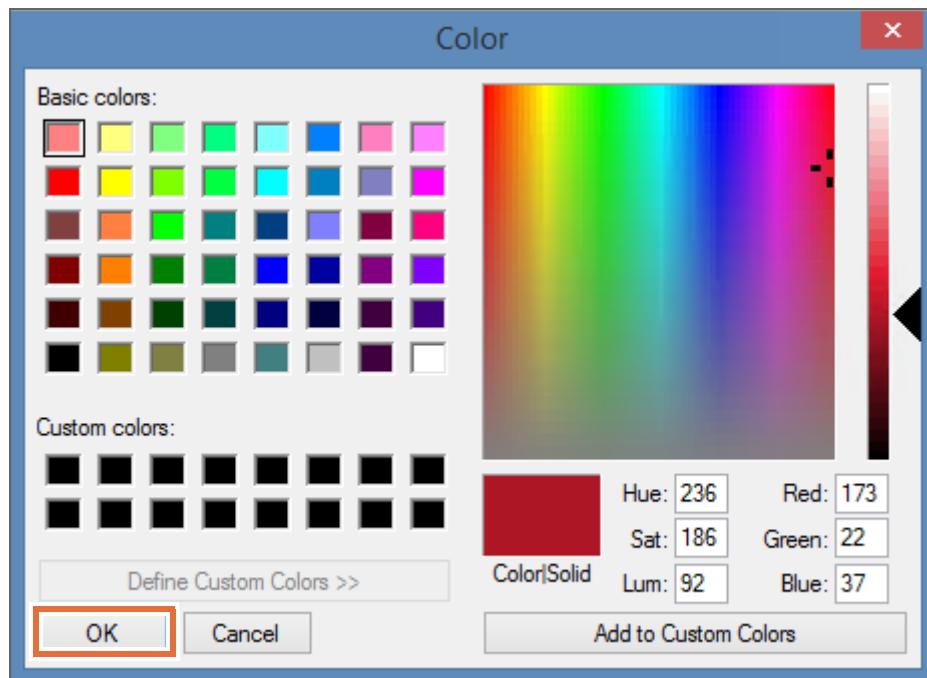
- a. In the Design page of the theme editor, change the variable `@bpm-color-primary` in the **Base Settings** section from `#325C80` to: `#bcc2c5`

Variable	Value
@bpm-neutral-darkest	#0d1111
@bpm-neutral-darker	#2d3737
@bpm-neutral-dark	#3c4646
@bpm-neutral	#586464
@bpm-neutral-light	#6d7777
@bpm-neutral-lighter	#c8d2d2
@bpm-neutral-lightest	#dfe9e9
@bpm-color-primary-light	lighten(desaturate(spin(@bpn
@bpm-color-primary	#bcc2c5
@bpm-color-primary-dark	darker(desaturate(spin(@bpn



Note

Many variables also have a swatch that you can click, then choose a value directly from a picker, and click **OK**.



- ___ b. When you “blur” or click away from the color input field, you immediately see the color change for the affected controls on the right. Click any control on the right to see the CSS that affects the appearance of that object on the page.

@bpm-color-primary	#bcc2c5
@bpm-color-primary-dark	darker(desaturate(spin(@bpm-color-primary, 10%, 10%, 10%)))
@bpm-color-primary-darker	darker(desaturate(spin(@bpm-color-primary-dark, 10%, 10%, 10%)))
@bpm-color-info-light	lighten(@bpm-color-info, 10%)
@bpm-color-info	#c0e6ff
@bpm-color-info-dark	darker(spin(@bpm-color-info, -10%, 10%, 10%))
@bpm-color-info-darker	darker(desaturate(spin(@bpm-color-info-dark, 10%, 10%, 10%)))
@bpm-color-success-light	lighten(@bpm-color-success, 10%)
@bpm-color-success	#c8f08f

Radio Button Group **Date Time Picker**

Option 1
 Option 2

Check Box Group

Option 1
 Option 2

Tuesday, May 9, 2017 ×

Font family: @bpm-font-family-base
 Font size: @bpm-font-size-base
 Font color: @bpm-fill-normal
 Background color: @bpm-color-primary
 Border color: @bpm-neutral

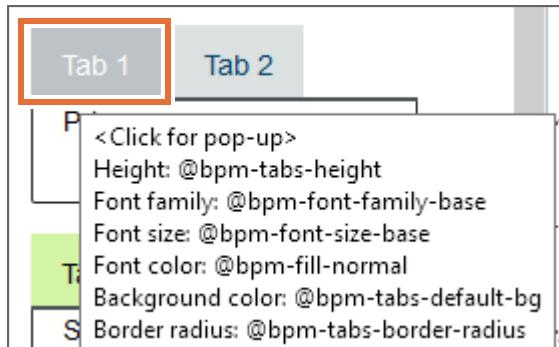
- ___ 4. Change the font color of the tabs.

- ___ a. Click the **Containers** tab on the right.

The screenshot shows a user interface design application. At the top, there are four tabs: 'Controls', 'Tables', 'Containers' (which is highlighted with a red box), and 'Pop-ups and Dialogs'. Below the tabs, there is a section titled 'Tabs' with four examples:

- Default:** Shows two tabs: 'Tab 1' (blue) and 'Tab 2' (light blue). The content area is white.
- Primary:** Shows two tabs: 'Tab 1' (red) and 'Tab 2' (light red). The content area is white.
- Info:** Shows two tabs: 'Tab 1' (green) and 'Tab 2' (light green). The content area is white.
- Success:** Shows two tabs: 'Tab 1' (yellow) and 'Tab 2' (light yellow). The content area is white.
- Warning:** Shows two tabs: 'Tab 1' (orange) and 'Tab 2' (light orange). The content area is white.
- Danger:** Shows two tabs: 'Tab 1' (purple) and 'Tab 2' (light purple). The content area is white.

- ___ b. Hover over the Primary Tabs control in the first row in the right column. It shows all the variables that are associated with this control. For example, to change the font color of the tabs, you need to change the value of the `@bpm-tabs-primary-color` variable.



- ___ c. To change the font color, collapse the **Base Settings** section and expand the **Tabs** section.

<code>@bpm-nav-link-padding</code>	10px 15px
<code>@bpm-nav-tabs-border-color</code>	<code>@bpm-neutral-dark</code>
<code>@bpm-tabs-default-color</code>	<code>@bpm-text-color-negative</code>
<code>@bpm-tabs-default-bg</code>	<code>@bpm-color-primary</code>

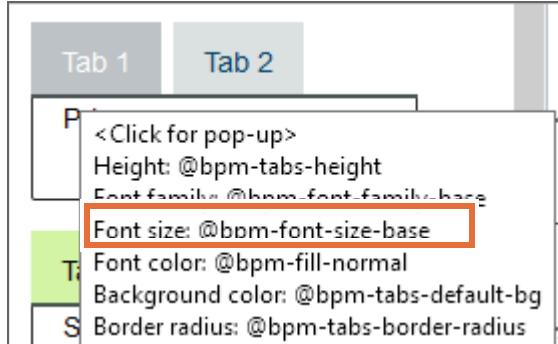
- ___ d. Look for the variable `@bpm-tabs-primary-color` and change the color to: #008000

<code>@bpm-tabs-default-accent</code>	desaturate(lighten(@bpm-	light blue
<code>@bpm-tabs-default-content-ba</code>	<code>@bpm-fill-normal</code>	white
<code>@bpm-tabs-primary-color</code>	#008000	green
<code>@bpm-tabs-primary-simple-color</code>	<code>@bpm-text-color</code>	black
<code>@bpm-tabs-primary-bg</code>	<code>@bpm-color-primary</code>	grey
<code>@bpm-tabs-primary-border</code>	<code>darker(@bpm-tabs-prima</code>	grey

You can choose to use any color of your choice. If you run into any problem, return to the step and follow that exactly, and then you can customize the color.

___ 5. Change the font size.

To change the font size, you need to change the value of the `@bpm-font-size-base` variable.



- ___ a. Collapse the Tabs section, and expand the **Base Settings** section.
 ___ b. Look for the variable `@bpm-font-size-base` and change the value to: 16px

		Overview	Design	Source
@bpm-fill-normal	#fff			
@bpm-border-normal	lighten(@bpm-neutral, 2%)			
@bpm-link-color	@bpm-color-primary			
@bpm-text-color	#000203			
@bpm-font-family-sans-serif	"Helvetica Neue", Helvetica, A			Lo&rem Ipsum
@bpm-font-size-base	16px			
@bpm-line-height-base	1.5			

___ 6. Change the background color of the input controls.

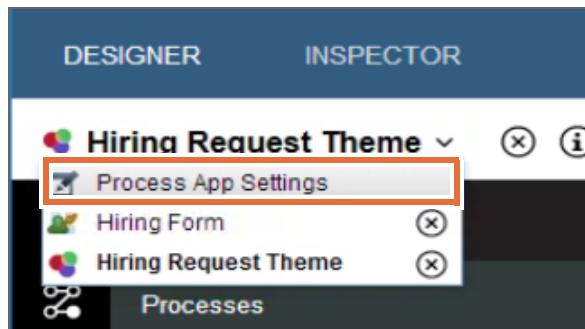
- ___ a. Click the **Controls** tab on the right to view the input fields.

The screenshot shows the 'Controls' tab selected. It displays three examples of input fields: 'Input' (a standard text input), 'Input Small' (a smaller text input), and 'Input Large' (a large text input).

- ___ b. Expand the **Forms** section on the left.
- ___ c. Change the variable `@bpm-input-bg` to: `#d9ece3`

<code>@bpm-label-font-family</code>	<code>@bpm-font-family-base</code>	Loem Ipsum
<code>@bpm-label-font-weight</code>	<code>bold</code>	Loem Ipsum
<code>@bpm-label-text-color</code>	<code>inherit</code>	
<code>@bpm-input-bg</code>	<code>#d9ece3</code>	
<code>@bpm-input-bg-disabled</code>	<code>@bpm-neutral-lightest</code>	
<code>@bpm-input-color</code>	<code>@bpm-text-color</code>	

- ___ d. Click empty space outside the input control to refocus the mouse. The colors of the input controls are now highlighted with the chosen color.
 - ___ e. Save your work.
7. Apply the new theme to the HR Recruitment Processes process application.
- ___ a. Open the **Process App Settings** from the History menu on the top.



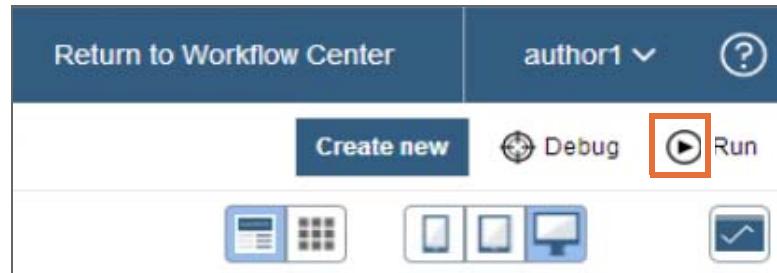
- ___ b. In the Coach Designer Settings section, click **Select**.

Theme:	<code>Classic</code>	System Data	Select...	New...	Clear
Heritage Coach XSL:	<code>CoachDesigner.xsl</code>	System Data	Select...	New...	Clear
Heritage Coach CSS:	<code>coach_designer.css</code>	System Data	Select...	New...	Clear

- ___ c. Select **Hiring Request Theme**. The theme is now changed to the new theme.

Theme:	<code>Hiring Request Theme</code>	Select...	New...	Clear	
Heritage Coach XSL:	<code>CoachDesigner.xsl</code>	System Data	Select...	New...	Clear
Heritage Coach CSS:	<code>coach_designer.css</code>	System Data	Select...	New...	Clear

- ___ d. Save your changes. The save generates the CSS that the browser uses to display the controls in the layout.
- ___ 8. Verify the theme changes
 - ___ a. Open the **Hiring Form** Client-Side Human Service.
 - ___ b. Click the **Run** icon at the upper-right side to display the coach with the custom theme. Initially when you view the coach, the server must generate the CSS and render the new coach on the screen, so a delay might occur before you see the effect.



The coach has the changes that you made in the Hiring Request Theme.

A screenshot of the 'Hiring Request' form. The form has a header with tabs: 'Requisition Details', 'Position Details', 'Recruiting Details', and 'Compensation Details'. To the right of the tabs is a 'More' button. The main content area contains several input fields with rounded corners and a light green background color. The fields are labeled: 'Requisition Number', 'Requester', 'Date Details', 'Date Of Request', 'Date Position Available', 'Hiring Comments', and a 'Submit' button at the bottom.

- ___ c. Close the browser window.

Part 3: Customize controls

The coach framework offers many ways of customizing input controls and views. This part of your exercise shows some of the popular ways to customize your coach.

- ___ 1. Add an icon to the Date Details panel.
 - ___ a. Select the **Requisition Details** tab on the coach.
 - ___ b. Click the **Date Details** panel.

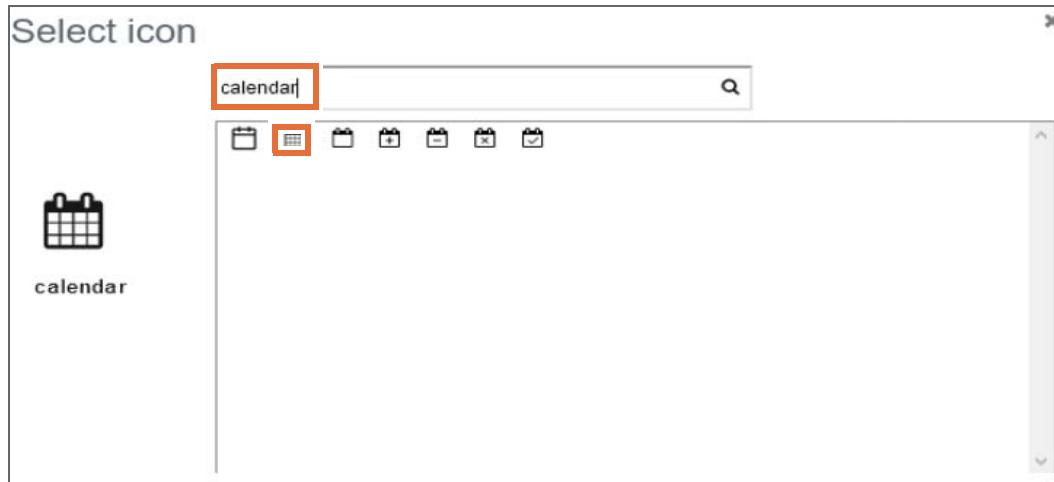
A screenshot of a user interface titled "Date Details". It contains two input fields: "Date Of Request" and "Date Position Available", both represented by light green rectangular boxes.

- ___ c. With the Date Details panel highlighted, click the **Select Icon** icon.



- ___ d. In the search field, enter calendar.

- ___ e. Select the second icon in the list labeled **calendar**.

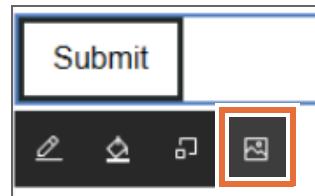


- ___ f. The Date Details section now displays a calendar icon in the header of the panel.



- ___ 2. Change the appearance of the submit button.

- ___ a. Select the **Submit** button on the canvas.
___ b. With the Submit button highlighted, click the **Select Icon** icon.

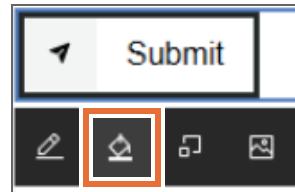


- ___ c. In the search field, enter send.

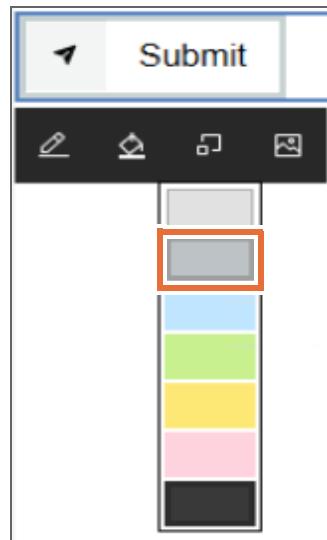
- ___ d. Select the first icon in the list.



- ___ e. With the Submit button highlighted, click the **Select Color** icon.



- ___ f. Select the **Primary** color.

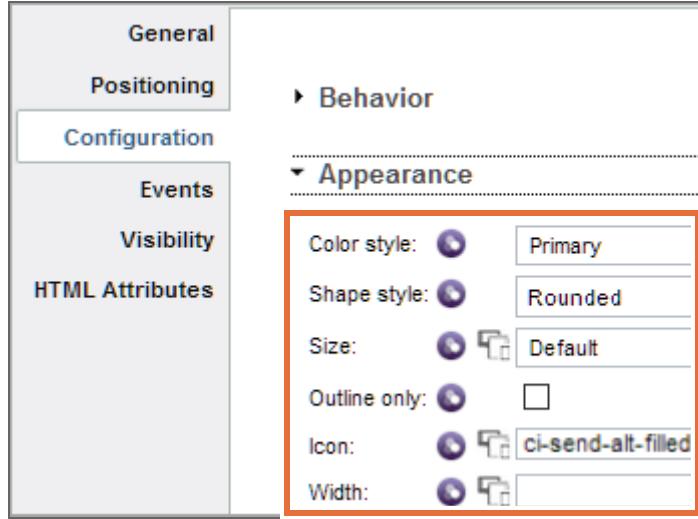


Note

The picker icons support the most popular configurations. For some configurations, you need to access the Configuration properties below.

- ___ g. With the **Submit** button highlighted, in the Properties section below, click the **Configuration** tab and expand the **Appearance** settings.

- ___ h. Notice the Color style and Icon which was configured in the prior steps. Set the shape style to **Rounded**. The button on the canvas reflects these settings as you make your configuration changes.



- ___ i. Click empty space outside the input controls to refocus the mouse.
 ___ j. The Submit button now displays the new configuration.



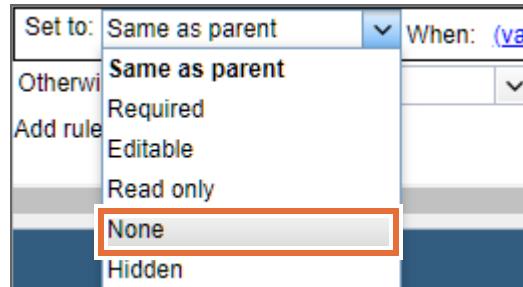
- ___ k. Save your work.
 ___ 3. Set the Num Employees Needed input control to visible only if the Multiple Employees Needed checkbox is selected.

It is common to hide controls and display them only when applicable. Since the Num Employees Needed input control is not needed unless there are multiple employees, this control is hidden until the Multiple Employees Needed checkbox is selected.

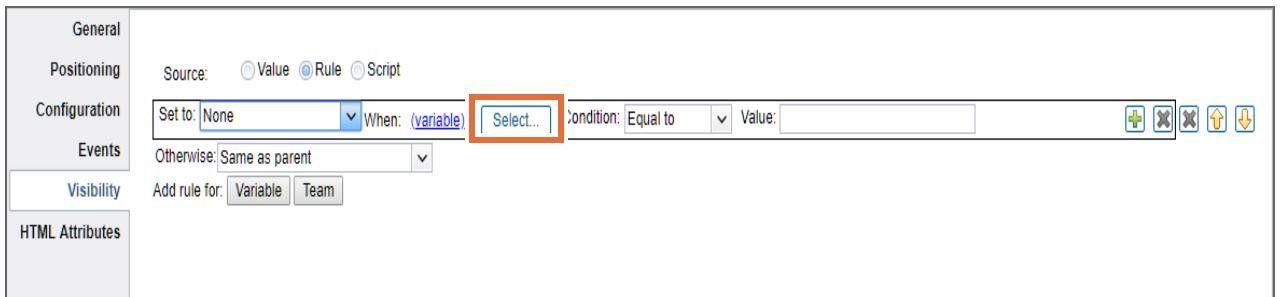
- ___ a. Click the **Recruiting Details** tab.
 ___ b. Select the **Num Employees Needed** input control
 ___ c. Select the **Visibility** properties section below
 ___ d. Click the **Rule** option for Source.
 ___ e. Click **Variable** for the Add rule for option.



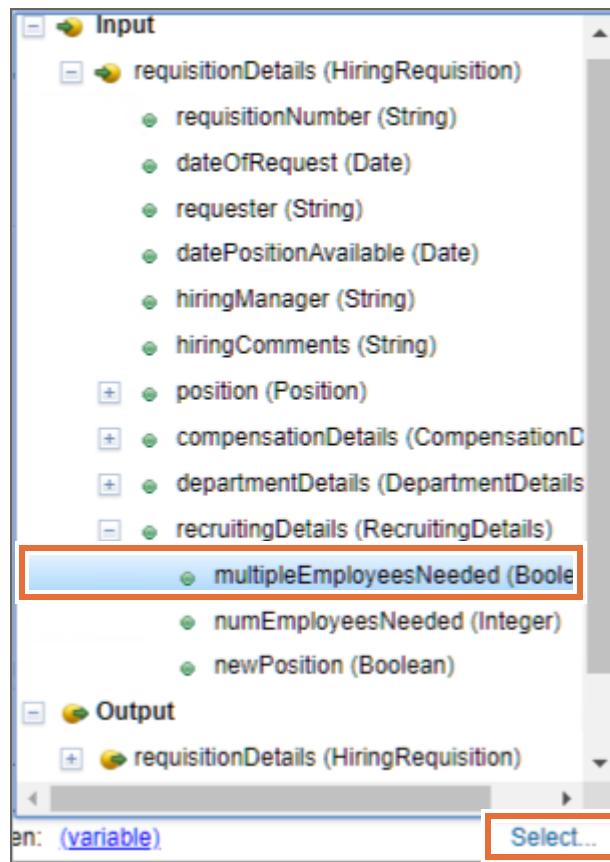
___ f. Select **None** for the Set to value.



___ g. Click **Select**.



___ h. Navigate to **Input > requisitionDetails > recruitingDetails > multipleEmployeesNeeded (Boolean)**



- ___ i. Your rule is now configured.

The rule determines when to set the input control to “none.” Since the input control is a Boolean, the value is depicted as true (Value checkbox marked) or false (Value checkbox not marked). This rule states:

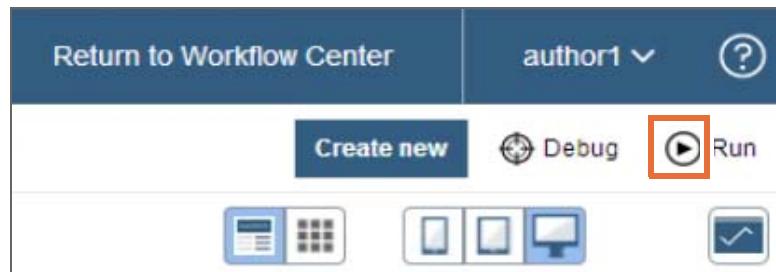
Set visibility to “None” when `multiple.EmployeesNeeded` has a false value otherwise inherit the visibility setting from the parent.

Information

Visibility settings are described below:

Visibility Setting	Description
Same as parent	Inherits visibility setting from the parent (in this case that would be the panel).
Required	Input control must have data before coach can complete
Editable	Input control is editable
Read only	Input control can only be viewed, not edited
None	Input control is not displayed including the space that it takes up on the coach
Hidden	Input control is not displayed, not including the space that it takes up. In this case, the space the control takes up does not change even though the control is hidden. This is useful if you do not want fields moving as a result of a field not displaying.

- ___ j. Save your work.
- ___ 4. Test Num Employees Needed input control visibility setting
- ___ a. Run the coach.



- ___ b. In the Coach window, click the **Recruiting Details** tab.

- ___ c. If you configured the visibility setting correctly, the **Num Employees Needed** input control is not visible since the **Multiple Employees Needed** value is false.

- ___ d. Test the setting by clicking the **Multiple Employees Needed** checkbox, setting the value to true. The **Num Employees Needed** input control displays below.
 ___ e. Close the browser window that you ran the coach in.

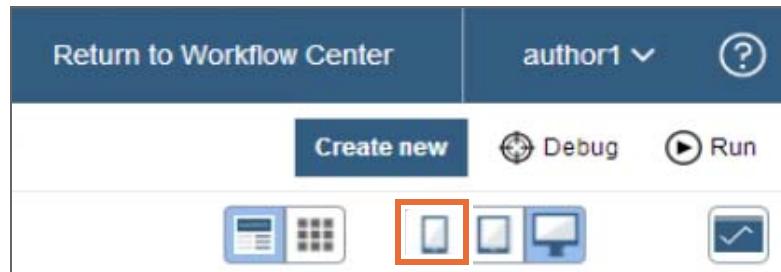
Part 4: Modify the mobile layout

You are going to change the layout of the coach to the mobile layout so that the coach is displayed correctly on a mobile device. You make a few of the sections collapsible and change the width of a few input controls based on the width of the browser window.

- ___ 1. Configure the width of the Submit button to respond to the screen size.
 - ___ a. Toggle the Properties section off by selecting the **Properties** icon in the bottom status bar.



- ___ b. Click the **Small screen** icon above the palette.



Hint

Press **Ctrl** and + or **Ctrl** and - to increase and decrease the browser zoom setting.

- ___ c. The Hiring Form is presented in a small screen size that resembles a phone. Your screen might look different if you are focused on a different tab.

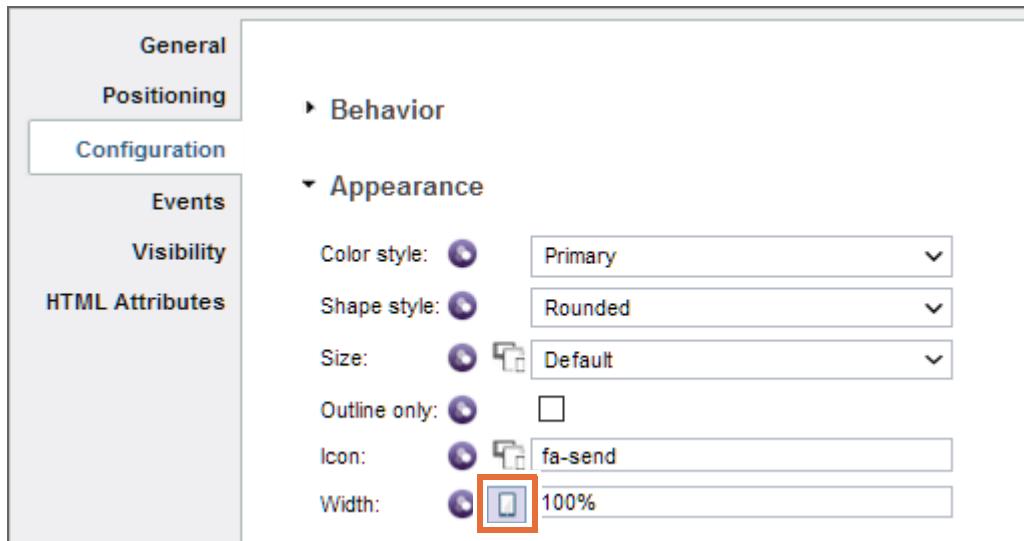
The image shows a mobile application interface for a hiring form. At the top, there's a status bar with signal strength, the IBM logo, the time (9:41 AM), battery level (100%), and a signal icon. Below the status bar is a navigation bar with several tabs: 'Requisition Details' (which is selected and highlighted in blue), 'Position Details', 'Recruiting Details', 'Compensation Details', 'Department Details', and a green '+' button. The main content area is titled 'Requisition Details' and contains the following fields:

- Requisition Number:** An empty input field.
- Requester:** An empty input field.
- Date Details:** A field with a calendar icon to its right.
- Date Of Request:** An empty input field.
- Date Position Available:** An empty input field.
- Hiring Manager Comments:** An empty input field.

At the bottom of the form is a 'Submit' button with a paper airplane icon. Below the form are three navigation icons: a vertical ellipsis, a square, and a left arrow.

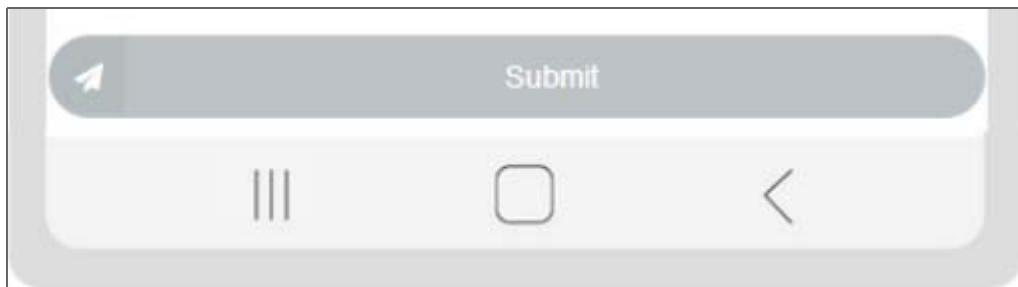
- ___ d. Toggle the **Properties** section on.
___ e. Select the **Submit** button.

- ___ f. Click the **Configuration** properties tab and expand the **Appearance** properties settings.
- ___ g. Enter 100% in the **Width** configuration value. Refocus the mouse.



After you refocus the mouse, the icon changes to the small screen icon. This means that the setting is only made for the small screen size. You can configure different widths for the three different screens. The original setting for the large screen size has not changed. Make sure you refocus the mouse after changing the settings to verify the correct screen size setting.

- ___ h. The Submit button now expands to 100% of the screen.



- ___ 2. Configure the width of the Date Details panel to respond to the screen size.

You can change the width settings in either the Configuration tab or the Positioning tab. When adding values for multiple screen sizes, it is recommended to use the Positioning tab.

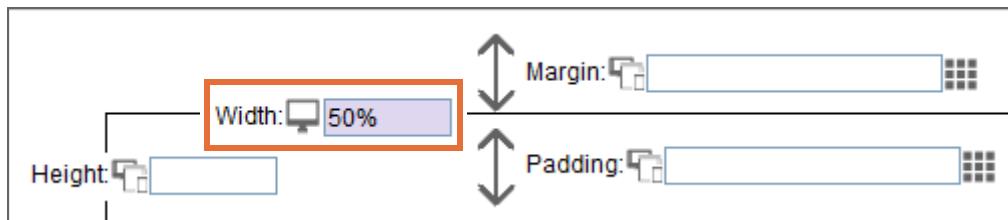
- ___ a. On the **Requisition Details** tab, select the **Date Details** panel.



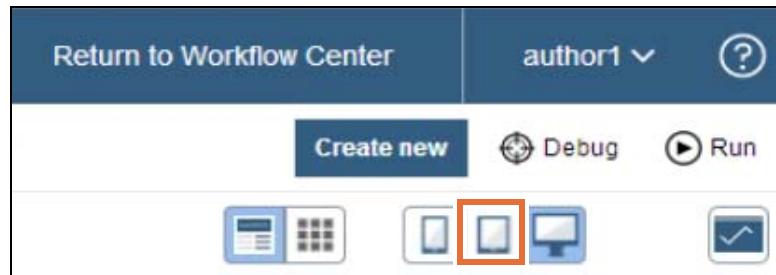
- ___ b. Click the **large screen** icon above the palette.



- ___ c. Set the **Positioning > Width** properties to 50%. The entered width is applicable only for large screen devices. Click the space outside the input control to refresh the icon. The icon is set to large screen.
- ___ d. Refocus the mouse.



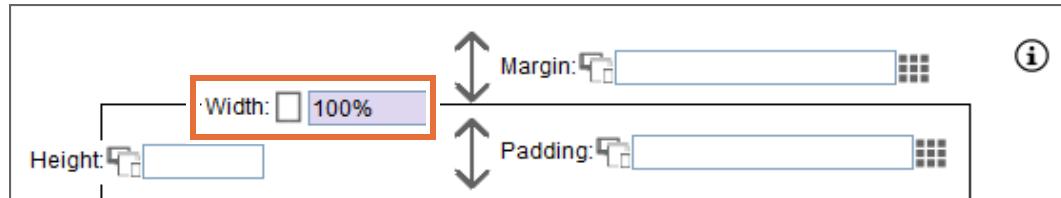
- ___ e. Click the **medium screen** icon above the palette.



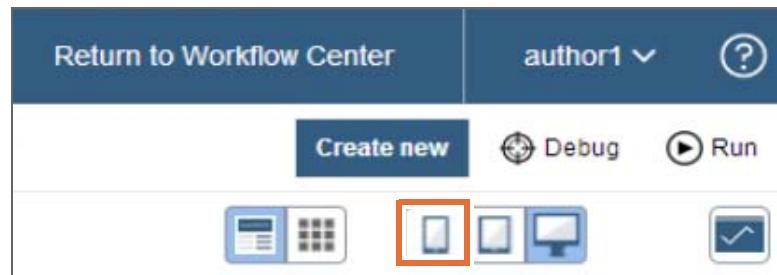
- ___ f. Set the **Positioning > Width** properties to 100%. The entered width is applicable only for medium screen devices. Click the space outside the input control to refresh the icon. The icon is set to medium screen.

When you set the tablet setting (medium screen), the small screen automatically is set to 100%. This setting is inherited from the tablet format setting, so you do not have to set it to 100%.

- ___ g. Refocus the mouse.



- ___ h. Click the **small screen** icon above the palette.



- ___ i. Verify that the width settings are set to 100%.

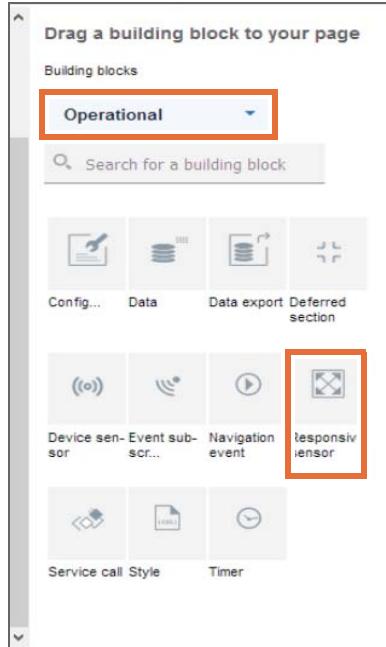


- ___ j. Save your work.

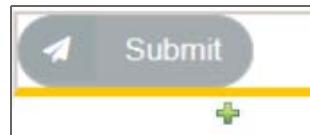
- ___ 3. Configure a responsive sensor.

- ___ a. Click the **large screen** icon above the palette to return to the web page layout.

- ___ b. Select **Operational** in the **Building blocks** panel and select **Responsive sensor**



- ___ c. While holding down the left mouse button, drag the **Responsive sensor** below the **Submit** button. Let go of the mouse button when the bottom border displays a gold highlight.



- ___ d. While holding down the left mouse button, drag a **Configuration** below the Responsive sensor. Let go of the mouse button when the bottom border displays a gold highlight.
- ___ e. Under the **Configuration** properties, enable the option to turn **Debugging On**.



- ___ f. Access the Layout building blocks and while holding down the left mouse button, drag a vertical layout over the **Responsive Sensor**
- ___ g. When the **Drop content here** section is highlighted, let go of the mouse button

The Responsive Sensor requires a horizontal or vertical layout to correctly calculate the screen size. When the coach is run, the browser width is displayed in this Responsive Sensor as soon as the browser is resized to test the responsive settings.

- ___ h. Save your work.
- ___ 4. Verify the coach display for different screen sizes.
 - ___ a. Click the **Run** button.

A new browser window displays the coach and the width is shown at the bottom. The Date Details panel in this image spans the entire browser window because the initial width is set to a medium screen size setting. Depending on the size of your screen, the layout width might vary.

The screenshot shows a web-based form with the following structure:

- Header:** A navigation bar with tabs: Requisition Details, Position Details, Recruiting Details, Compensation Details, and a menu icon.
- Section 1:** Requisition Details
 - Requisition Number:** A large input field.
 - Requester:** A large input field.
- Section 2:** Date Details
 - Date Of Request:** A large input field.
 - Date Position Available:** A large input field.
- Section 3:** Hiring Comments
 - Hiring Comments:** A large input field.
- Footer:** A "Submit" button with a paper airplane icon and a note indicating the width of the page is 800px.

Recall the sizes for each screen.

Table 2. Screen sizes

Icon	Size	Resolution
	Small	640 pixels or less
	Medium	641 - 1024 pixels
	Large	More than 1024 pixels

- ___ b. Slowly reduce the width of the browser. When you pass the 640 pixels threshold, the Submit button expands to 100% of the screen width.

The screenshot shows a user interface for a requisition form. At the top, there are three tabs: "Requisition Details" (selected), "Position Details", and "Recruiting Details". To the right of the tabs is a menu icon. The main area is titled "Requisition Details". It contains several input fields:

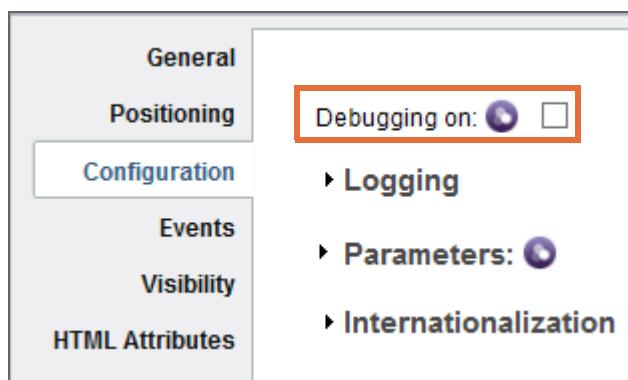
- "Requisition Number" (empty)
- "Requester" (empty)
- "Date Details" (with a calendar icon)
- "Date Of Request" (empty)
- "Date Position Available" (empty)
- "Hiring Comments" (empty)

At the bottom is a grey "Submit" button with a paper airplane icon. A tooltip "Width: 639px" is shown near the bottom center of the page. The entire "Hiring Comments" section and the "Submit" button are highlighted with a red border.

- ___ c. Now expand the browser window. When the width passes 1024 pixels, the Date Details panel contracts to 50% of the browser width because of the responsive control settings based on the large screen width setting.

The screenshot shows a web application interface for 'Requisition Details'. At the top, there are tabs for 'Requisition Details', 'Position Details', 'Recruiting Details', 'Compensation Details', and 'Department Details'. The 'Requisition Details' tab is active. Below the tabs, the main content area has a title 'Requisition Details'. It contains several input fields: 'Requisition Number' (empty), 'Requester' (empty), 'Date Details' (containing 'Date Of Request' and 'Date Position Available' fields, both empty and highlighted with a red border), and 'Hiring Comments' (empty). At the bottom left is a 'Submit' button with a paper airplane icon. A status message 'Width: 1025px' is displayed at the bottom right.

- ___ d. Close the browser window.
 ___ e. Click the **Configuration** on the canvas. On the **Configuration** properties menu, clear the **Debugging on** checkbox to remove the width display on the coach.



- ___ f. Save your work.

You have completed this exercise.

Playback 1: User interface design and implementation is now complete. In this exercise, you completed the following:

- Created tabs on a coach
- Changed the appearance of a coach by applying a custom theme
- Changed the coach layout for a mobile format
- Configured responsive controls
- Debugged the coach by using a responsive sensor

As part of the development process, you now review the playback and examine its functions in the Process Portal. In the next exercise, you test the following functions by performing a playback:

- Does the process follow the correct paths?
- Does the application work as expected in the Process Portal?

As part of the playback, you demonstrate the process application in the Process Portal. You then create a toolkit in the Workflow Center and export your process application.

End of exercise

Exercise 7. Playback 1: Conducting the Playback session

Estimated time

01:00

Overview

This exercise covers how to conduct a Playback of your process. The exercise demonstrates the process, following various paths that flow from the exclusive gateways in the process and demonstrates tasks that are assigned. It describes the task that is created in the Process Portal inbox, depending on the swimlane and routing settings for an activity. You also create a toolkit to store and share these assets.

Objectives

After completing this exercise, you should be able to:

- Log on to the Process Portal and create an instance of a process
- Demonstrate that the process follows the various paths modeled
- Use Process Portal to view the state of activities in a process
- Create a toolkit
- Create a snapshot in the Workflow Center
- Export the process application

Introduction

The following list provides a quick view of what to accomplish in the playback portion:

1. Log on to the Process Portal and create a process instance.
2. Log on to the Process Portal as two different teams to show that the correct process team is being assigned the correct task.
3. Complete the human activities.
4. Demonstrate that the correct path is followed.
5. Change variable values and demonstrate following a different path.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Prepare for the playback](#)

[Part 2: Demonstrate the Review Needed path](#)

[Part 3: Demonstrate the Review Not Needed path](#)

[Part 4: Demonstrate activity tracking](#)

[Part 5: Reset the environment](#)

[Part 6: Create the Hiring Requisition Toolkit and take a snapshot](#)

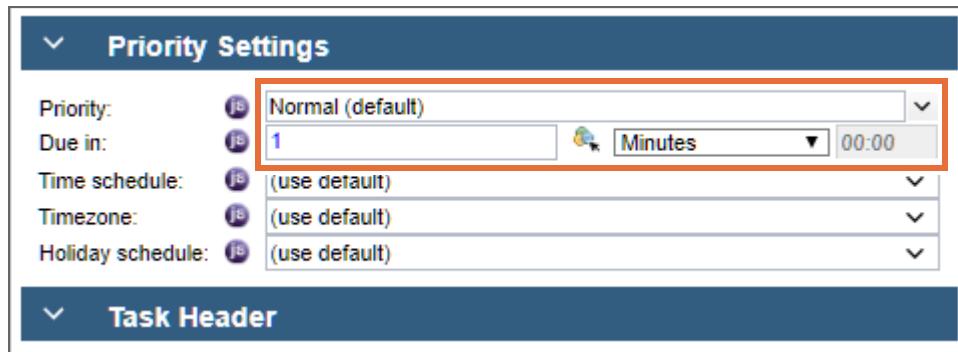
Exercise instructions

Part 1: Prepare for the playback

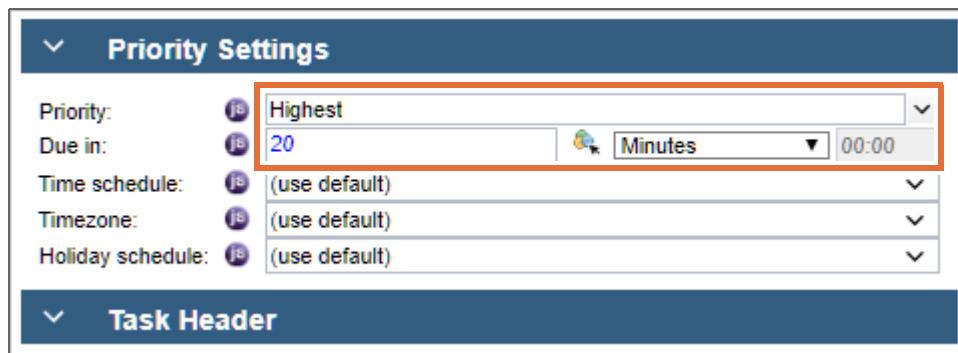
To enable a well-run playback in a customer engagement, you should practice running through the model beforehand. It is suggested that you run through the playback with a colleague several times to ensure a smooth presentation. This can also aid in finding unexpected issues.

It is common to prepare your environment for the playback to exemplify specific functionality explicitly demonstrated for the playback. In the case for this playback, to facilitate activities that time out in order to demonstrate different activity statuses in the Gantt chart, you change the timeout settings in two activities.

- ___ 1. Change the priority and timeout value for the **Complete Hire Request** activity.
 - ___ a. Open the **HR Recruitment Process** process application.
 - ___ b. Open the **Hiring Request Process**.
 - ___ c. Click the **Complete Hire Request** activity.
 - ___ d. In the **Priority Settings** section of the **Implementation** properties at the bottom, leave the **Priority** setting as the default and set the activity to be due in 1 minute.



- ___ 2. Change the priority and timeout value for the **Review Posting** activity.
 - ___ a. Click the **Review Posting** activity.
 - ___ b. In the Priority Settings section of the **Implementation** properties at the bottom, select **Highest** for the Priority setting.
 - ___ c. Set the activity to be due in 20 minutes.



- ___ d. Save your work.

When you demonstrate viewing work items later in this exercise, the different priority settings are displayed.

Part 2: Demonstrate the Review Needed path

During this playback session, you demonstrate the process, following various paths that flow from the exclusive gateways in the processes, and demonstrate tasks that are assigned.

You accomplish a process playback session just as you would do when you seek consensus to move the project to the next playback phase. You log in as different users of the process to demonstrate tasks that different users accomplish in the process, demonstrating coaches that you created in the current playback phase.

- ___ 1. Log on to the Process Portal.
- ___ a. Maximize the **IBM Process Center Quick Start** browser window from the Windows taskbar.
- ___ b. Click the **Process Portal** link.

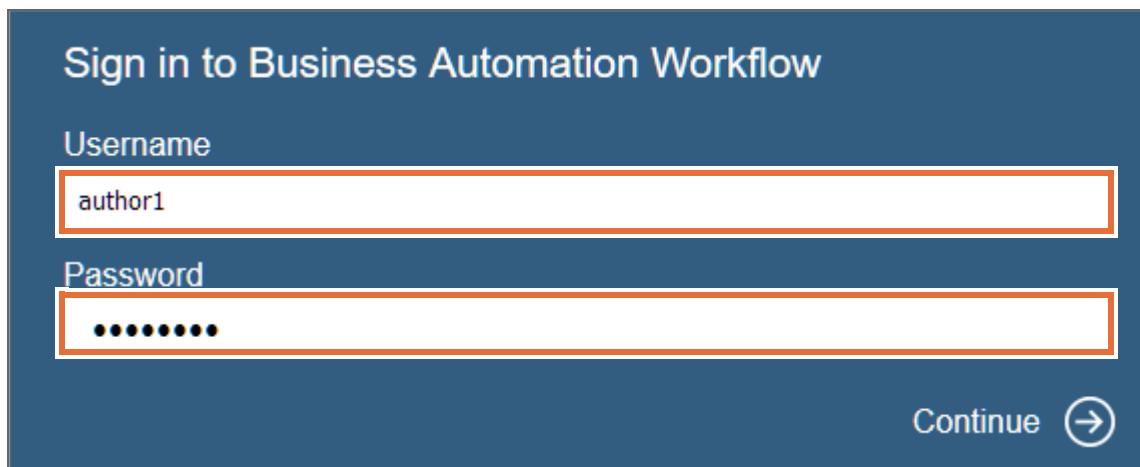
Process application consoles & tools

Create, manage, share, and test high-level containers such as process applications and toolkits.
[Process Center Console](#)

Test and administer the business user interface for completing tasks.

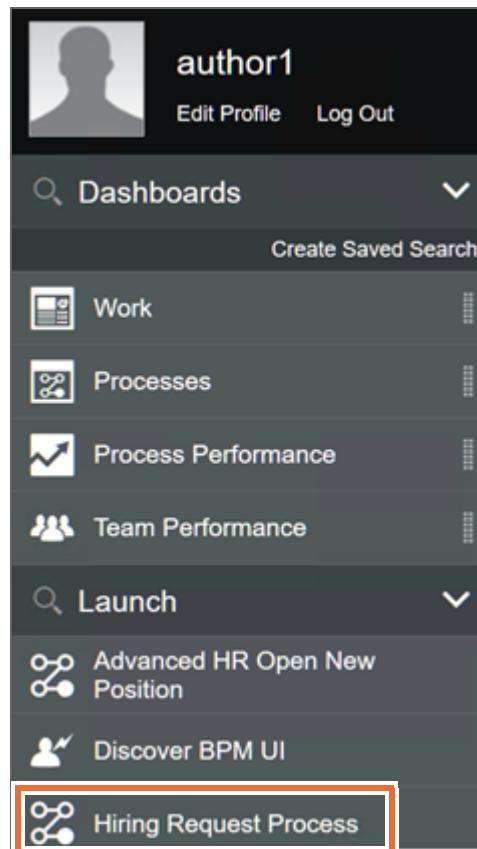
[**Process Portal**](#)

If prompted, log in with `author1` as the username and `passw0rd` as the password.



The dialog box has a dark blue header with the text "Sign in to Business Automation Workflow". Below the header are two input fields: "Username" containing "author1" and "Password" containing "*****". To the right of the password field is a "Continue" button with a right-pointing arrow.

- __ 2. Complete the first activity, Submit Hiring Request, in the Process Portal.
- __ a. In the left frame, click **Hiring Request Process** to start an instance of the process.



- __ b. A notice is displayed to indicate that the process is started.

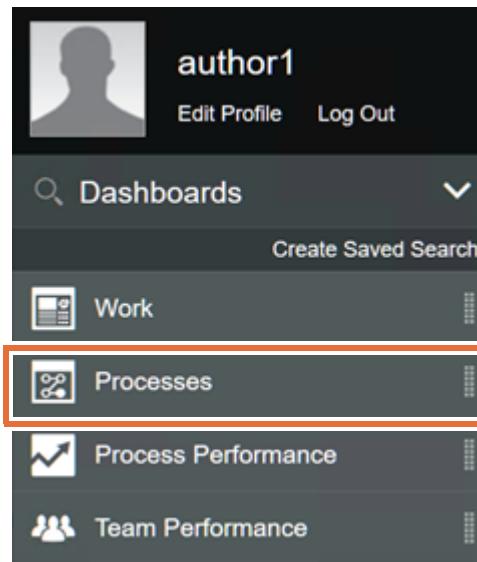
You received a new task titled Step: Submit Hiring Request

The first task starts automatically in the portal because you assigned the first task to Lane and Last User. A window is displayed with the human service coach.

- ___ c. Click the **Recruiting Details** tab, and select the **New Position** checkbox so that it follows the Review Needed path. Click **Submit**.

The screenshot shows a form with several tabs at the top: Requisition Details, Position Details, Recruiting Details, Compensation Details, and Department Details. The Recruiting Details tab is active and highlighted with a red box. Below it, there is a section titled 'Recruiting Details' containing two checkboxes: 'Multiple Employees Needed' (unchecked) and 'New Position' (checked). At the bottom left of the form is a 'Submit' button, which is also highlighted with a red box.

- ___ 3. View the task that is assigned to the next activity, **Approve New Hire Request**, in the process.
___ a. In the Library click **Processes**.



Troubleshooting

If the process is not displayed, stop the Process Center server, restart Windows, then restart the Process Center server.

- __ b. The Processes page opens. Click **Hiring Request Process**.

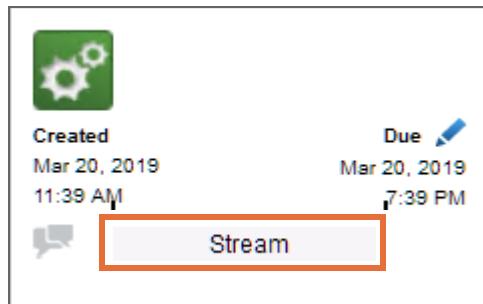
The screenshot shows a search bar at the top with the placeholder "Type a search filter". To the right of the search bar are two buttons: "Active" and "Completed". Below the search bar, there is a list of processes. The first process in the list is highlighted with a red border and labeled "Hiring Request Process:203". To the left of this process is a circular icon with a green and grey gradient. To the right of the process name is the due date "Due: March 20, 2019 7:39 PM". Below the process name, it says "Created: March 20, 2019 11:39 AM". At the bottom of the list, it says "Showing 1 out of 1 results".

The process instance ID is the number next to the name of the process. The process instance ID can be different in your environment. If you have more than one instance, click the one with the highest instance number (the number after the colon).

- __ c. The process tab displays data about the instance. The bottom section that is labeled **Tasks** shows the next incomplete task in the process. Do **NOT** click the task. This task is assigned to the **General Managers** team.

The screenshot shows the details of the selected process instance. At the top, there is a "Data" tab which is currently active, indicated by a downward arrow icon. Below the "Data" tab, it says "No details found.". To the right of the "Data" tab, there are two other tabs: "Documents" and "Tasks". The "Documents" tab is also active, showing the message "No documents or folders were found at this level.". The "Tasks" tab is shown below, with a list of tasks. The first task in the list is highlighted with a red border and labeled "Step: Approve New Hire Request". To the left of this task is a circular icon with a green and grey gradient. To the right of the task name is the due date "Due: March 20, 2019 12:45 PM". Below the task name, it says "General Managers". At the top right of the "Tasks" section, there are two buttons: "Open" and "Completed".

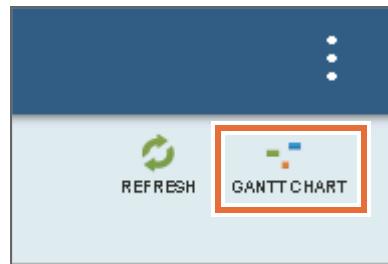
- ___ d. On the right, the creation date and the process due date are shown, along with a timeline that shows the current duration of the instance. Click the **Stream** area to view the actions that occur for this instance. Sometimes it takes a moment for the word Stream to appear.



- ___ e. To add to the stream, type Please approve into the stream field and click **Post**. Your comment appears as the most recent entry in the stream.

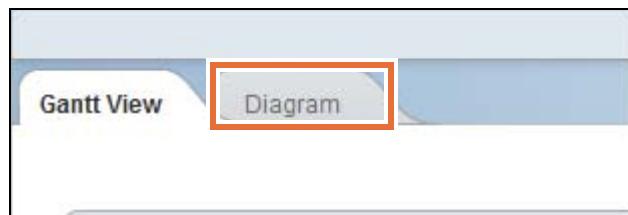
A screenshot of the Stream interface. At the top, there is a process instance card with creation details and a Stream button. Below it, a modal window is open with a text input field containing "Please approve" and a blue "Post" button, which is also highlighted with a red rectangular border. The main Stream area shows several previous entries: 1. A system message about a task being created for General Managers at 11:45 AM. 2. A user message from "author1" indicating they completed a task at 11:45 AM, with a link to "Show audit trail". 3. A system message about the task being claimed by "author1" at 11:39 AM. 4. A final system message about the task being created for "author1" again at 11:39 AM.

- ___ f. Click the **GANTT CHART** icon in the upper-right corner to go to the Gantt chart view.

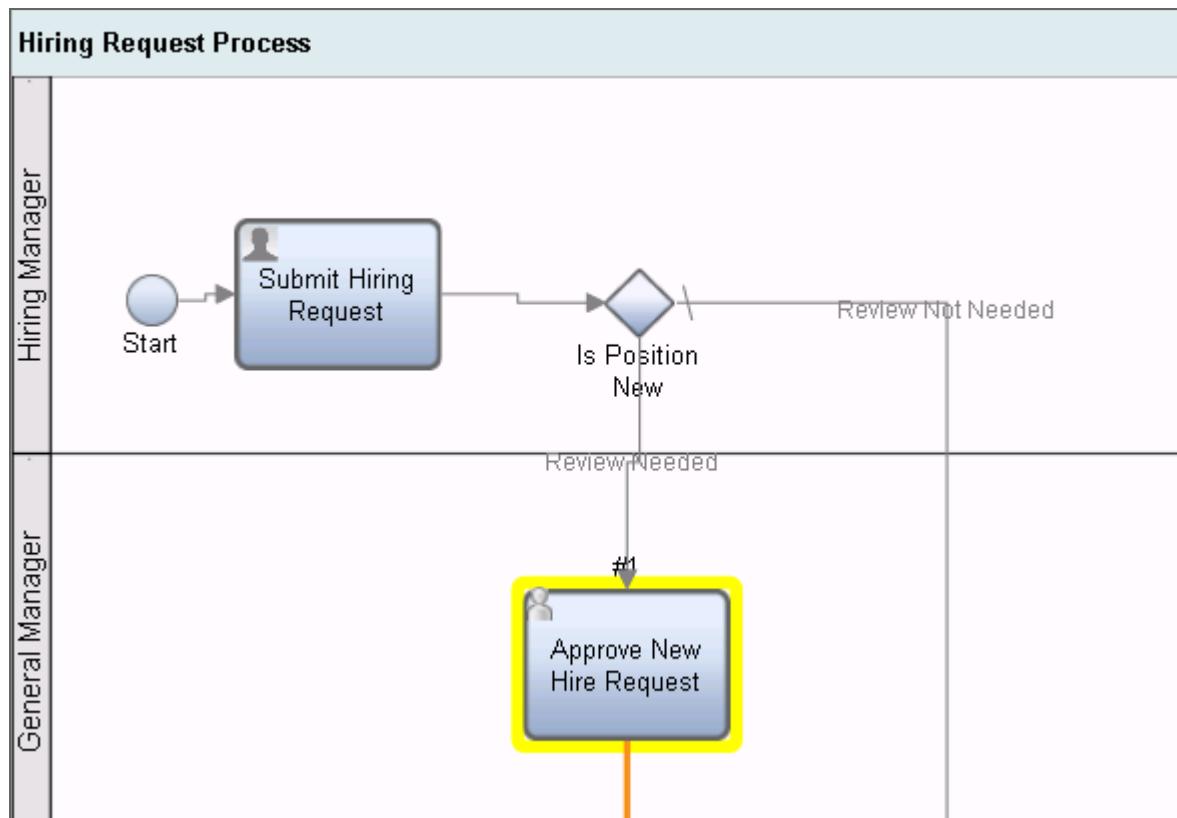


The Gantt chart is part of the Process Performance tab. When you click the link, you open the **Process Performance > Instance Details** page for the process instance. Wait for a few seconds as the loading of the page might take some time. You review the Gantt Chart later in this exercise after completing some activities. For now, you review the model and where you are at in the process.

- ___ g. Click the **Diagram** tab for a model view of the same process instance.



- ___ h. The model view of the process instance is displayed.



This view shows the process with the Approve New Hire Request task that is highlighted for the **Review needed** path, and in the next part of this exercise you work on the **Review not needed** path.

- ___ 4. Complete the Approve New Hire Request activity.
 - ___ a. In the library, click **Log Out** to log out of the Process Portal.

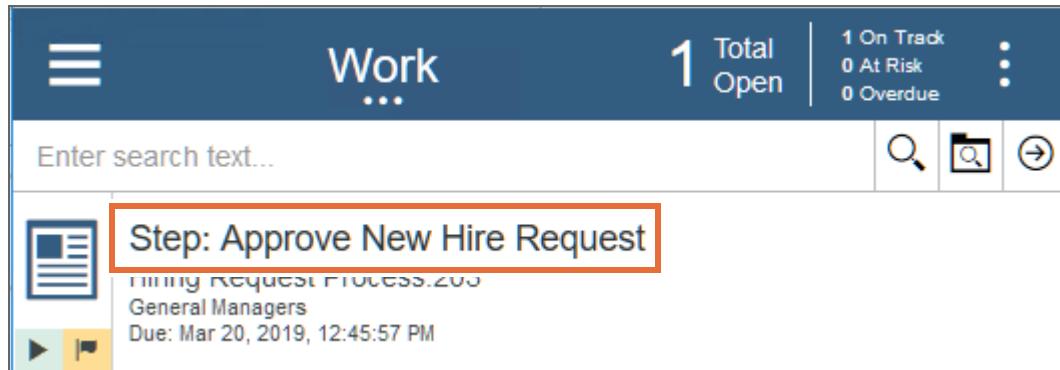


- ___ b. Log in to the Process Portal with the **Username** user1, **Password** passw0rd, and click **Continue**.

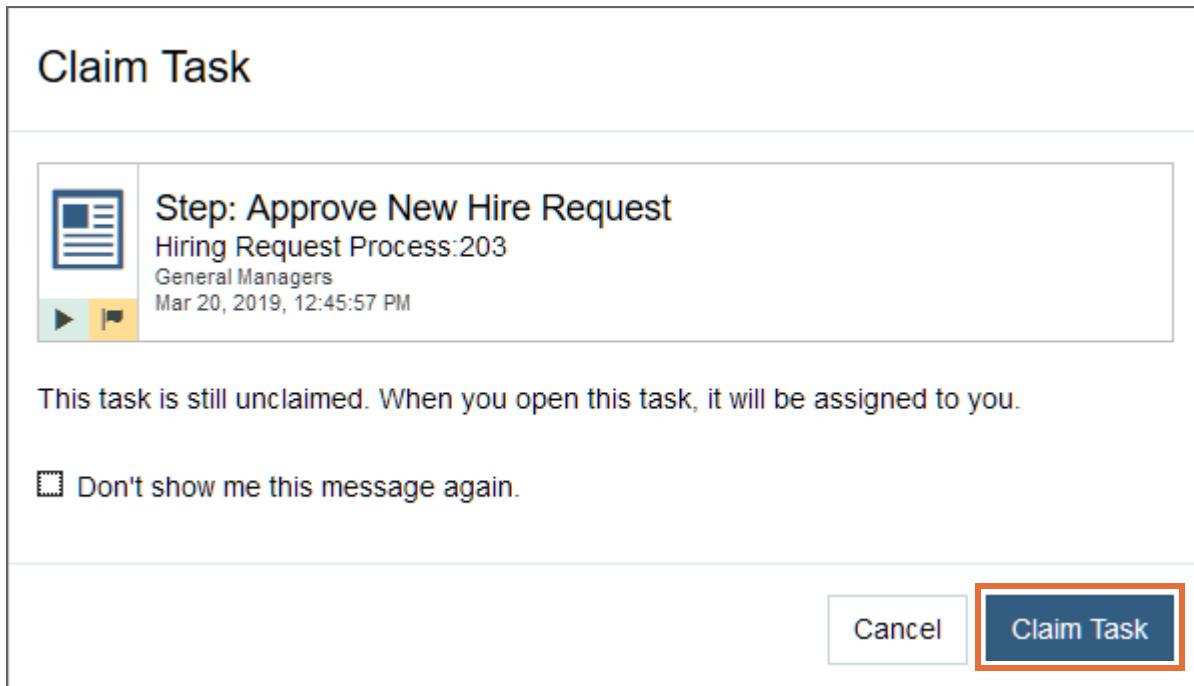
Licensed Materials - Property of IBM. © Copyright 2008, 2018 IBM Corporation. IBM, the IBM logo, and WebSphere are trademarks of IBM Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

IBM

- ___ c. A task is in the inbox for the Hiring Request Process process. The task stopped at the Approve New Hire Request activity. Notice that the task is assigned to the General Managers team, of which user1 is a member. Click the **Step: Approve New Hire Request** task. Depending on the amount of time spent, your task might be either On Track, At Risk, or Overdue.



- ___ d. Click **Claim Task**, and the task is assigned to you.



- ___ e. A human service coach is displayed for the Approve New Hire Request activity.

- __ f. Drag the size of the browser window to display all tabs.

Requisition Details

Requisition Number

Requester

Date Details

Date Of Request

Date Position Available

Hiring Comments

Submit

- __ g. Enter any data in the fields and complete this task by clicking **Submit** in the human task service.
- __ h. Click **Processes** in the library.

- __ i. Click the **Hiring Request Process** instance with the same instance ID you worked on in the last step. The next task is assigned to `author1`. Click the task **Step: Complete Hire Request** and you get a message that states “You are not authorized to see the details of this task.”

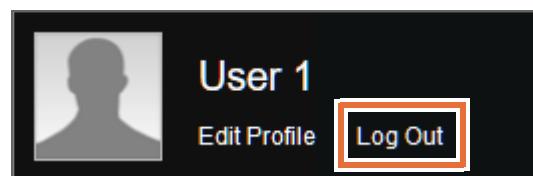
The screenshot shows the Process Portal interface. The top navigation bar includes 'Data', 'Documents', and 'Tasks'. The 'Data' section shows 'No details found.' The 'Documents' section shows 'No documents or folders were found at this level.' The 'Tasks' section shows a single task:

- Step: Complete Hire Request** (highlighted with an orange border)
- Due: March 20, 2019 5:02 PM
- Assigned to author1
- You are not authorized to see the details of the task.

The right sidebar shows 'Created Mar 20, 2019 11:39 AM' and 'Due Mar 20, 2019 2:30 AM'. Below that is the 'Activities' section with a 'Filter' button and status indicators: 'Ready | In Progress | Completed | All'. It also states 'No activities were found.'

This would be the place to start the discussion around task status tracking. This task shows as overdue due to the 1 minute time-out you configured earlier. If the task is not overdue, wait one minute and refresh the page (**F5**).

- __ j. Log out of the Process Portal.



If Log Out is not displayed, click the **Main Menu** icon on the left side.



Part 3: Demonstrate the Review Not Needed path

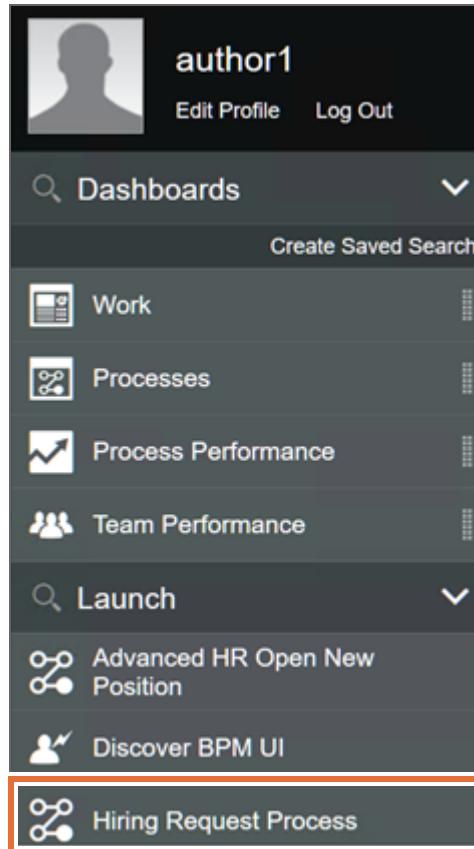
Now that you demonstrated the **Review Needed** path, this next part of the playback demonstrates the **Review Not Needed** path. It follows the same approach where you simulate the real-world experience of the different users inside the process as they log on to the portal and complete their tasks inside the process.

- 1. Show the process as it follows the **Review Not Needed** path, and create an instance to test the “existing position” scenario.

- a. Log in to the Process Portal with `author1` as the **username** and `passw0rd` as the **Password**.

The incomplete step (Complete Hire Request) from the prior path shows up in the Work page.

- b. Create an instance of the **Hiring Request Process** by clicking the link in the Launch section.



- c. You immediately see in a dialog box that you received a new task. A window is displayed with the human service coach.

- ___ d. Click the **Recruiting Details** tab, and ensure that the **New Position** checkbox is cleared so that it follows the Review Not Needed path.

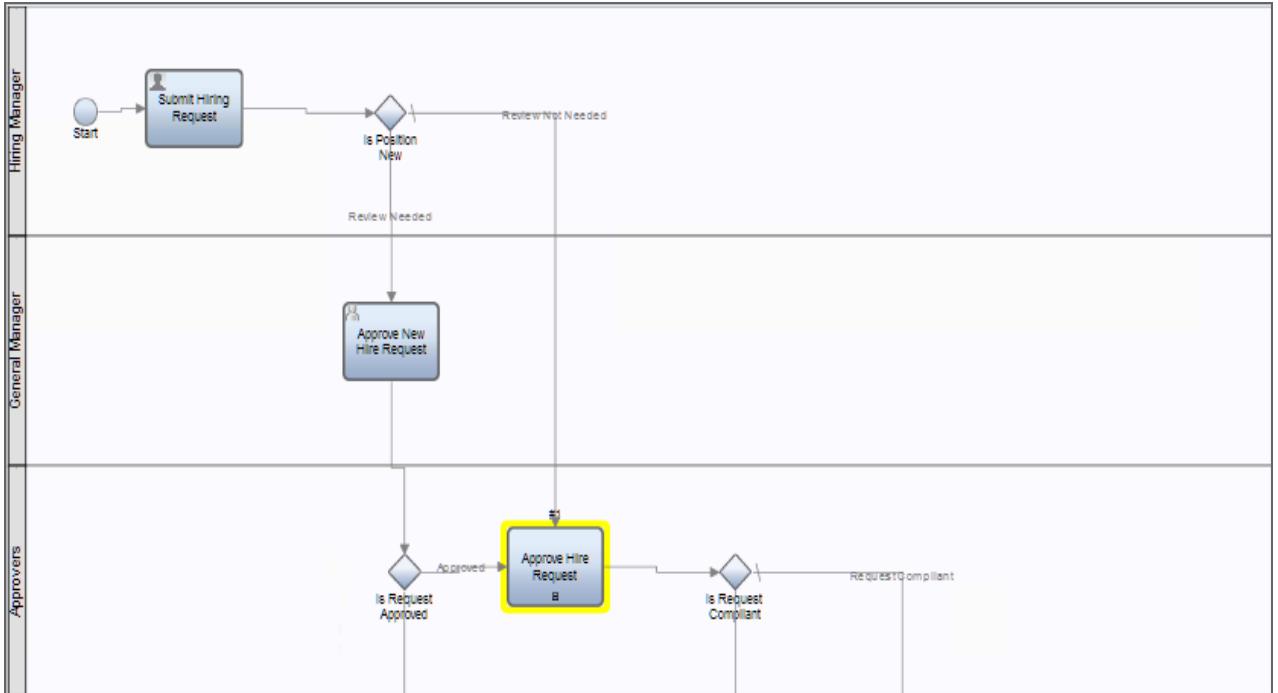
- ___ e. Enter any data for the other fields and click **Submit**.

You now have two items on your Work page.

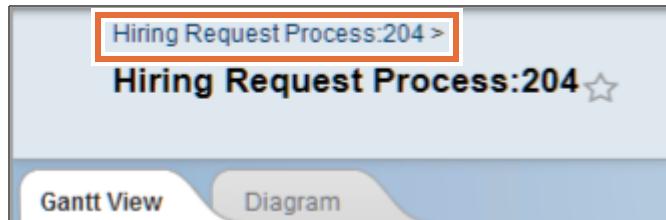
- ___ 2. Complete the process.

- ___ a. In the library, click **Processes**.
- ___ b. Click the process with the highest process instance ID. Your process instances are highlighted in green, yellow, or red to reflect their due date status (On Track, At Risk, Overdue).

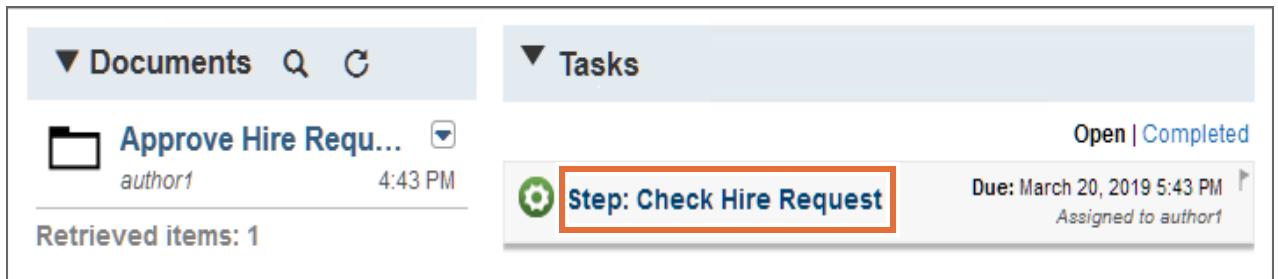
- ___ c. This time, the process follows the Review Not Needed path. Instead of passing to **Approve New Hire Request**, the process steps through the **Approve Hire Request** linked process. Click **GANTT CHART**, and then click **Diagram** to view the process diagram.



- ___ d. Click the link at the top to return to the Task.



- ___ e. Click the **Step: Check Hire Request** task link. Recall this is the first task for the linked process.



- ___ f. You are presented with the default user interface because you did not create a coach for this task. Click the Details tab to view details for the process instance. Click **Done** to complete the task.

Step: Check Hire Request

Overview Details

This is a sample coach for the following activit

Step: Check Hire Request

Done Complete Later

- ___ g. Now the process moves to the next task. Click **Step: Override Hire Request**.

▼ Documents Q C

▼ Tasks

Approve Hire Requ... author1 4:43 PM Open | Completed

Retrieved items: 1

Step: Override Hire Request Due: March 20, 2019 6:13 PM All Users

- ___ h. Click **Claim Task**.

Claim Task

This task is still unclaimed. When you open this task, it will be assigned to you.

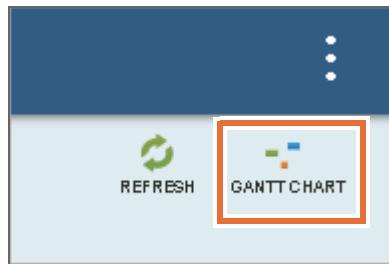
Don't show me this message again

Claim Task

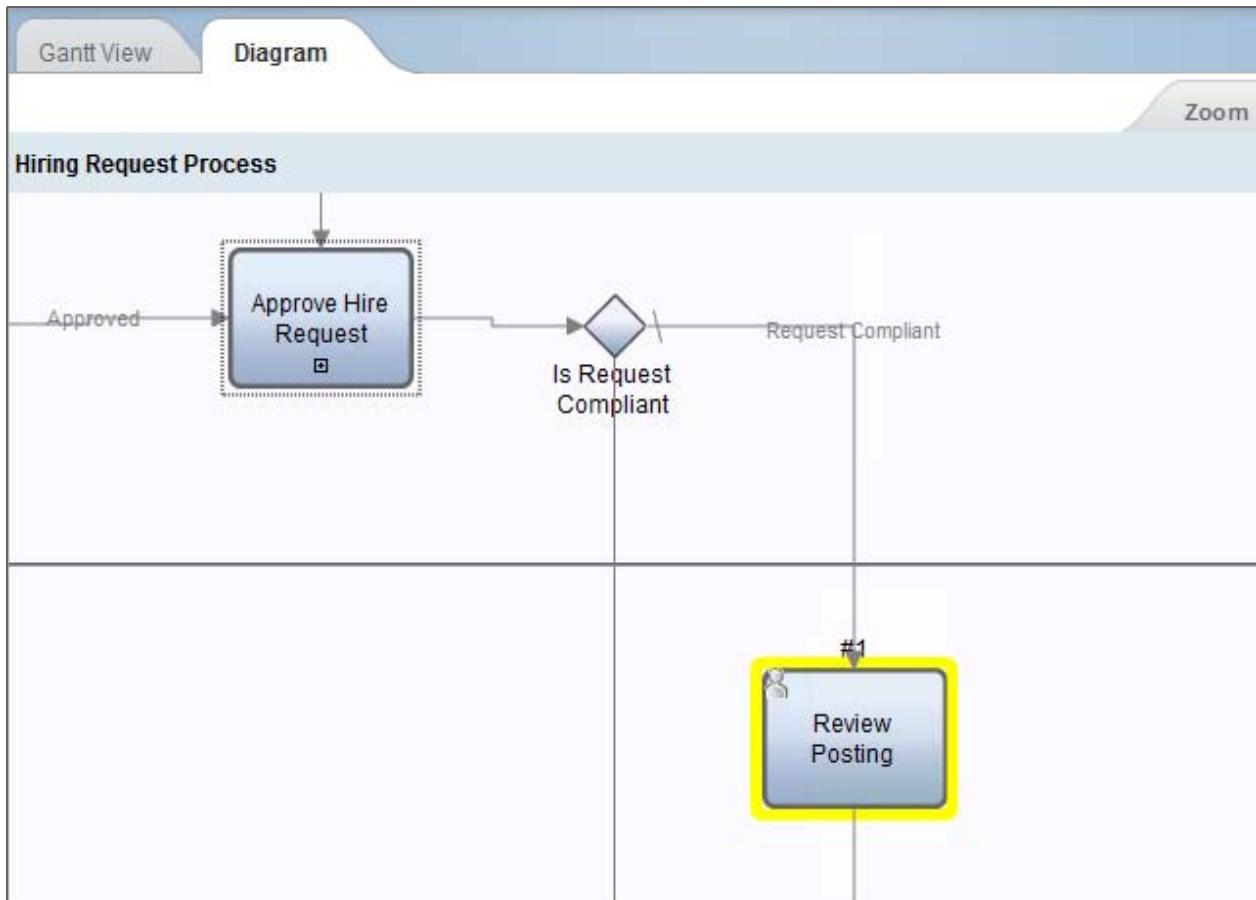
- ___ i. Click **Done** to complete the task.

- ___ j. The process now moves to **Step: Review Posting**.

- __ k. Click the **GANTT CHART** icon on the upper-right corner of the window.



- __ l. Click the **Diagram** tab for a model view of the same process instance.



This time, the process follows the **Review Not Needed** path. Instead of passing to **Approve New Hire Request**, the process stepped through the **Approve Hire Request** linked process, through the **Is Request Compliant** gateway, and then to the **Review Posting** activity. The process followed a different path from the first instance, which demonstrates that the **Is Position New** gateway and process flow data are working correctly when driven by business data on a coach. If the **Complete Hire Request** activity is active instead of the **Review Posting** activity, verify you configured the **Is Request Compliant** gateway correctly.

Part 4: Demonstrate activity tracking

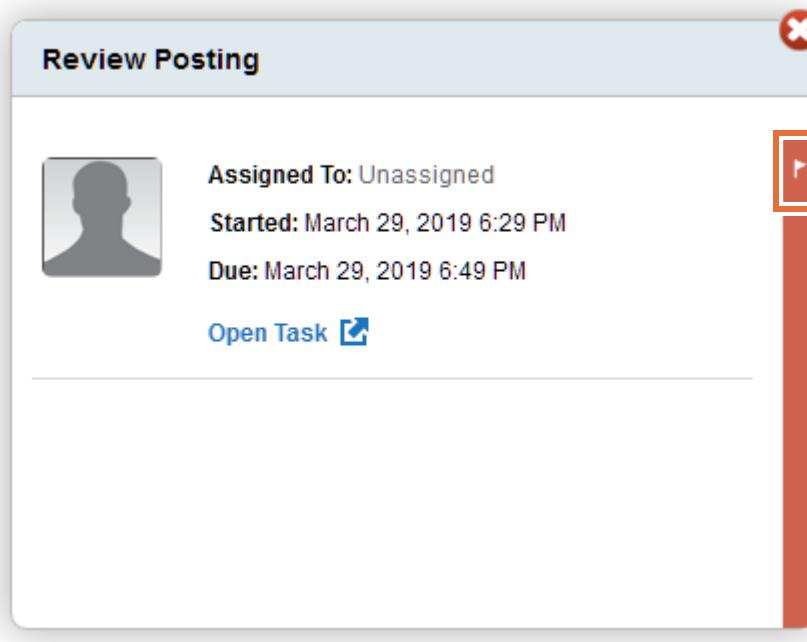
Now that the two processes have arrived at the activities that you changed settings for, you can continue the conversation around priority tracking and activity timeouts. It is a common practice to create mock scenarios when you perform playbacks to demonstrate specific functionality. In this case, you demonstrate how Process Portal flags activities that are highest priority and those that have passed their due date.

- 1. Display the status of the Review Posting activity in the Gantt chart.
 - a. Click the **Gantt View** tab.
 - b. Use the Zoom control to zoom out so you can see the prior activities.
 - c. Mouse over the prior activity to display the name of the task.

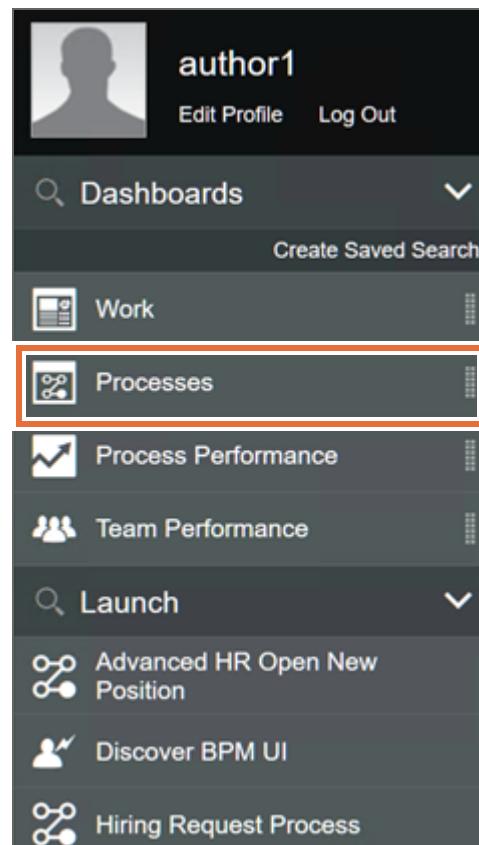


- d. This chart displays the time that it took to complete each task. Your chart might look different depending on the amount of time that is spent on each activity. In the Gantt chart, four activities are displayed. Roll the mouse over the other three completed activities to display their names.
- e. The first three activities were completed on time. The Review Posting activity is not complete. Depending on the amount of time spent, the Review Posting activity might also be past due. In the screen capture, the activity is displayed in yellow because it is at risk of becoming overdue. Click the **Review Posting** activity to view more details.
- f. A pop-up window displays more details for the Review Posting activity.

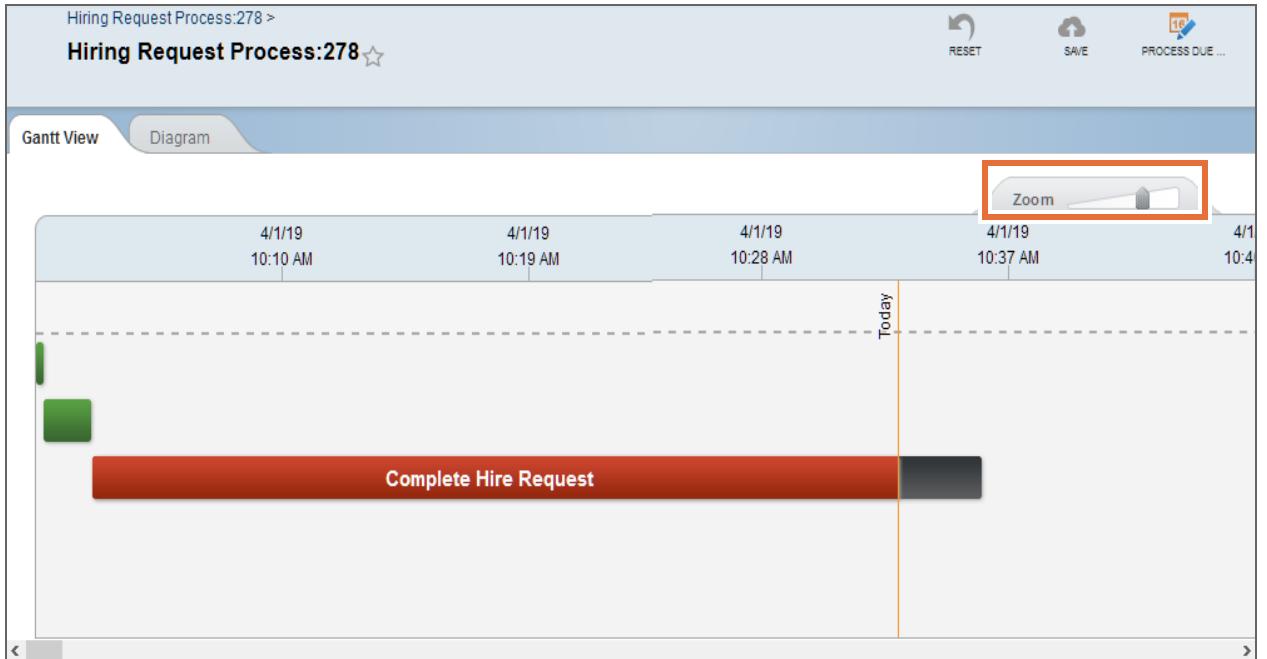
- ___ g. Roll the mouse over the flag. The setting that you configured earlier is displayed: **Very high**.



- ___ h. Click the red “x” to exit the window.
___ 2. Display the status of the Complete Hire Request activity in the Gantt chart.
___ a. In the Library click **Processes**.

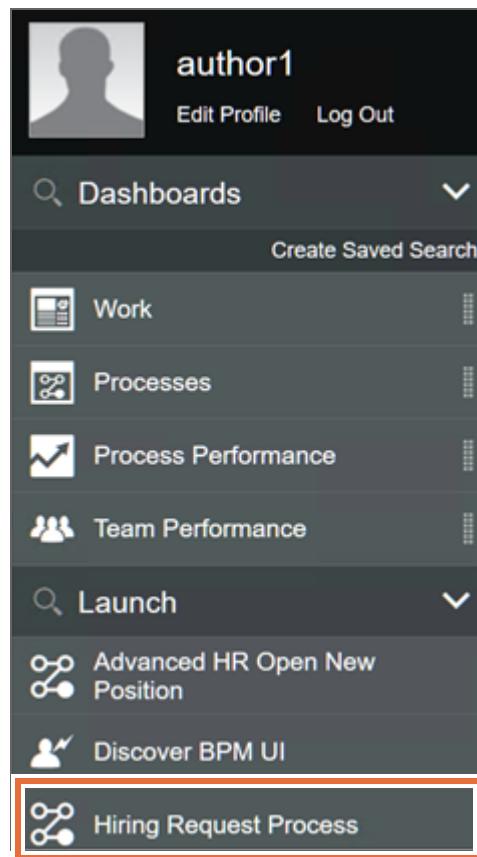


- b. Click the earlier process (the process with the smaller process instance ID).
- c. Click **Gantt Chart**.
- d. Use the Zoom control to zoom out as necessary to view all activities.

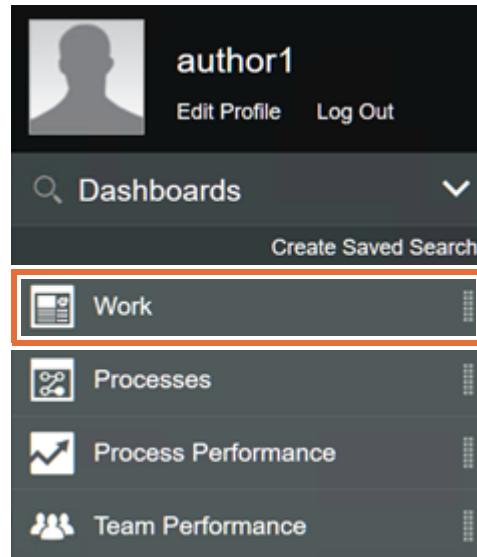


- e. The Complete Hire Request activity is highlighted in red, depicting it is past due. Click the **Complete Hire Request** activity to view more details.
 - f. Roll the mouse over the flag. The default setting you configured earlier is displayed: **Medium**.
 - g. Click the red “x” to exit the window.
3. Demonstrate the status of the activities on the Work page.

- __ h. Click the **Main Menu** icon to return to the Library. To display an activity on track, create an instance of the **Hiring Request Process** by clicking the link in the Launch section.



- __ i. Do not complete the task. Click the **Main Menu** icon to return to the Library. Click **Work** to return to the Work page.



- __ j. You now have three work items displayed.

- ___ k. Mouse over the priority and status icons to display their textual representation. For the purposes of the demonstration, there is no need to complete the activities. The project team agrees the priority and status functionality meets their needs. The next step is to create a toolkit, create a snapshot, and export the process application.

The screenshot shows the IBM Process Designer interface. On the left is a sidebar with links like Dashboards, Work, Processes, etc. The main area is titled 'Work' and shows a search bar and a list of work items. The first three items in the list are for a 'Hiring Request Process'. Each item has a title, a due date, and a status section. The status section includes a priority icon (yellow circle with a black dot) and a status icon (orange box with a document icon). The first two items have their priority and status icons highlighted with a red box.

The status of your work items might vary depending on the timing.



Reminder

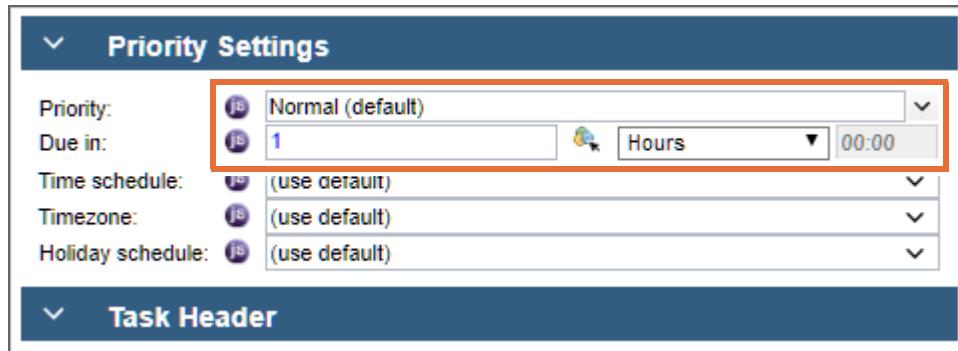
In this course, you did not implement all the coaches for the sake of brevity. In a real-world situation, one of the requirements to move to Playback 3 is that all the coaches are implemented in the process and all of them have the final presentation elements in place (CSS and JavaScript, for example).

Part 5: Reset the environment

When demonstrating mock scenarios, it is important to reset the environment back to the defaults before moving on.

- ___ 1. Reset the timeout value for the Complete Hire Request activity.
 - ___ a. Return to Process Designer and log in as author1.
 - ___ b. Open the **Hiring Request Process** unless it is already open.
 - ___ c. Click the **Complete Hire Request** activity.

- ___ d. In the **Priority Settings** section of the **Implementation** properties at the bottom, set the activity to be due in 1 hour.



- ___ 2. Reset the priority and timeout values for the Review Posting activity.
- ___ a. Click the **Review Posting** activity.
 - ___ b. In the Priority Settings section of the **Implementation** properties at the bottom, select **Normal** for the Priority setting.
 - ___ c. Set the activity to be due in 1 hour.
 - ___ d. Save your work.

Part 6: Create the Hiring Requisition Toolkit and take a snapshot

Now you have numerous assets, some of which might be useful to other developers. Create a toolkit from the assets you created as part of the process application HR Recruitment Processes.

A toolkit is a collection of assets that are shared between process apps or other toolkits during development. Attentive application of toolkits is an efficient method for managing the reusability and compartmentalization of project assets.

Starting in version 19.0.0.1, instead of duplicating effort creating and deploying two separate projects, you can use the enhanced Workflow Center user interface to put the process and case implementation in a workflow project that you can deploy as a single solution. You can also create, manage, and deploy your projects (such as process apps, case solutions, and toolkits) here.

- ___ 1. Create the Hiring Requisition Toolkit
- ___ a. Click **Return to Workflow Center**.

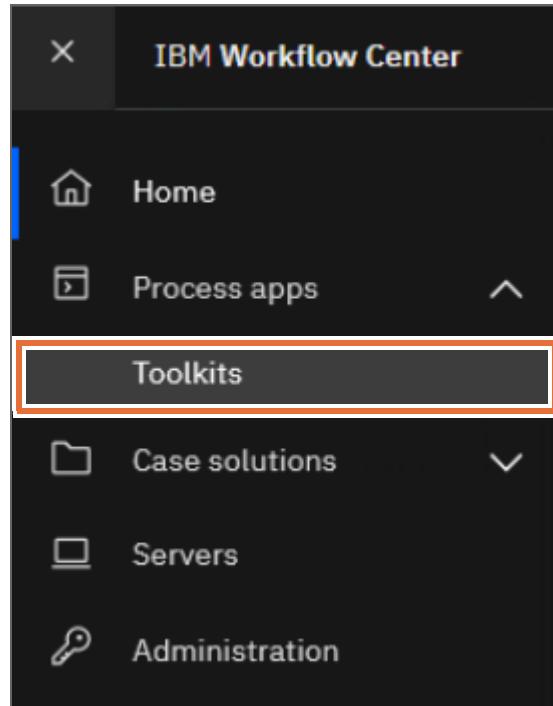


Hint

The **Return to Workflow Center** link returns you to the Workflow Center you originated from. If you opened the process application from the Classic Workflow Center, this link returns you there.

- ___ b. On the Workflow Center main page, click the navigation icon in the upper left and expand the **Process apps** section.

__ c. Click **Toolkits**



__ d. On the Toolkits page, click **Create**.

The screenshot shows the 'Toolkits (9)' page. At the top right, there is a blue 'Create' button with a '+' icon, which is highlighted with a red rectangular box. Below the header, there are search and filter options, and a message indicating 'Displaying 9/9 projects'. Three toolkit cards are listed: 'UI Toolkit', 'SAP Guided Workflow (deprecated)', and 'System Governance'. Each card has a 'Details →' link at the bottom right.

__ e. Enter the name for the toolkit: **Hiring Requisition Toolkit**

__ f. Verify that the Acronym is automatically created: **HRT**



Troubleshooting

If the Acronym control is not displayed, verify you are using the Advanced view mode.

- __ g. Clear the **Open in Process Designer** checkbox if it is checked.
- __ h. Select **Traditional** for the Target environment.
- __ i. Enter a brief description: This toolkit contains all common artifacts that are associated with the HR Recruitment Processes.



Hint

While completing the form, be sure not to click outside of it. Once the focus is taken away, the form is withdrawn. If this happens, click **Create** again to initiate the dialog.

- __ j. Click **Create**.

Create a toolkit

Name
Hiring Requisition Toolkit

Acronym ⓘ
HRT

Open in Designer

Target environment
Traditional

Description
Highlight your text to see formatting options.
This toolkit contains all common artifacts that are associated with the HR Recruitment Processes.

Cancel **Create**

- __ k. Verify that the Hiring Requisition Toolkit shows up as a tile in the **Toolkits** list.

The screenshot shows the 'Toolkits (10)' page in the IBM Workflow Center. There are three toolkit tiles displayed:

- Hiring Requisition Toolkit**: This toolkit contains all common artifacts that are associated with the HR Recruitment Processes. It has a wrench and screwdriver icon, a star rating, and a 'Details' button.
- UI Toolkit**: This toolkit has a wrench and screwdriver icon, a star rating, and a 'Details' button.
- SAP Guided Workflow (deprecated)**: This toolkit is for SAP Guided Workflow. It has a wrench and screwdriver icon, a star rating, and a 'Details' button.

- __ 2. Move the client-side human service into the toolkit.
- On the Workflow Center main page, click the navigation icon in the upper left and click **Process apps**.
 - Click **HR Recruitment Process**.
 - In the library, right-click the **User Interface > Hiring Form** client-side human service.
 - Click **Move Item to > Other Toolkit**.

The screenshot shows the 'HR Recruitment Process' library. On the left, there is a sidebar with icons for Processes, User interface, Services, Events, Teams, Data, and Performance. The 'User interface' section is selected. On the right, there is a list of items under 'Client-Side Human Service'. One item, 'Hiring Form', has a context menu open. The menu includes options like Open, Favorite, Tags, Move item to, Open snapshot..., Revert..., Copy to tip, Conv item to, Duplicate, Rename, and Delete. The 'Move item to' option is highlighted with an orange border, and the 'Other Toolkit...' option within it is also highlighted with an orange border.

- __ e. Select **Hiring Requisition Toolkit**.



- __ f. Keep the defaults and click **MOVE**. The order of the dependent artifacts might be different than in the screen capture.

Move Artifacts to Toolkit: Hiring Requisition Toolkit

Select artifacts to move:
Selected artifacts will be moved.

Hiring Form

Select dependent artifacts to move:
Broken references might occur if you do not move dependent artifacts.

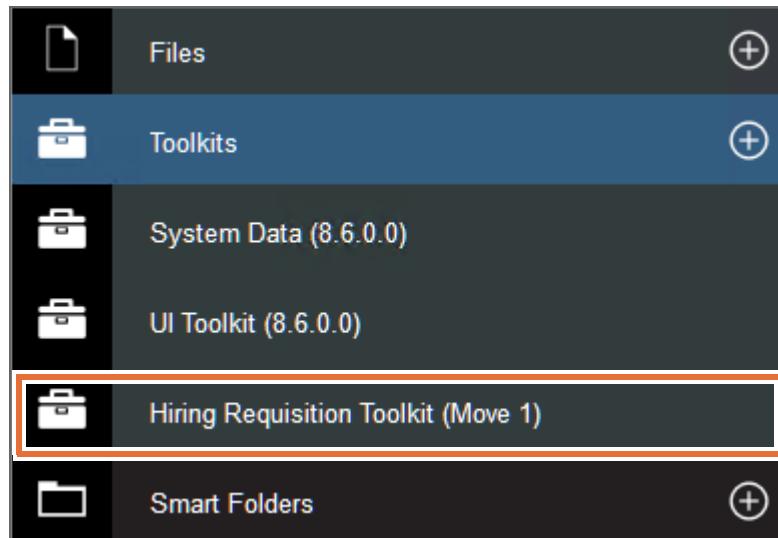
HiringRequisition
 CompensationDetails
 DepartmentDetails
 Position
 RecruitingDetails

Artifacts with dependencies on selected artifacts:
Selected items will be moved. Items that are not moved might cause broken references.

Hiring Request Process
 Approve Hire Request

MOVE **CANCEL**

- ___ g. All the dependent assets are also moved into the toolkit to include the variables that the client-side human service depends on. A dependency on the toolkit is automatically created inside the process application. In the Library section, expand **Toolkits** to see the Hiring Requisition Toolkit.

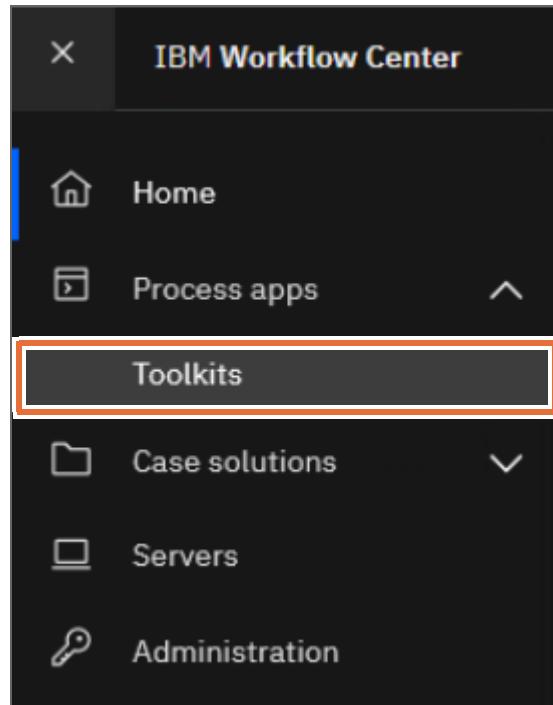


- ___ h. Return to the **Workflow Center**.



- ___ 3. Take a snapshot.

- ___ a. On the Workflow Center main page, click the navigation icon in the upper left and expand the **Process apps** section.
- ___ b. Click **Toolkits**



- __ c. Click **Details** on the Hiring Requisition Toolkit tile.

The screenshot shows the 'Toolkits (10)' page in the IBM Workflow Center. At the top, there are search and filter options ('Recently updated', 'All'), and buttons for 'Import' and 'Create'. Below this, a message says 'Displaying 10/10 projects'. Three toolkit cards are listed:

- Hiring Requisition Toolkit**: This toolkit contains all common artifacts that are associated with the HR Recruitment Processes. The 'Details →' button is highlighted with a red box.
- UI Toolkit**
- SAP Guided Workflow (deprecated)**: Toolkit of SAP Guided Workflow. The 'Details →' button is also present here.

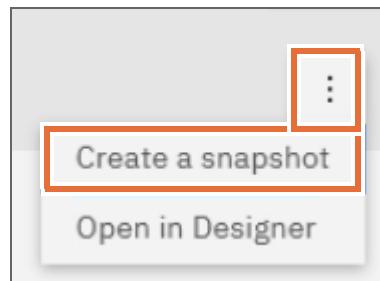
- __ d. Select the **Snapshots** tab.

- __ e. You notice that a snapshot has already been created. This was done automatically when you moved the assets to the Toolkit.

The screenshot shows the 'Schemas (1)' page with the 'Snapshots' tab selected. The tabs available are Overview, Snapshots, Permissions, History, and Comments. Below the tabs, it says 'Snapshots (1)'. A search bar and a dropdown menu are visible. The table below lists the snapshots:

Name	Status	Created	Last updated	Target environment
→ Current	New	author1 9/22/2020, 12:51:50 PM	author1 9/22/2020, 12:51:50 PM	Traditional
→ Move 1 (M1)	New	author1 9/22/2020, 3:35:42 PM	author1 9/22/2020, 3:35:42 PM	Traditional

- ___ f. Open the list of options by clicking the **three dots** on the right of the **Current snapshot** tile and select **Create a snapshot**.



Recall that a snapshot captures the state of the library items within a process application or toolkit at a specific point in time. Snapshots usually represent a milestone or are used for playbacks or for installation. Since Playback 1 is now complete, it is a good time to take a snapshot.

- ___ g. Name the snapshot **Baseline** and add a description: This is the baseline snapshot of the Hiring Requisition Toolkit. Then, click **Create**.

Snapshot name
Baseline

Description
Highlight your text to see formatting options.
This toolkit contains all common artifacts that are associated with the HR Recruitment Processes.

Cancel Create

- ___ h. The snapshot **Baseline** is created.

Snapshots (2) ⓘ	
Name ⓘ	Status
→ Current	
→ Baseline (B)	New
Move 1 (M1)	New

- ___ 4. Update the toolkit version on the process application.

Design toolkit hierarchies to avoid mutual dependencies by factoring out common content into its own toolkit so that the other toolkits can refer to it independently.

If you update the dependency on a child toolkit, you need to take a new snapshot of the parent toolkit (or process application) for the change to be effective for whoever uses that toolkit. This can lead to a situation with mutually dependent toolkits.

- ___ a. Click the navigation icon in the upper left and select **Process apps**.

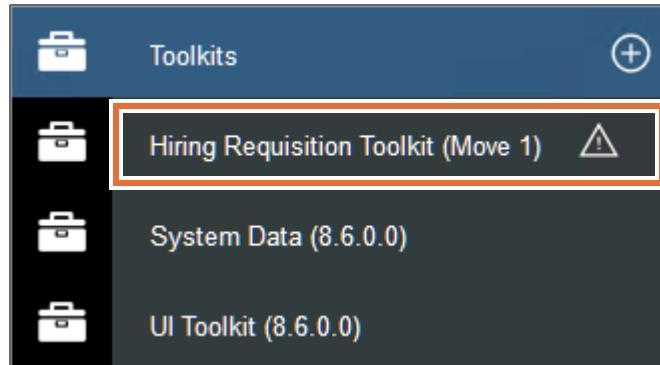


Troubleshooting

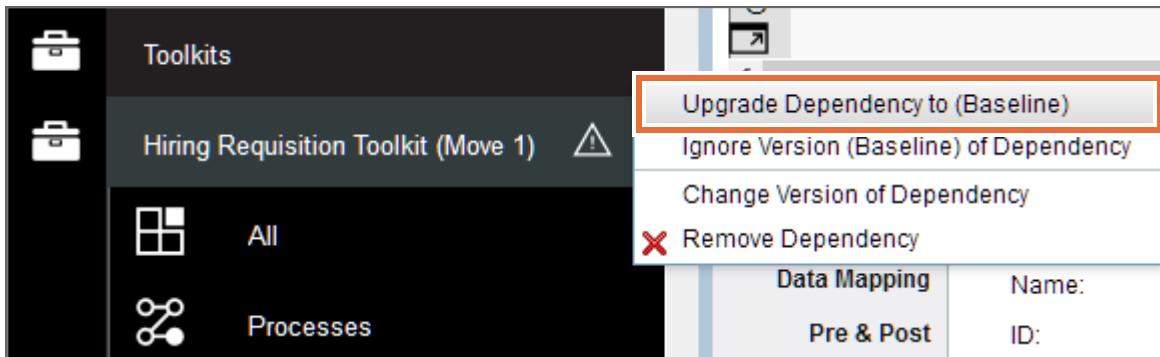
If the Process Apps page displays the Toolkits, wait approximately a minute for the page to refresh.

- ___ b. Open the **HR Recruitment Process** process application.

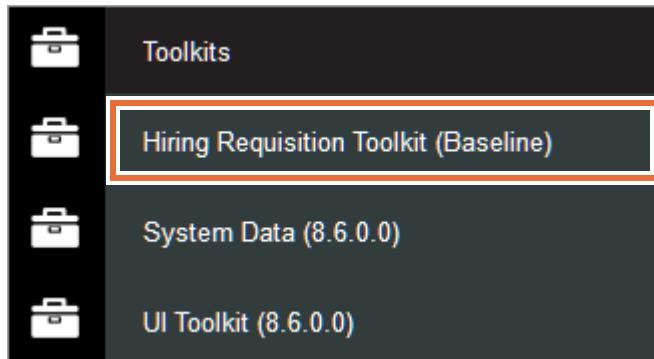
- ___ c. When you return to the process application, Expand the **Toolkits** category in the Library section. The Hiring Request Toolkit shows a warning because the process application is not pointing to the most recent snapshotted version.



- ___ d. Click the arrow next to the Toolkit and select **Upgrade Dependency to (Baseline)**.



- ___ e. In the Toolkits category, verify that the Hiring Requisition Toolkit (Baseline) is now linked to the HR Recruitment Processes process application and that the warning is gone.



Information

Because you created a toolkit and created a dependency, each solution file you import from this point in the training will include the Hiring Requisition Toolkit.

You have completed this exercise.

In this exercise, you completed Playback 1. You conducted a playback session that demonstrates two unique process scenarios. You used the Process Portal to demonstrate the playback from the

user point-of-view. After completing Playback 1, you created a snapshot for the toolkit and process application and then exported the process application.

This completes Playback 1. The process, data, and coaches have now all been approved to move on to the next playback. Once all of these components are verified, concentration can be paid to integration with other systems, which is covered in the next playback.

End of exercise

Exercise 8. Playback 2: Integrations

Estimated time

01:30

Overview

This exercise covers how to create implementation assets to support Playback 2: integrations.

Objectives

After completing this exercise, you should be able to:

- Create a decision service
- Create and configure a UCA
- Start a process with a message start event
- Use tagging to organize assets
- Query a database to obtain information and populate a list variable
- Create environment variables (ENVs) and exposed process variables (EPVs)
- Change a text control to a single select control

Introduction

In Playback 2: Integrations, you implement the process interactions and integrations so the process can have all the functions that are needed to complete any process activity. Not all the functions are developed within the process application when it comes to user interactions. Some of the business data can be found, for instance, in systems within an organization. So it is important to be able to retrieve the data, and that is why integrations play a vital role in having a full and robust process application.

Other process interactions involve events within the process model. In this stage of Playback 2: Integrations, you implement those system activities that set business data and process flow data with rules and automations. These intermediate events or start events might need unique event handlers, such as listeners for messages that trigger an event. In Playback 2, you implement, test, and finalize all remaining process application interactions.

Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

Index

[Part 1: Create a decision service](#)

[Part 2: Implement a message start event](#)

[Part 3: Apply asset tags](#)

[Part 4: Create a service to query a database and populate a list](#)

[Part 5: Change an input to a single select on a coach](#)

Exercise instructions

Part 1: Create a decision service

The Hire Request business process is a candidate to include a decision service in the process flow. Decision services allow the business process to make routine decisions that are based on real business rules to speed up the process and eliminate user error.

If the salary is not compliant, the criteria that they must use is to route hiring requisitions for administrator override. The salary compliance depends on the job level. Different job levels have different compliance ceilings. In the as-is process, a user decides. In the to-be process, a user was going to decide, but now they want the system to decide based on business rules.

Your task is to implement that decision service in a process. You created the process model during Playbacks 0 and 1, and in Playback 2, you implement all the services.

This exercise is long, so a break is included in the middle of the instructions. The instructions include some swapping back and forth and moving around different assets in the library. The sequence is intentionally set to simulate the steps that a developer takes when creating these assets, and reflects real-world development efforts.

- ___ 1. Create a decision service.
 - ___ a. Open the **HR Recruitment Process** process application.
 - ___ b. In the library, click the (+) plus sign next to Services. Click **Service Flow** to create a service.
 - ___ c. Name the service **Salary Compliance Service** and click **FINISH**.

New Service Flow

A service flow lets you build, test, and run a set of services. [Learn More](#)

Name:

Use as a team service:

FINISH **CANCEL**

- ___ 2. Add a Business Action Language rule.
- ___ a. The service is opened to the Diagram tab. Expand the top activity and drag a **Decision** activity from the service palette onto the sequence flow between the Start and End events.



A **green plus (+)** is displayed when you are in an area that the item can be dropped. A **red circle with an (x)** appears if the mouse is hovering in an area where it cannot be dropped. Make sure the sequence line is highlighted and the mouse pointer turns to a **green plus (+)** sign.



- ___ b. The activity is placed inside the sequence flow.
- ___ c. Name the Decision: Check Compliance





Reminder

You can rename artifacts two ways: either through the **General > Common** properties section, or by double-clicking the artifact and renaming it on the canvas.

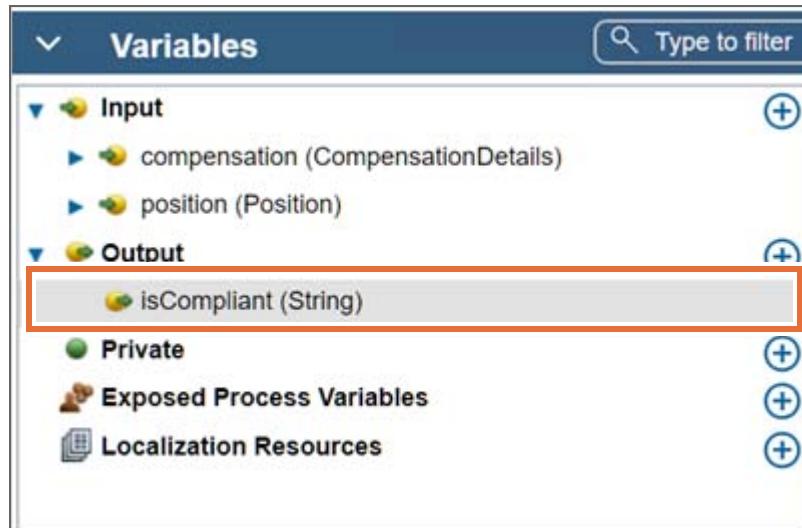
- ___ 3. Add variables to the service.
 - ___ a. Click the **Variables** tab.
 - ___ b. Add an input and name the variable: compensation

- ___ c. Click **Select** next to the Variable type. To filter the variable type list, type: Com and then click the variable type CompensationDetails.

- ___ d. Add a second input variable named `position` of type **Position**.



- ___ e. Add an output variable: `isCompliant (String)`.



- ___ f. Save your changes.

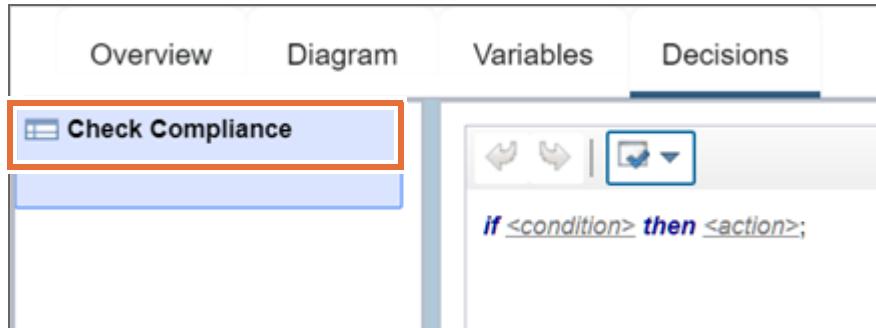
4. Build the condition.

First, you build the condition, then you build the action to be taken when the condition is met.

- ___ a. Click the **Decisions** tab.



- __ b. Click the **Check Compliance** BAL rule on the left.



A default rule is added to the BAL rule editor. The condition and action options are editable. If you click either option, an editor is displayed to assist you with writing the rule. As is the case with the JavaScript editor, you can also use content assist (Ctrl+Space).

The BAL rule is a simple set of if-then statements. In this example, if the salary is over or under a threshold, it is not compliant. Otherwise, it is compliant. The threshold levels are:

- Associate. Minimum salary: 40,000, maximum: 60,000
- Manager. Minimum salary: 50,000, maximum: 75,000
- Director. Minimum salary: 70,000, maximum: 95,000
- President. Minimum salary: 100,000, maximum: 150,000



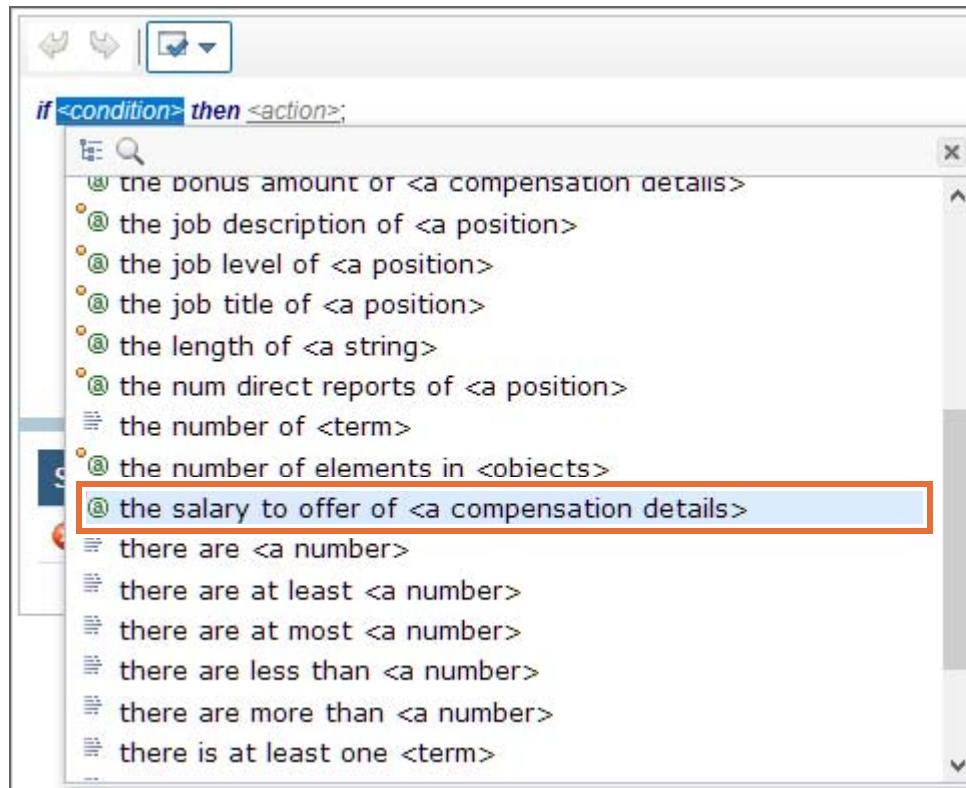
Troubleshooting

The lab environment might occasionally cause unexpected behavior when using content assist. If the content assist window ever disappears, press the spacebar to see the content assist window display again.

Alternatively, if you are having difficulty with the content assist, you can copy and paste the rules text that is provided in the file `BAL Rules.txt` at the location: `C:\labfiles\Support files\Ex8`

- __ c. To build the condition, click the `<condition>` statement for your rule, and select the following rule piece:

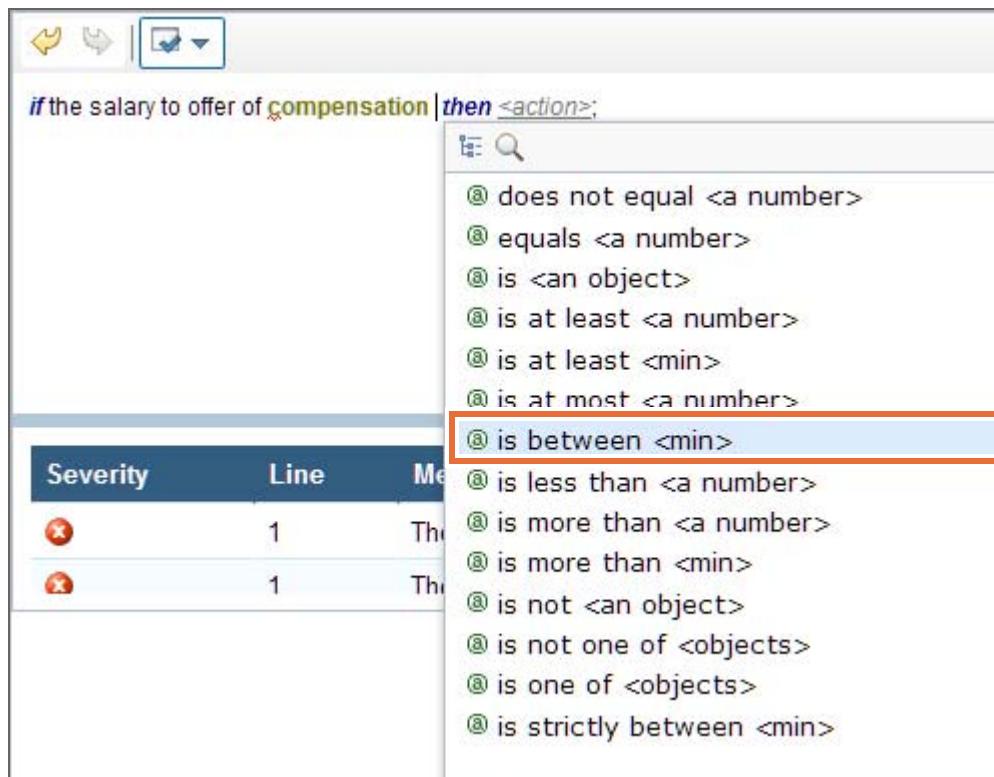
the salary to offer of <a compensation details>



- __ d. Add a space and use the content assist (Ctrl+Space) to add compensation



- __ e. Use content assist (Ctrl+Space) to add the following: is between <min>



- __ f. Use content assist to add: <number>. Immediately after being added, the number turns to 0. Enter 40000 then add a space.
- __ g. Use content assist to add and <max>
- __ h. Enter 60000
- __ i. Add a space and use content assist to add and <condition>
- __ j. Use content assist to add the job level of <a position>
- __ k. Use content assist to add position
- __ l. Use content assist to add contains <a string>
- __ m. Enter "Associate"
- __ n. Save your work.

___ 5. Build the action.

___ a. Click the <action> statement of your rule, and select the following rule pieces:

```
set <variable>
isCompliant
to <variable value>
<string>
"1";
<space>
else
set <variable>
isCompliant
to <variable value>
<string>
"0";
```



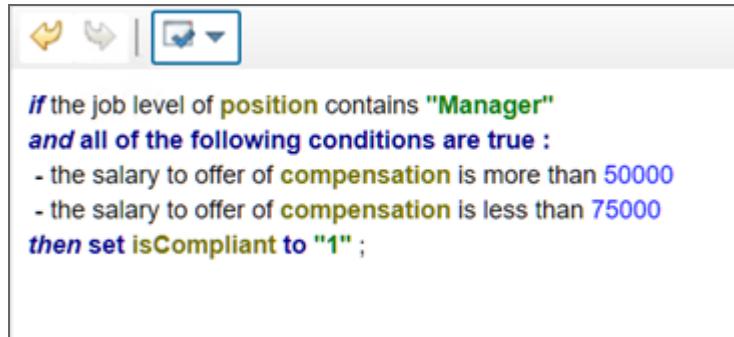
The BAL rule reads much like a natural sentence. You assign a value in the first rule, and the other rules below it evaluate. If one of the rules evaluates to true, then it overwrites the variable.

___ b. Click the **Add action rule (+)** symbol to create a second rule.



- ___ c. Build the next rule by using the previous method to flag the salary as non-compliant when it does not fall within the same range. Alternatively, if you are having difficulty with the content assist, you can copy and paste the rules text that is provided in the file BAL Rules.txt at the location: C:\labfiles\Support files\Ex8. Build the rule to read:

if the job level of position contains "Manager"
and all of the following conditions are true:
- the salary to offer of compensation is more than 50000
- the salary to offer of compensation is less than 75000
then set isCompliant to "1";



- ___ d. Click the **Add action rule (+)** symbol to create the third rule.
___ e. Copy and paste the prior rule and change the variables as required.

if the job level of position contains "Director"
and all of the following conditions are true:
- the salary to offer of compensation is more than 70000
- the salary to offer of compensation is less than 95000
then set isCompliant to "1";

- ___ f. Click the **Add action rule (+)** symbol to create the final rule.
___ g. Copy and paste the prior rule and change the variables as required.

if the job level of position contains "President"
and all of the following conditions are true:
- the salary to offer of compensation is more than 100000
- the salary to offer of compensation is less than 150000
then set isCompliant to "1";



Reminder

To avoid any errors, the rules text is provided in the file **BAL Rules.txt** at the location:
C:\labfiles\Support files\Ex8

- __ h. Change the position of the different rules to make the first rule you created the top rule, and each subsequent rule underneath the prior one by using the arrows on the right of each rule. You can start by scrolling to the rule for Associate and click the arrow to bring it to the top.



- __ i. Verify that all the four rules are created and are placed in the correct order.

```

if the salary to offer of compensation
is between 40000 and 60000 and
the job level of position contains "Associate"
then set isCompliant to "1";
else set isCompliant to "0";

if the job level of position contains "Manager"
and all of the following conditions are true:
- the salary to offer of compensation is more than 50000
- the salary to offer of compensation is less than 75000
then set isCompliant to "1";

if the job level of position contains "Director"
and all of the following conditions are true:
- the salary to offer of compensation is more than 70000
- the salary to offer of compensation is less than 95000
then set isCompliant to "1";

if the job level of position contains "President"
and all of the following conditions are true:
- the salary to offer of compensation is more than 100000
- the salary to offer of compensation is less than 150000
then set isCompliant to "1";

```

- ___ j. Save your work.
- ___ 6. Debug the service: **jobLevel** equals Associate.

To make sure that the rules are working as expected, debug your service by providing default values to the variables.

- ___ a. Switch to the **Variables** tab.
- ___ b. Click the input variable **compensation (CompensationDetails)**. Select the **Has Default** checkbox under the Default Value section on the bottom right.

The screenshot shows the 'Variables' tab in a service editor. On the left, there's a tree view of variables: Input (compensation, position), Output (isCompliant), Private, Exposed Process Variables, and Localization Resources. The 'compensation (CompensationDetails)' variable is selected. On the right, the 'Details' panel shows the variable's name as 'compensation' and its type as 'CompensationDetails'. Below that, the 'Default Value' section is expanded, showing a checked checkbox labeled 'Has default'. A code editor below contains the following JavaScript-like code:

```

1 var autoObject = new tw.object.toolkit.HRT.CompensationDetails();
2 autoObject.salaryToOffer = 45000.0;
3 autoObject.bonusAmount = 0.0;
4 autoObject

```

- ___ c. Change the value for the variable **salaryToOffer** to 45000.0

The screenshot shows the 'Default Value' code editor. It displays the same JavaScript code as the previous screenshot, but the line `autoObject.salaryToOffer = 45000.0;` is highlighted with a red rectangular box.

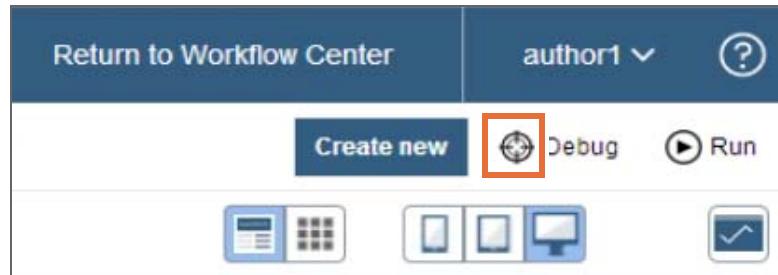
- ___ d. Click the input variable **position (Position)**. Select **Has Default** under the Default Value section on the bottom right. Enter "Associate" for the variable **jobLevel**:

```

var autoObject = new tw.object.toolkit.HRT.Position();
autoObject.jobTitle = "";
autoObject.jobLevel = "";
autoObject.department = "";
autoObject

```

- ___ e. Click the **Debug** button at the top.



Hint

Artifacts are automatically saved when you debug or run a service.

- ___ f. The view changes from the designer view to the inspector view.

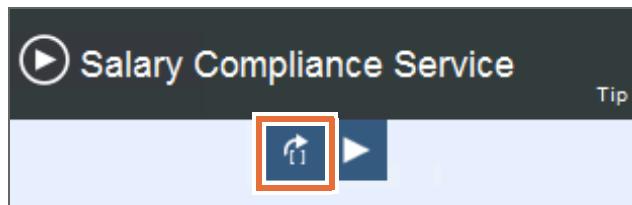


- ___ g. The debug window opens on the right to show information about the service. Expand the **Data** section on the bottom and verify that the default variables are assigned correctly.

The screenshot shows the 'Salary Compliance Service' debug window. At the top, there are navigation icons: a back arrow, a forward arrow, and a refresh icon. Below these are sections for 'Details' and 'Locations'. The 'Data' section is expanded and highlighted with an orange border. It contains several variable assignments:

- `compensation(CompensationDetails)`
- `salaryToOffer(Decimal) 45,000` (highlighted with an orange border)
- `bonusAmount(Decimal) 0`
- `position(Position)`
- `numDirectReports(Integer) 0`
- `jobTitle(String) <Empty>`
- `jobDescription(String) <Empty>`
- `jobLevel(String) Associate` (highlighted with an orange border)
- `isCompliant(String) <Empty>`

- ___ h. Click the **Step Over** icon on the top to test the first decision rule with the value for variable **salaryToOffer** as `45000` and the value for variable **jobLevel** as `Associate`.



- ___ i. The service runs the BAL rule, the service completes, and the status changes to Finished. The correct response with this set of inputs is that the request is compliant. Verify that you get an `isCompliant` variable value of 1 because for the Associate position the salary is in the range 40000 – 60000 and you tested with the value 45000, which makes this request compliant.

Salary Compliance Service

No actions were found.

Details

Instance ID: 53
Process Ann: HR Recruitment Processes

Status:	Finished
---------	----------

Locations

No locations were found.

Data

- compensation(CompensationDetails)
 - salaryToOffer(Decimal) 45,000
 - bonusAmount(Decimal) 0
- position(Position)
 - numDirectReports(Integer) 0
 - jobTitle(String) <Empty>
 - jobDescription(String) <Empty>
 - jobLevel(String) Associate

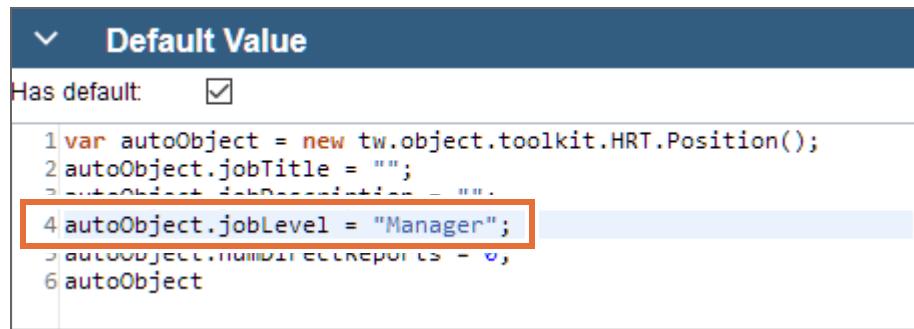
isCompliant(String)	1
---------------------	---

- ___ 7. Debug the service: **jobLevel** equals Manager.
___ a. Click the **Designer** tab to return to the Designer view.



- ___ b. On the **Variables** tab, select the **position (Position)** input variable

- ___ c. Change the value of the **jobLevel** variable to "Manager" and save the service. Switch to the Diagram tab and click the **Debug** button on the top.

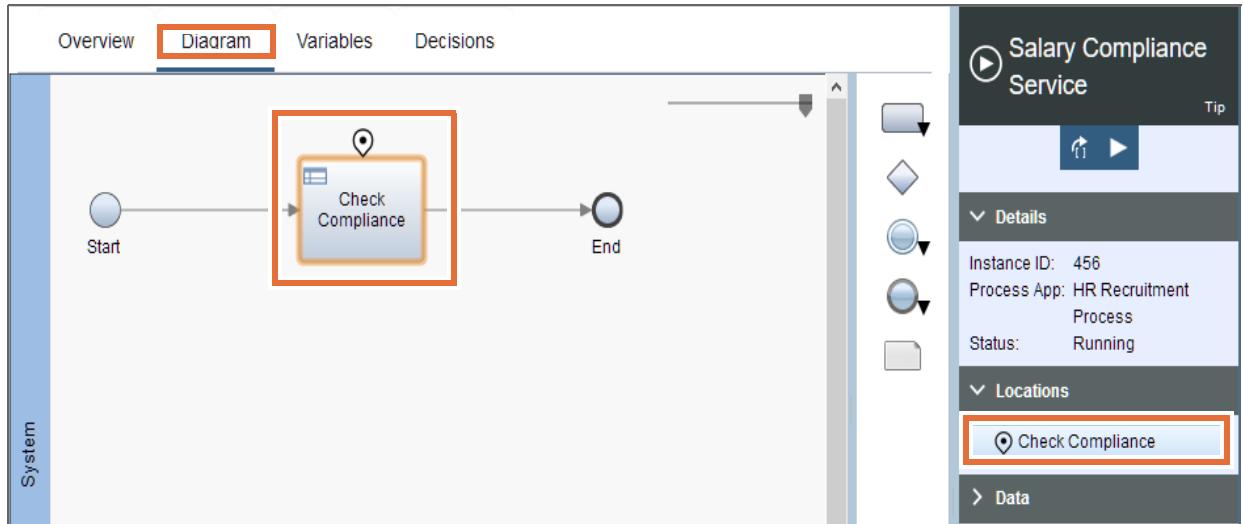


```

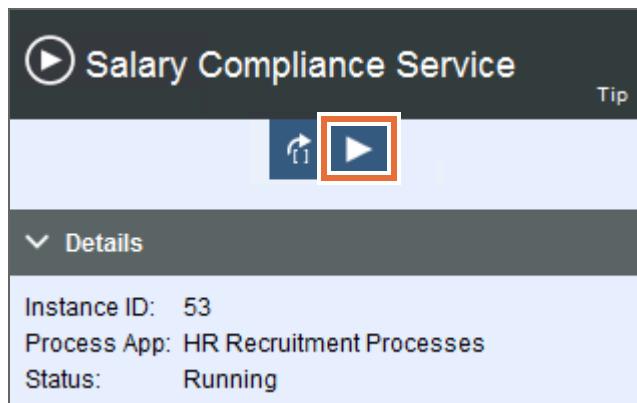
var autoObject = new tw.object.toolkit.HRT.Position();
autoObject.jobTitle = "";
...
4 autoObject.jobLevel = "Manager";
autoObject.numberOfReports = 0;
autoObject

```

- ___ d. The information window opens on the right. The token on the Check Compliance service step shows the next step in the service to be run. The token matches the step that is shown in the **Locations** section in the information window.



- ___ e. Expand the **Data** section and click the **Run** button on the top to run the service to completion.



The Step Over button runs a single step in the service where the token is located. The Run button runs the service until a coach is displayed or until the service completes.

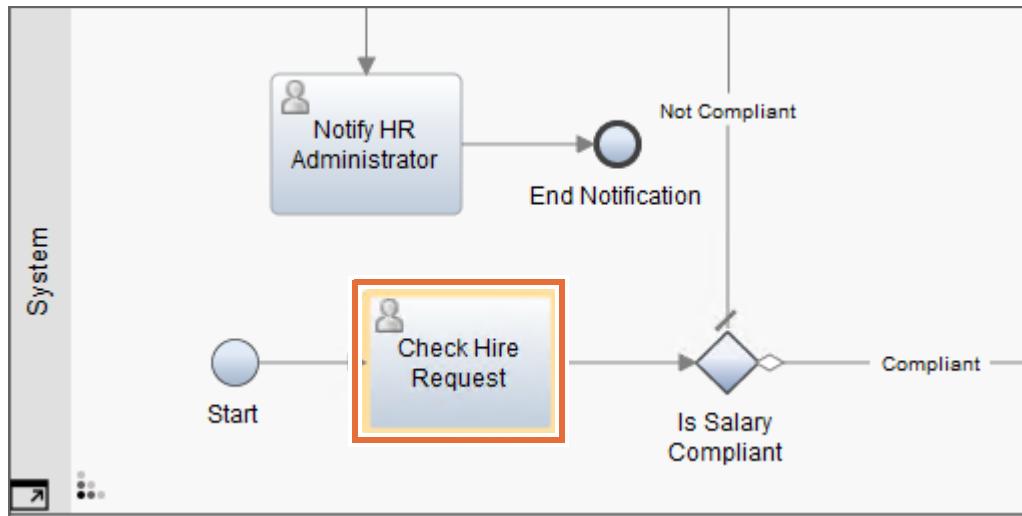
- ___ f. The service request is not compliant. Expand the business objects and verify that you get an `isCompliant` variable value of 0. This is because the salary for the Manager position is in the range 50000 – 75000 and you tested with salary 45000, which makes this request non-compliant.

The screenshot shows the interface of the Salary Compliance Service. At the top, there is a header with a checkmark icon and the text "Salary Compliance Service". Below the header, a "Tip" section states "No actions were found." The main content area has three sections: "Details", "Locations", and "Data". The "Data" section contains two objects: "compensation(CompensationDetails)" and "position(Position)". The "isCompliant(String)" variable under "position(Position)" is highlighted with a red rectangle. The variable's value is shown as 0.

Object	Value
compensation(CompensationDetails)	salaryToOffer(Decimal) 45,000 bonusAmount(Decimal) 0
position(Position)	numDirectReports(Integer) 0 jobTitle(String) <Empty> jobDescription(String) <Empty> jobLevel(String) Manager isCompliant(String) 0

- ___ g. Return to the **Designer** view of Process Designer.
- ___ h. Click the **Variables** tab and clear the **Has Default** checkbox under the **Default Value** section for the input variables **compensation (CompensationDetails)** and **position (Position)**.
- ___ i. Save your changes.
- ___ 8. Add the rule to the process.
- ___ a. Open the **Approve Hire Request** process.

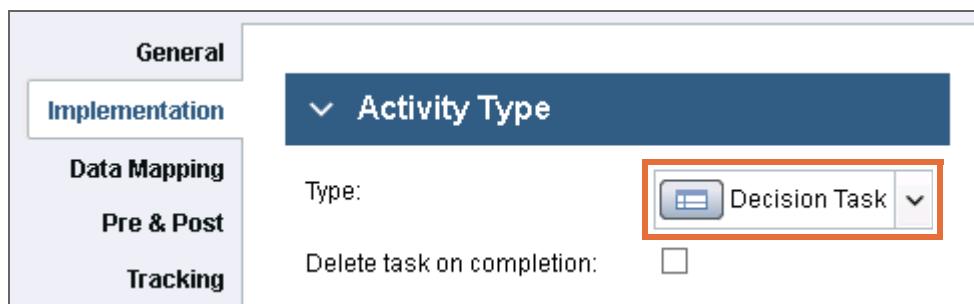
- __ b. Select the **Check Hire Request** activity in the **System** lane.



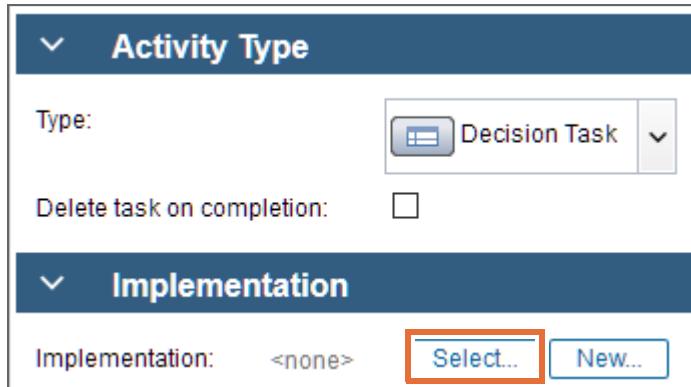
- __ c. Click the **Implementation** properties menu.



- __ d. Change the activity type to **Decision Task**.



- __ e. Click **Select** in the **Implementation** section.

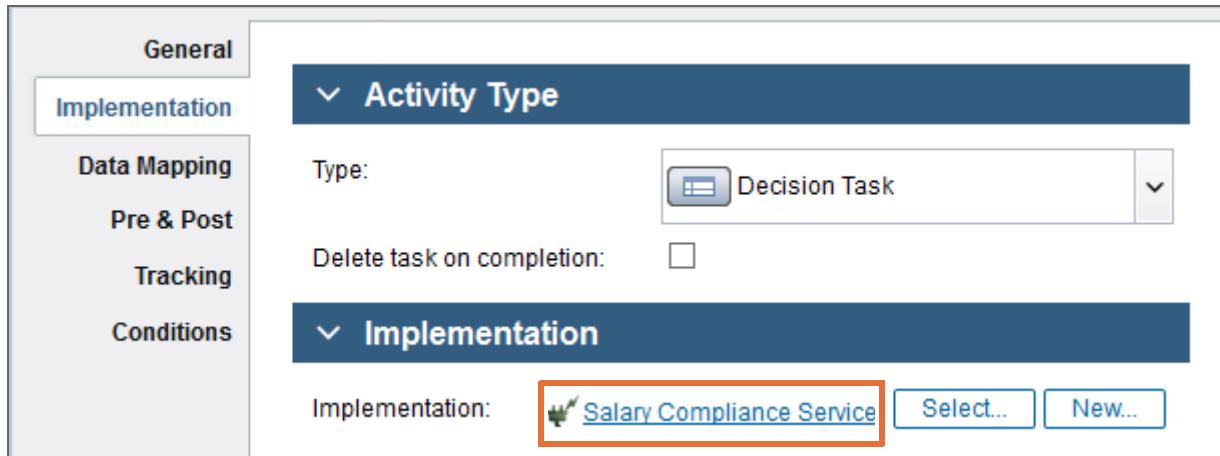


Information

If you want to run an automated service that does not require routing, you must select the **Delete task on completion** checkbox. This option deletes the task from the system and the process inspector after the task completes. When you select this checkbox, the Process Server does not retain audit data for the task. By default, this checkbox is not selected.

Deleting the task on completion from the task table in the database provides some performance advantages. However, in this case you do not select this option because you want to avoid losing some auditing data, including the details of when that system task ran and how long it took.

- __ f. Select **Salary Compliance Service**.



- __ 9. Map the variables for the **Check Hire Request** activity.

- __ a. Open the **Data Mapping** properties menu.
- __ b. Enter `tw.local.requisitionDetails.compensationDetails` for the first input to map it with **compensation**.

- ___ c. Enter `tw.local.requisitionDetails.position` for the second input to map it with **position**.

Input Mapping

<code>tw.local.requisitionDetails.compensationDetails</code>	compensation (Compensation...)
<code>tw.local.requisitionDetails.position</code>	position (Position)

- ___ d. Enter `tw.local.isCompliant` for the output to map it with **isCompliant**.

Output Mapping

isCompliant (String)	<code>tw.local.isCompliant</code>
--------------------------------------	-----------------------------------

Notice that you do not need to send the full `requisitionDetails` object into every activity. If a linked process or an activity needs only a small amount of data from the parent object, send in what is necessary. This approach helps the performance of your processes at run time. If you know that variables hold the same data, it also helps to name those variables the same across your processes because it helps with maintenance and troubleshooting.

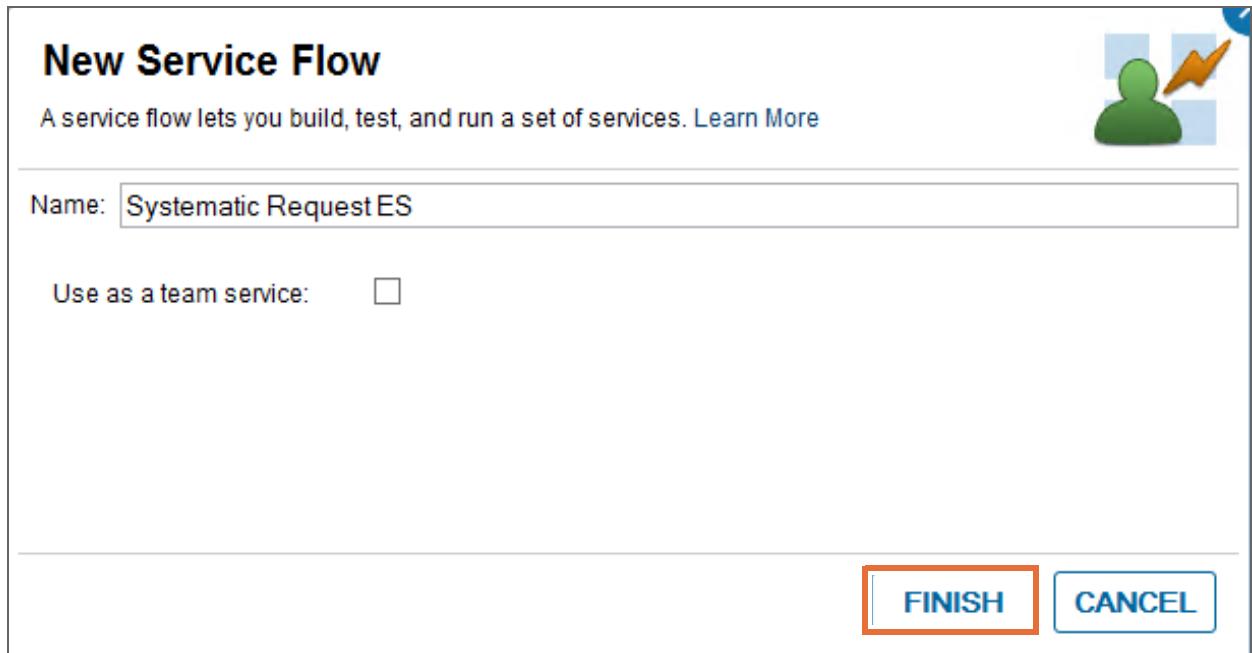
- ___ e. Save your work.

Part 2: Implement a message start event

A business requirement has been added specifying the process needs to automatically run every month. Now it is time to implement the process so an instance of the process can be initiated monthly.

- ___ 1. Create a service to enable the undercover agent (UCA).
 - ___ a. In the library, click the **(+)** plus sign next to the **Services** menu.
 - ___ b. Click **Service Flow**.
 - ___ c. Name the service: Systematic Request ES

__ d. Click **FINISH**.



New Service Flow

A service flow lets you build, test, and run a set of services. [Learn More](#)

Name: Systematic Request ES

Use as a team service:

FINISH **CANCEL**

__ e. Click the **Variables** tab. Continuing from Part 1 of this exercise, create a requisitionDetails (HiringRequisition) input and output variable.



Variables

- Input**
 - requisitionDetails (HiringRequisition)
- Output**
 - requisitionDetails (HiringRequisition)
- Private
- Exposed Process Variables
- Localization Resources

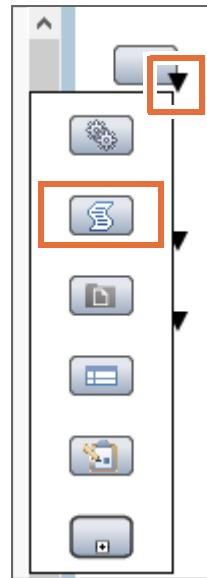


Important

Make sure that any single variable that is passed in and out of a service is spelled the exact same for both fields. Remember, variable names ARE case-sensitive. If you have any differences in the variable names, then the system considers them two different variables. This error frustrates many developers when troubleshooting why your variable is not passing out the expected data. Copying and pasting variable names ensures that your variables are the exact same, and reduces problems that result from typographical errors.

__ f. Save your work.

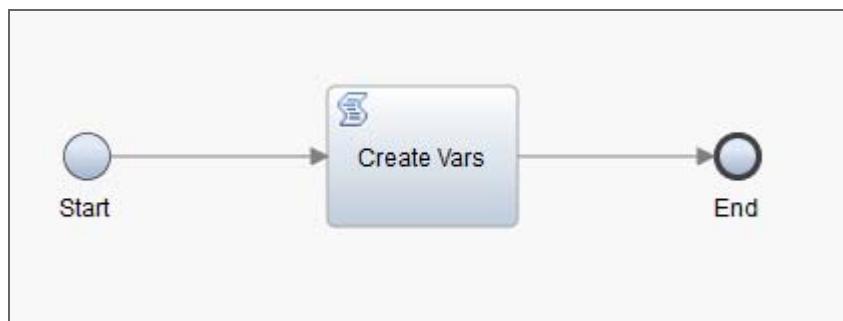
- ___ 2. Create a server script to initialize the variables and pass it out of the service.
- ___ a. Return to the **Diagram** tab. Expand the **Activity** selection and drag a server script from the palette onto the canvas.



A **green plus (+)** is displayed when you are in an area that the item can be dropped. A **red circle with an (x)** appears if the mouse is hovering in an area where it cannot be dropped. Make sure the sequence line is highlighted and the mouse pointer turns to a **green plus (+)** sign.

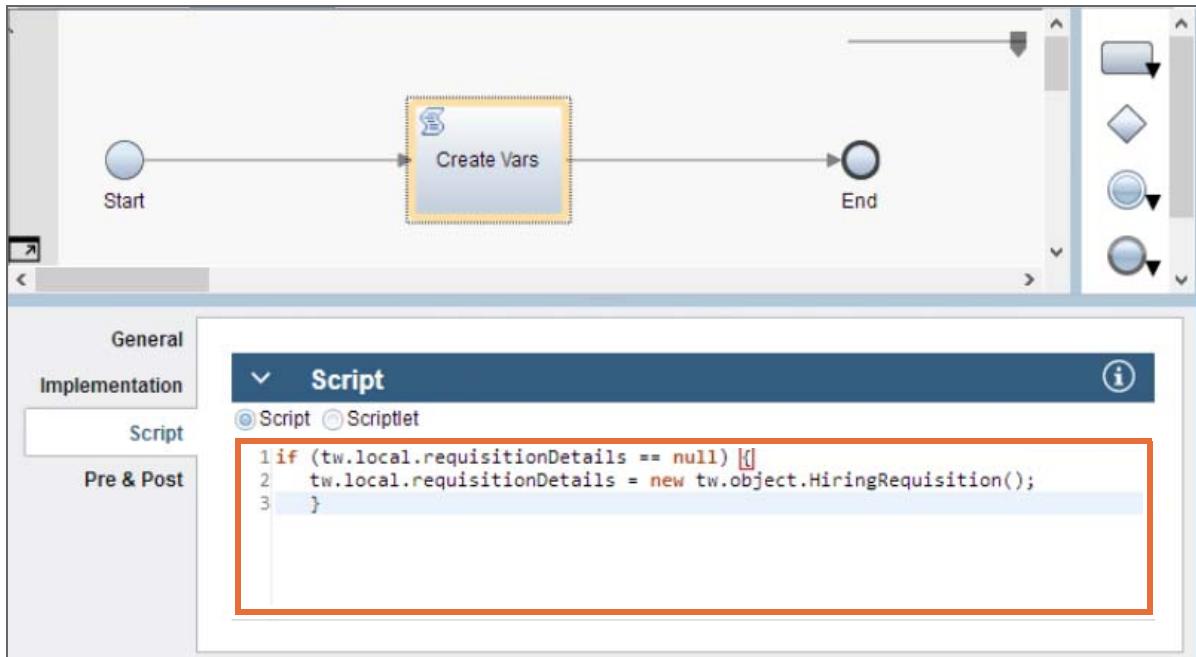


- ___ b. The activity is placed inside the sequence flow.
- ___ c. Name the server script: Create Vars

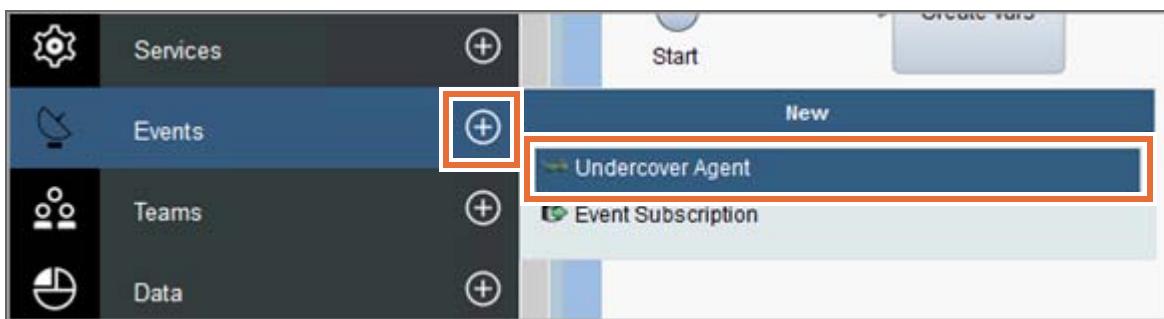


- ___ d. Create a server script to initialize the variables and pass it out of the service. In the **Script** properties menu, copy the script from the `Script1.txt` file that is located in the `C:\labfiles\Support files\Ex8` folder. Optionally, you can enter the following script in the box:

```
if (tw.local.requisitionDetails == null) {
    tw.local.requisitionDetails = new tw.object.HiringRequisition();
}
```



- ___ e. Save the service.
- ___ 3. Create and enable an Undercover Agent.
- ___ a. Click the (+) plus sign next to **Events**, and click the **Undercover Agent** option.



- ___ b. Name the UCA: Systematic Request

Make sure that the **Time Elapsed** schedule type is selected. Click **FINISH** to complete the creation of the UCA.

New Undercover Agent

Undercover agents (UCAs) are started by an event and invoke a Process.
UCAs pass data to the Process through an associated service or variable.

Name: Systematic Request

Schedule Type: Time Elapsed

FINISH **CANCEL**

- ___ c. In the Details section of the UCA settings page on the right, click **Select** next to Attached Service. Select the **Systematic Request ES** service. Verify that the **Enabled** checkbox is selected for the new undercover agent.

Details

Queue Name: Async Queue

Attached Service: **Systematic Request ES** **Select...** **New...**

Enabled:

- ___ d. Create the schedule according to the following script:

Every month (**all months**) on the **first Saturday** at **Midnight on the hour**.

Select all the months, then the **First**, **Saturday**, **Midnight**, and **On the hour** options.

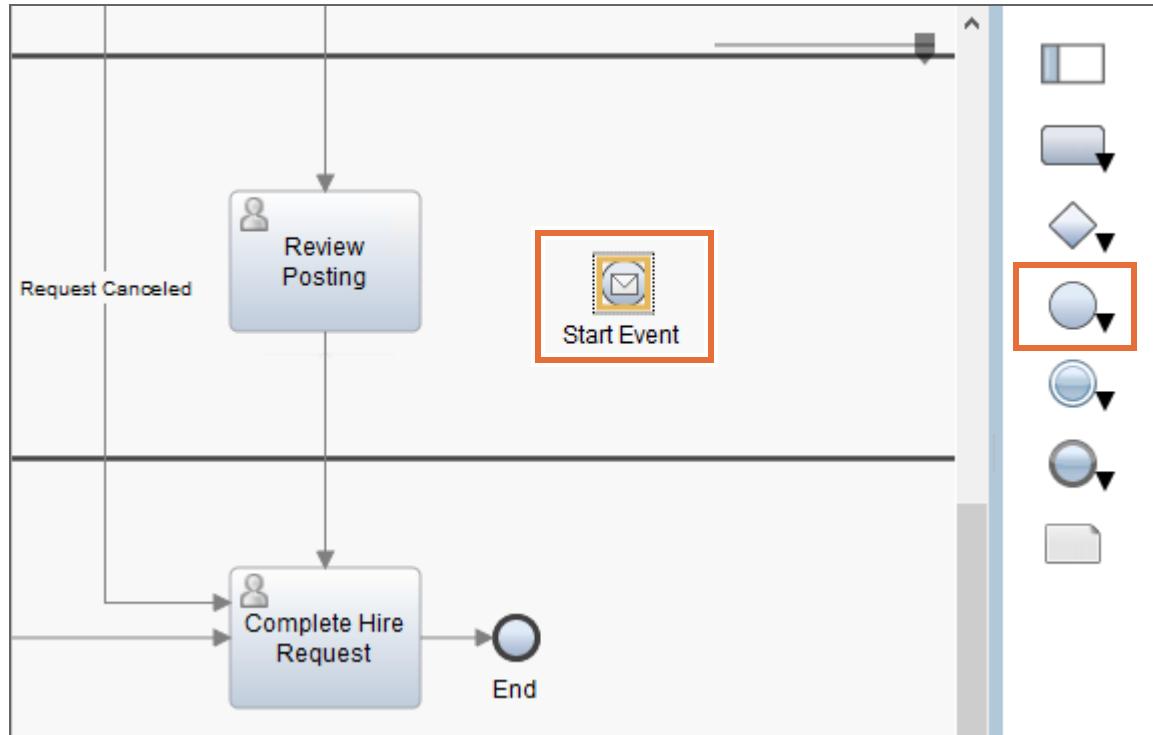
This setting generally provides time for the purging service to complete over the weekend.

The screenshot shows the Undercover Agent configuration interface. The main title is "Undercover Agent". The "Scheduler" tab is active, showing the following settings:

- Schedule Type:** Time Elapsed
- Event Marker:** Timer
- Run Now:** A button to trigger the run immediately.
- Details:** Queue Name: Async Queue, Attached Service: Systematic Request ES, Enabled: checked.
- Parameter Mapping:** A section with a warning icon and a link to "requisitionDetails (Hi...)".
- Time Schedule:** A grid of dropdown menus for scheduling. The columns represent months (January to October), frequency (Every ...), days of the week (Monday to Sunday), times (Midnight to 9:00 am), and minutes (On the hour to 45 past the hour). The "Saturday" option is selected in the days column.

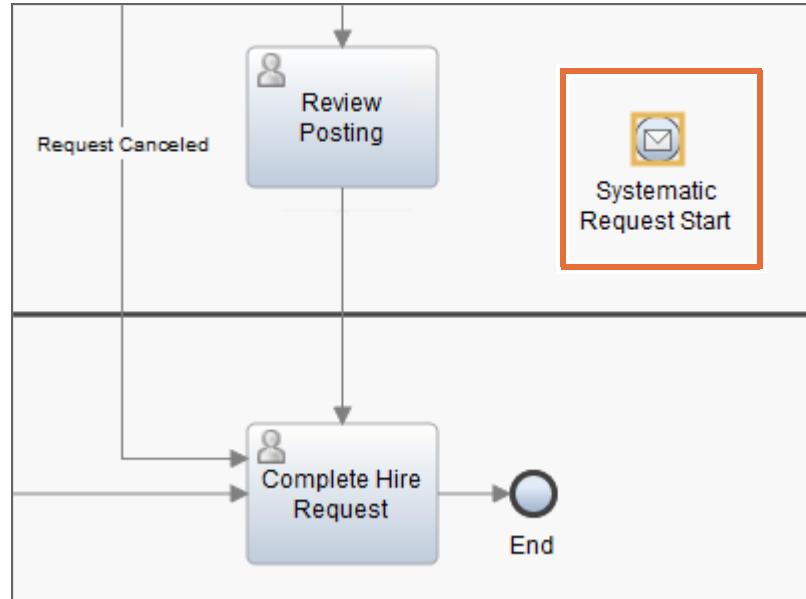
- ___ e. Save your work.
- ___ 4. Implement the **Systematic Request** start message event on the **Hiring Request Process** process.
 - ___ a. Open the **Hiring Request Process** process.

- ___ b. Drag a **Start** event from the palette to the process near the **End** event.

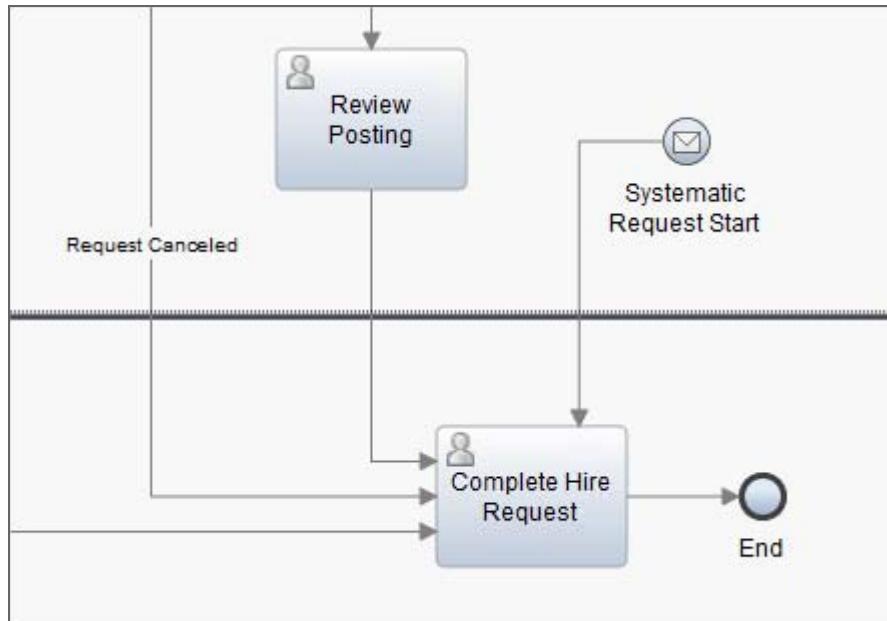


Because a start event is already on the canvas, subsequent start events added to the canvas default to a Start message event. You can also expand the start event in the palette and drag a Start message event to the canvas to get the same result.

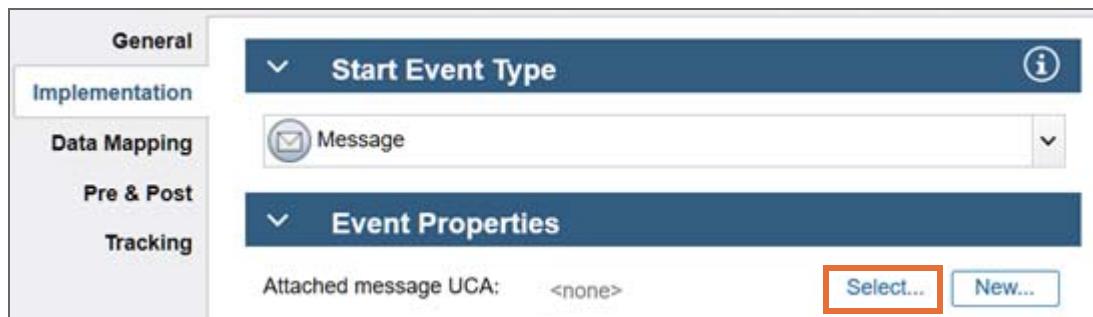
- ___ c. Rename the start message event to: Systematic Request Start



- ___ d. Connect Systematic Request Start to the **Complete Hire Request** activity.



- ___ e. Click the **Systematic Request Start** event, click the **Implementation** properties menu, and then click **Select** next to **Attached message UCA**.



- ___ f. Select the **Systematic Request** UCA that you created for the activity.
___ g. Save your work.



Troubleshooting

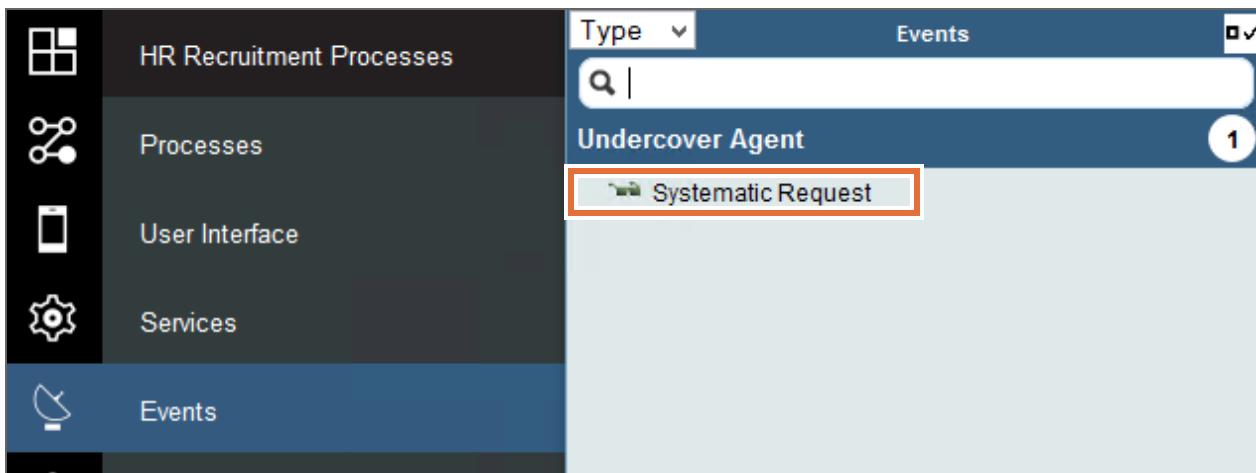
If you do not see the Systematic Request UCA, restart Process Designer.

- ___ 5. Map the Systematic Request start event UCA outputs to the process variables.
___ a. Click the **Data Mapping** properties menu.

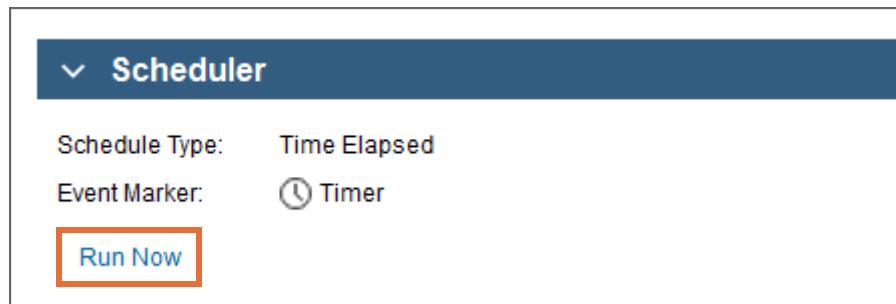
- ___ b. In the **Output Mapping** section, map the requisitionDetails output variable to the requisitionDetails (HiringRequisition) variable.



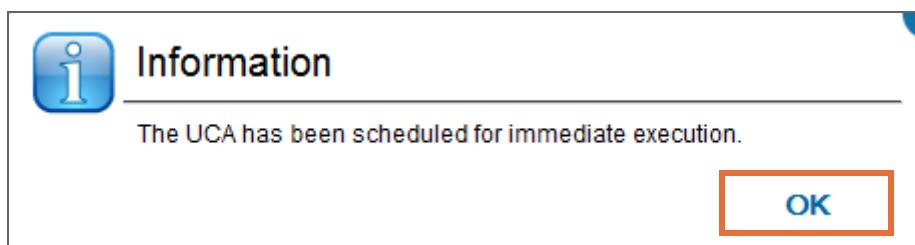
- ___ c. Save your work.
 ___ 6. Test the UCA.
 ___ a. Open the **Systematic Request** UCA listed in the Events section of the library.



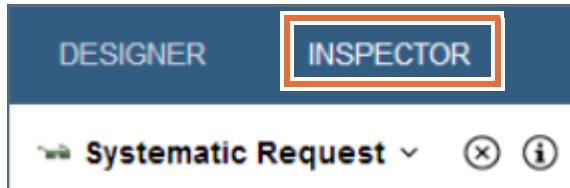
- ___ b. Click **Run now** in the Scheduler section.



- ___ c. A dialog box is displayed, which indicates that the UCA is scheduled for immediate execution. Click **OK**.



- __ d. Click the **INSPECTOR** view tab.



- __ e. At the top, click the first icon to **search** process instances.



- __ f. On the left, underneath the search filter criteria, click **Search** to view all the instances of all the processes in the Process Center Server.

The Search button changes to Refresh after using it. If you have used it before, it might display Refresh instead of Search.

- __ g. If everything was successful, the UCA and Message Start event created a **Hiring Request Process** instance. The instance that you created when you ran the test is the highest numbered instance that is shown in the right frame. This window shows all the instances that ran in Process Designer and their status, although your environment might not match what you see in this screen capture.

A screenshot of a search interface. On the left, there is a sidebar titled "Filter process instances by" with sections for "Status" (Active, Completed, Failed, Suspended, Terminated) and "Severity type". Below that are fields for "Person" (Name or user name) and "Last modified date" (From Date/Time and To Date/Time). At the bottom is a large blue "Refresh" button. On the right, there is a main panel with a search bar at the top. Below the search bar are links: "Select shown instances", "Select all instances", and "Clear selection". The main area displays a single result: "▶ Hiring Request Process:253 + Hiring Request Process" with the note "Last modified Apr 28, 2019 Due Apr 29, 2019".

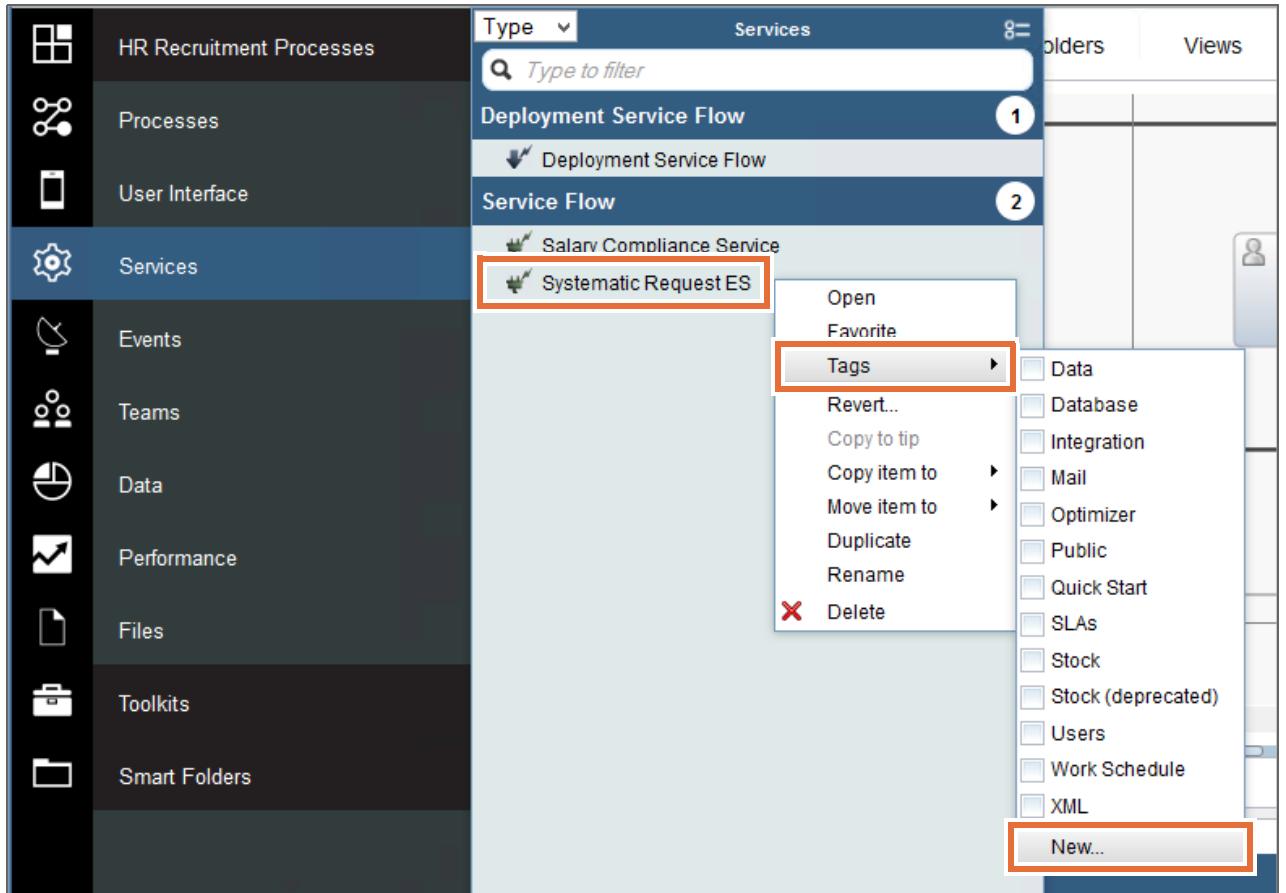
Part 3: Apply asset tags

The Process Designer library now has a number of assets for the **Hiring Request Process** process. You can use asset tagging to organize your existing process application assets. Asset tagging allows developers to accomplish tasks such as associating a UCA with its enabling service.

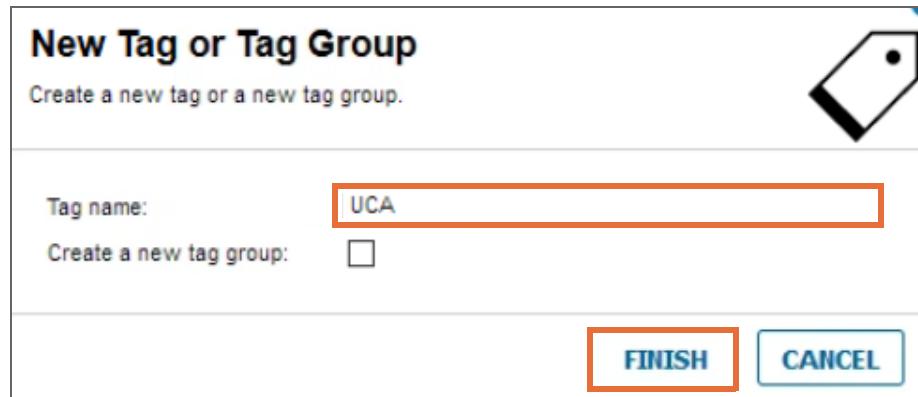
- __ 1. Tag the **Systematic Request Start** service.
 - __ a. Click **DESIGNER** to return to the Designer mode of the Process Designer.



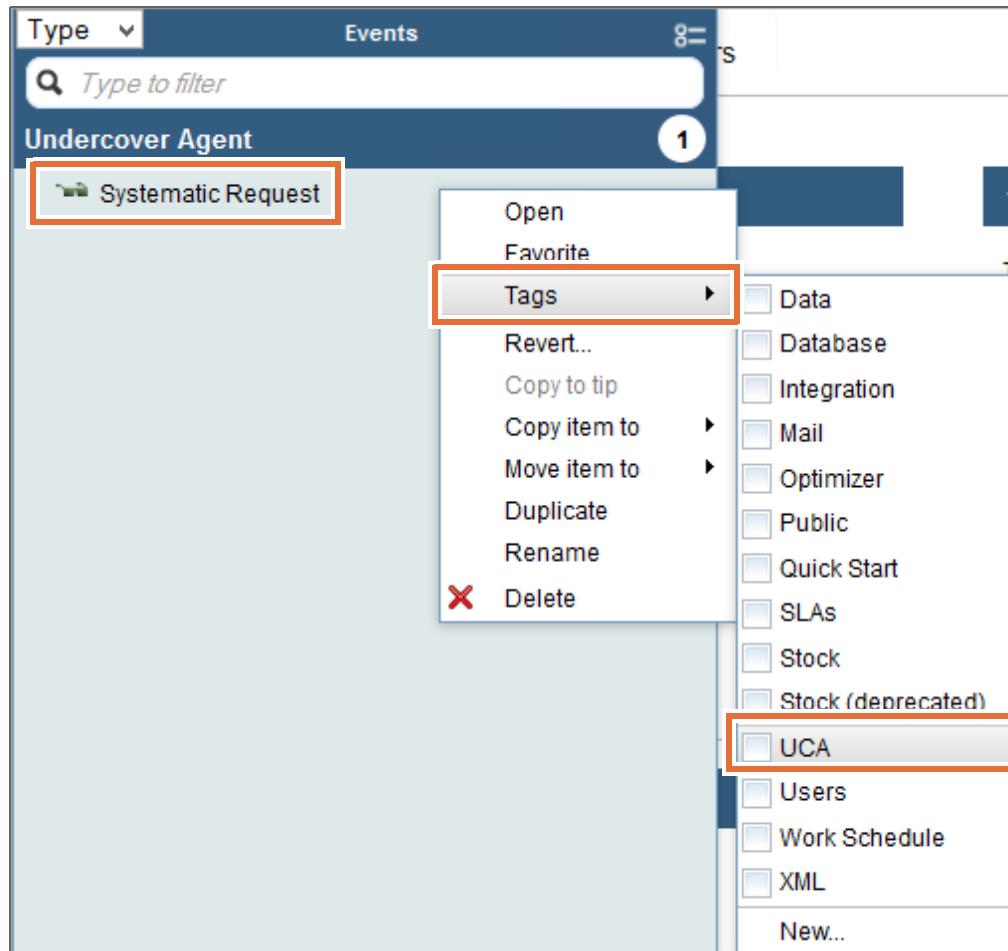
- __ b. In the Designer library, open the **Services** category.
- __ c. Right-click the **Systematic Request ES** service.
- __ d. Click **Tags > New**.



- __ e. Name the tag UCA and then click **FINISH**.



- __ f. Return to the library and open the **Events** category.
__ g. Right-click the **Systematic Request** UCA.
__ h. Click **Tags > UCA**.



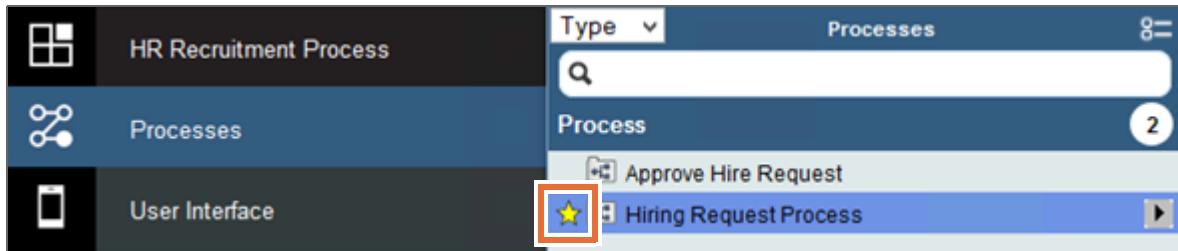
- ___ i. Click the **HR Recruitment Process** header at the top. This category lists all the artifacts in the library. The two artifacts are now tagged with UCA in parentheses after the artifact name.

The screenshot shows a library interface with a search bar and a 'Type' dropdown set to 'All'. The results are grouped into categories: 'Process' (2 items), 'Deployment Service Flow' (1 item), 'Service Flow' (2 items), 'Team' (1 item), 'Undercover Agent' (1 item), and 'Theme' (1 item). The 'Service Flow' category contains 'Salary Compliance Service' and 'Systematic Request ES (UCA)', both of which are highlighted with orange boxes. The 'Undercover Agent' category contains 'Systematic Request (UCA)', also highlighted with an orange box.

- ___ 2. Group artifacts by tag.
- ___ a. From the **HR Recruitment Process** menu, you can view all the assets in the library. At the top of the list, the default grouping is by Type. Click the arrow to the right of **Type**. Select **Tag** from the list.
- ___ b. The artifacts with the tag **UCA** are now grouped in the list.

The screenshot shows the same library interface, but the 'Type' dropdown is now set to 'Tag'. The results are grouped into 'No Tags' (6 items) and 'UCA' (2 items). The 'UCA' group contains 'Systematic Request (UCA)' and 'Systematic Request ES (UCA)', both of which are highlighted with orange boxes.

- ___ c. After viewing the **Name** grouping type, return the grouping type back to **Type**.
- ___ 3. Create a favorite.
 - ___ a. Click the **Processes** category.
 - ___ b. Hover to the left of the **Hiring Request Process**. The outline of a star appears next to the process name.
 - ___ c. Click the **star** next to the **Hiring Request Process** to designate it a favorite.



- ___ d. If it is not already expanded, open **Smart Folders** at the bottom of the library. Click **Favorites** to verify that the process is now part of the **Favorites** smart folder.



Optional

You have completed part 3 of 5, so you are now half done with this exercise. Now would be a good time to take a 10-minute break.

Part 4: Create a service to query a database and populate a list

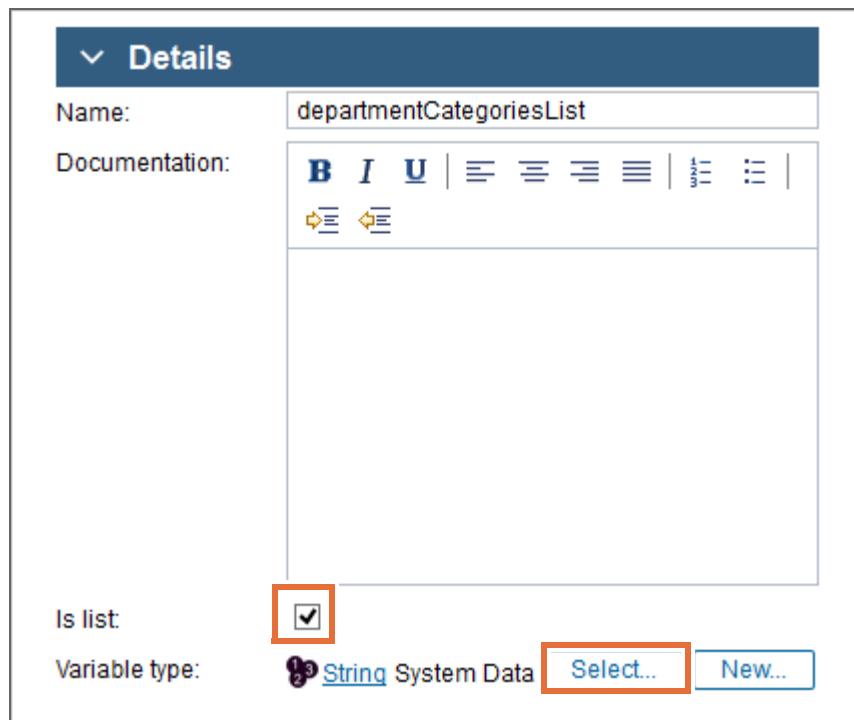
In this part of the exercise, you create a service to query a database and populate a list. The current systems require employees to look up different codes and enter them into the system. Because customers use this system, a better approach is to offer a menu with readable options that correspond to the codes in the database. The Job Level and Department Details values make up the menus.

The tables are already established in the TRAINING database as JOBLEVELS. You build a new business object that is called DepartmentDetails that contains the parameters: Division and Department. These parameters match the DIVISIONS and DEPARTMENTS tables in the TRAINING database.

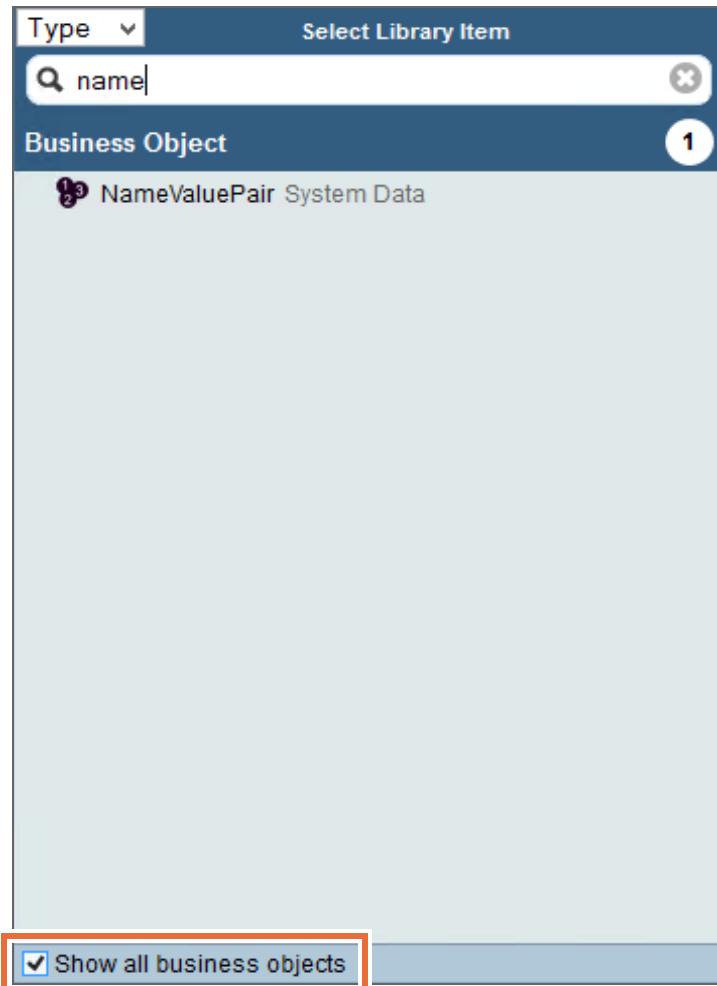
Use an environment variable (ENV) to hold database information for one query, and an exposed process variable (EPV) to hold the database information for the other.

- ___ 1. Create a service to retrieve the incident categories.
 - ___ a. In the Process Designer library, click the (+) plus sign next to **Services** and select **Service Flow**.
 - ___ b. Name the service **Retrieve Department Categories** and click **FINISH**.

- ___ 2. Add variables to the service. These variables hold the list data that is retrieved from the database.
 - ___ a. Click the **Variables** tab.
 - ___ b. Add an output variable named: departmentCategoriesList
 - ___ c. Select the **Is List** checkbox and then click **Select** for the variable type.



- ___ d. Select the **Show all business objects** checkbox at the bottom, and select **NameValuePair** from the list.



This variable is used to populate the list of options in a select menu on the coach.

- ___ e. Now add a **Private** variable: `sql (String)`



- ___ 3. Add the elements onto the canvas.
- ___ a. Click the **Diagram** tab of the service.
- ___ b. Drag a **Server Script** from the service palette to the canvas.

A **green plus (+)** is displayed when you are in an area that the item can be dropped. A **red circle with an (x)** appears if the mouse is hovering in an area where it cannot be dropped. Make sure the sequence line is highlighted and the mouse pointer turns to a **green plus (+)** sign.



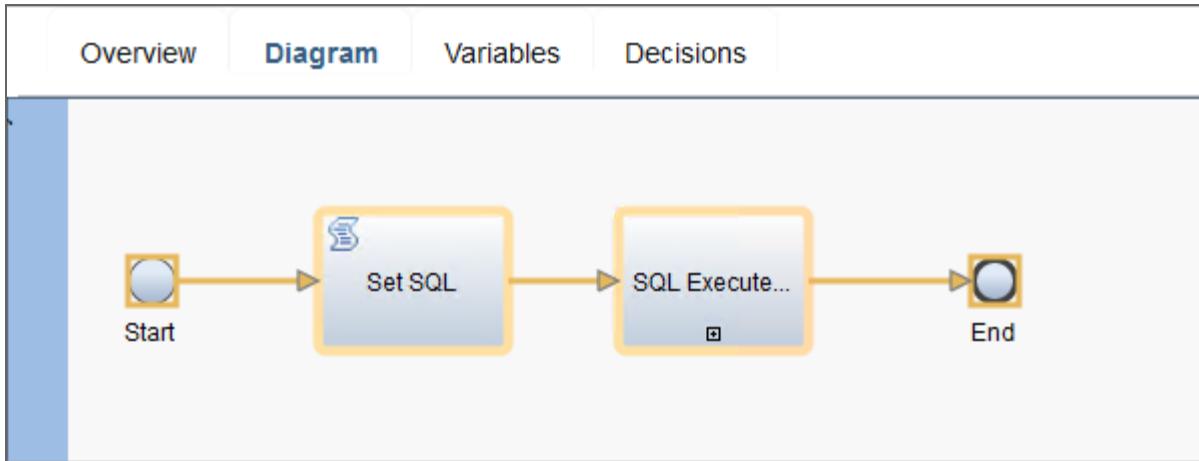
- c. The activity is placed inside the sequence flow.
- d. Rename the server script to: Set_SQL
- e. Expand the **Toolkits > System Data** toolkit category in the Process Designer library.
- f. Select the **Services** category. Drag the **SQL Execute Statement** service onto the sequence flow to the right of the Set_SQL server scriptlet.

Category	Service
Services	SLA Send Alert Email (SLAs)
	SQL Blob to File (Database)
	SQL Blob to File (SQLStatement) (Database)
	SQL Call Multiple Stored Procedures (Database)
	SQL Call Multiple Stored Procedures (SQLResult)
	SQL Call Stored Procedure (Database)
	SQL Call Stored Procedure (SQLResult) (Database)
	SQL Call Stored Procedure (SQLStatement) (Database)
	SQL Clob to File (Database)
	SQL Clob to File (SQLStatement) (Database)
SQL Execute Multiple Statements (Database)	
SQL Execute Multiple Statements (SQLResult)	
SQL Execute Script (SQLResult) (Database)	
SQL Execute Statement (Database)	
SQL Execute Statement (SQLResult) (Database)	
SQL File to Blob (Database)	
SQL File to Blob (SQLStatement) (Database)	
SQL File to Clob (Database)	
SQL File to Clob (SQLStatement) (Database)	
SQL Get Database Type (Database)	

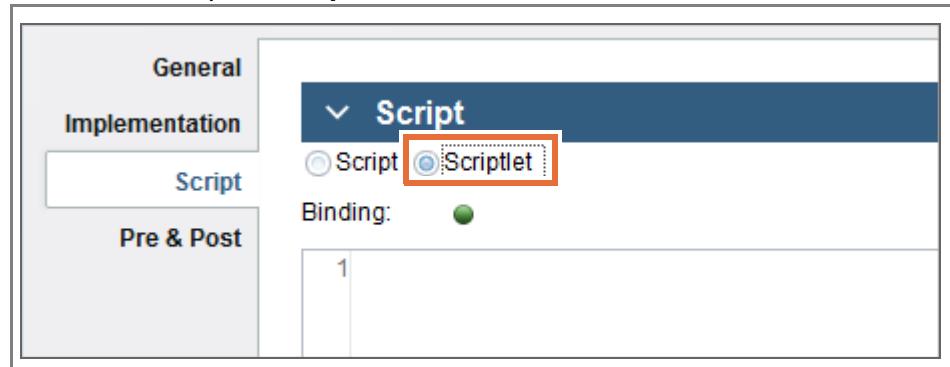
You can find the service that is grouped in with the services that are the Service Flow type, or change to sort by tag and view those items that contain the Database tag. You can also type `SQL Execute` to filter the list to find the service. Be sure to select the correct one as there are several **SQL Execute** services.

If you have problems with dragging the service from the library to the canvas, you can drag a Service Task activity from the palette to the canvas. Then, implement the service by using the SQL Execute Statement service from the System Data toolkit.

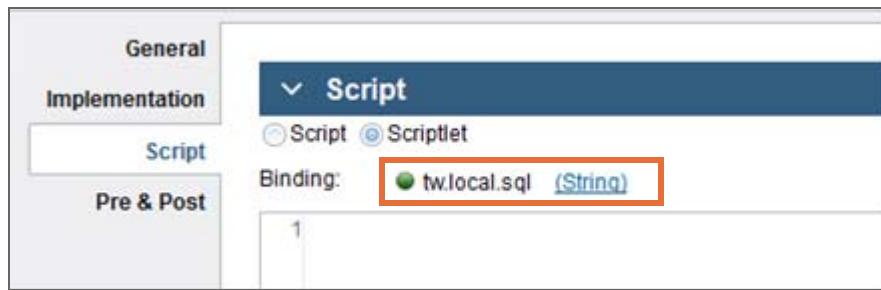
- ___ g. Verify your service flow.



- ___ h. Save your work.
- ___ 4. Implement the Set SQL server scriptlet.
 - ___ a. Click the Set SQL server script.
 - ___ b. Click the **Script** properties menu.
 - ___ c. You use a server scriptlet when assigning a value to a single variable. Change the service from Script to **Scriptlet**.

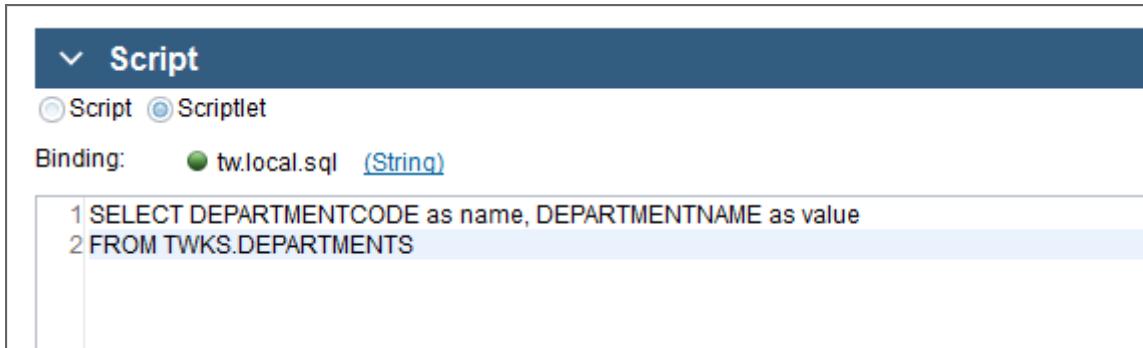


- ___ d. Click **Select** next to **Binding:** and click the **sql (String)** variable.

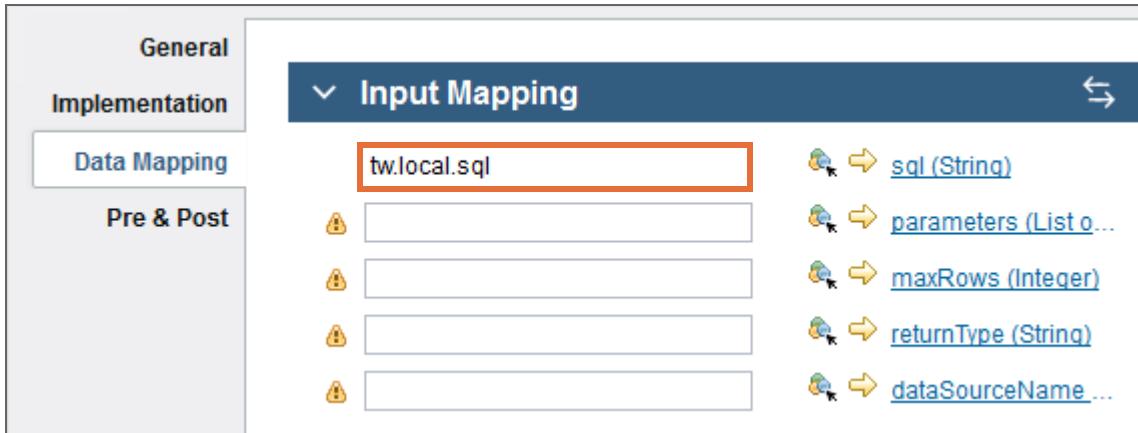


- ___ e. Enter the following SQL query in the script field. The intent is to map the results directly into a **NameValuePair** object, so you must rename the results to enable that direct mapping. Optionally, you can copy the script from the `Script2.txt` file that is located in the `C:\labfiles\Support files\Ex8` folder.

```
SELECT DEPARTMENTCODE as name, DEPARTMENTNAME as value
FROM TWKS.DEPARTMENTS
```



- ___ 5. Map the inputs and outputs of the SQL Execute Statement service.
- Click the **SQL Execute Statement** service step.
 - Click the **Data Mapping** properties menu option.
 - In the **Input Mapping** section, map the `tw.local.sql` variable to the `sql (String)` variable.



- For the `returnType (String)` variable mapping, type (include the quotation marks): "NameValuePair"

- ___ e. A JDBC data source was already created for you. In the **dataSourceName** field, type "jdbc/TrainingDB" (include the quotation marks).

- ___ f. In the **Output Mapping**, map the results (ANY) variable to the tw.local.departmentCategoriesList variable.

- ___ g. Click the **debug** icon to debug the service.
 ___ h. Make sure that you are on the **Diagram** tab so you see the token move between the steps in the service. Expand the **Data** section on the right and click **Step over**. If needed, you might need to click the Search icon from the top to close the search window.

The first step assigns the string to the sql variable for the database call.

- ___ i. For the SQL Execute step, click **Step Over** until the status changes to Finished.

**Important**

Two commands in the debugger are for stepping through a nested service: **Step Over** and **Step Into**. When a step in your service is a nested service that contains more than one step, you can use **Step Into** to debug the steps inside the nested service. If you click **Step Over**, the debugger runs all of the steps and moves the token to the end of the nested service, showing only the output after the service is complete.

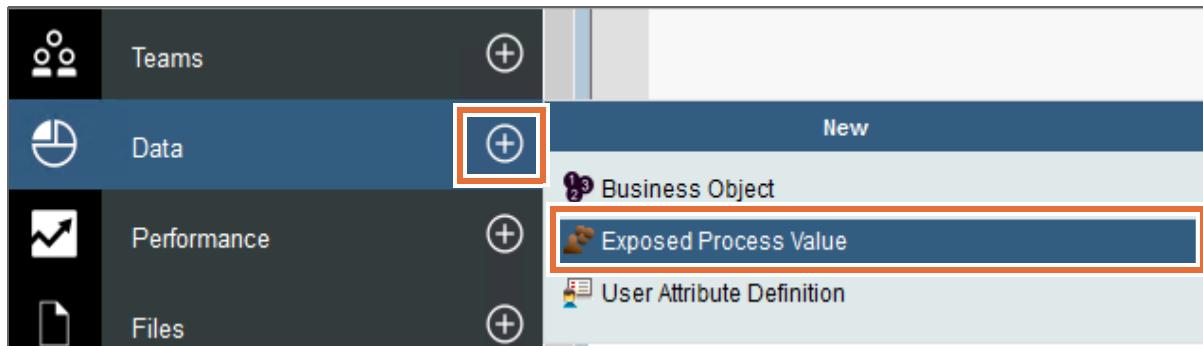
- ___ j. Expand the `departmentCategoriesList` in the Data section of the Inspector. The `departmentCategoriesList` variable is populated with the values in the database. Notice that the `departmentCategoriesList` variable contains items 0 - 4 as part of the list. This result means that the database returned five rows, which were mapped to the `NameValuePair` object per the configuration options of the **SQL Execute** service.

The screenshot shows the Oracle APEX Debugger interface. At the top, there is a title bar with a checkmark icon and the text "Retrieve Department Categories". Below the title bar, a "Tip" button is visible. The main area is divided into sections: "Details", "Locations", and "Data". The "Data" section is expanded, showing a list of items under the heading "departmentCategoriesList(NameValuePair)(list)". Each item is represented by a row with a plus sign icon and indices "item[0]" through "item[4]". The "name(String)" field for "item[0]" is "101" and the "value(String)" field is "Marketing". The "Data" section also contains a SQL query: "sql(String) SELECT DEPARTMENTCODE as nan FROM TWKS.DEPARTMENTS". The entire "Data" section is highlighted with a red rectangle.

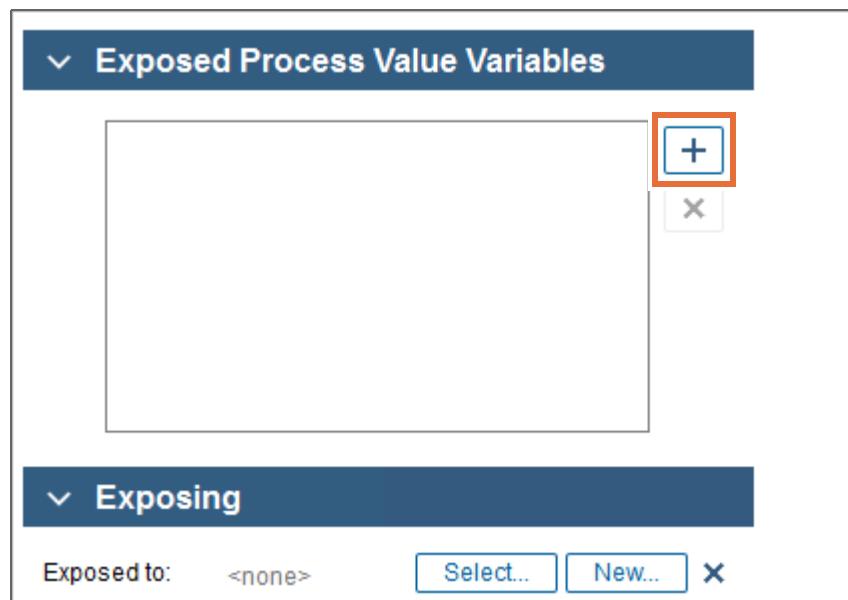
- ___ k. Click **DESIGNER** to return to the Designer view.
___ 6. Add an exposed process variable (EPV) to manage the data source name.

The data source name should normally be stored as an environment variable. EPVs are designed to store business data because they can be exposed to business users so they can set real-time process variables. In the next two steps, you practice the creation and use of both environment variables and EPVs.

- ___ a. Click the (+) plus sign in the **Data** menu option in the Process Designer library.
- ___ b. Select **Exposed Process Value**.



- ___ c. Name your EPV `DataSource` and click **FINISH**.
- ___ d. In the **Exposed Process Value** settings, click the (+) plus sign in the **Exposed Process Value Variables** section.



__ e. Set the following values in the **Variable Details** section:

- **External Name:** TrainingDatabase
- **Variable Name:** trainingDB
- **External Description:** This is the name of the JDBC data source for the training database.
- **Default Value:** jdbc/TrainingDB

Variable Details

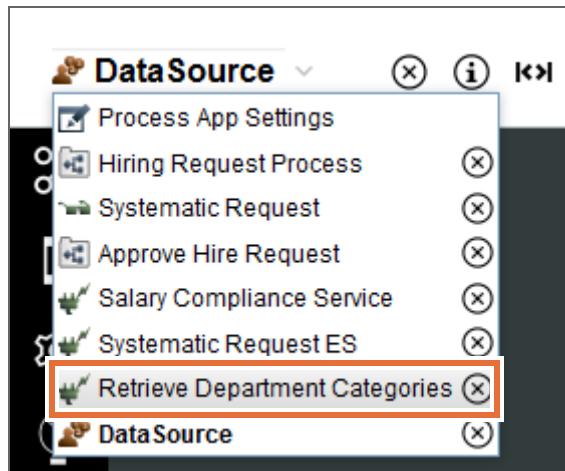
External name:	TrainingDatabase
Variable name:	trainingDB
External description:	<p>B I U ≡ ≡ ≡ ≡ </p> <p>≡ ≡ ⌂ ⌂ ⌂</p> <p>This is the name of the JDBC data source for the training database.</p>
Variable type:	1 2 3 String System Data Select...
Default value:	jdbc/TrainingDB
Use new values:	<input type="checkbox"/>

Because the variable type is String, you do not need to enclose the value in quotation marks.

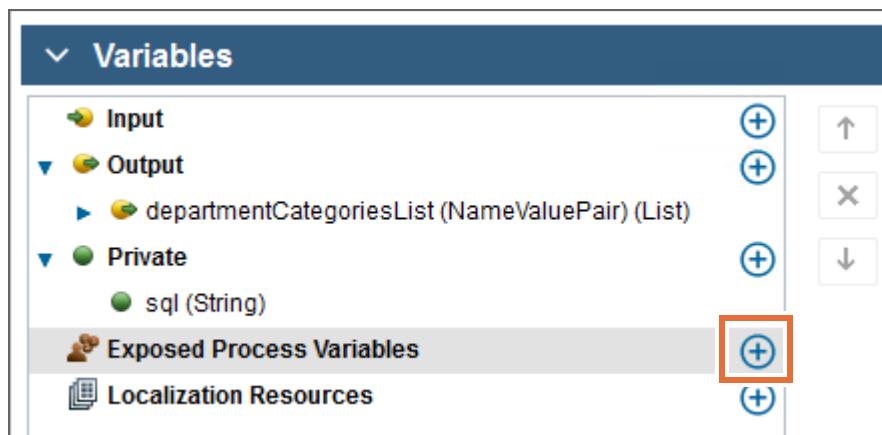
__ f. Save your work.

__ 7. Map the EPV

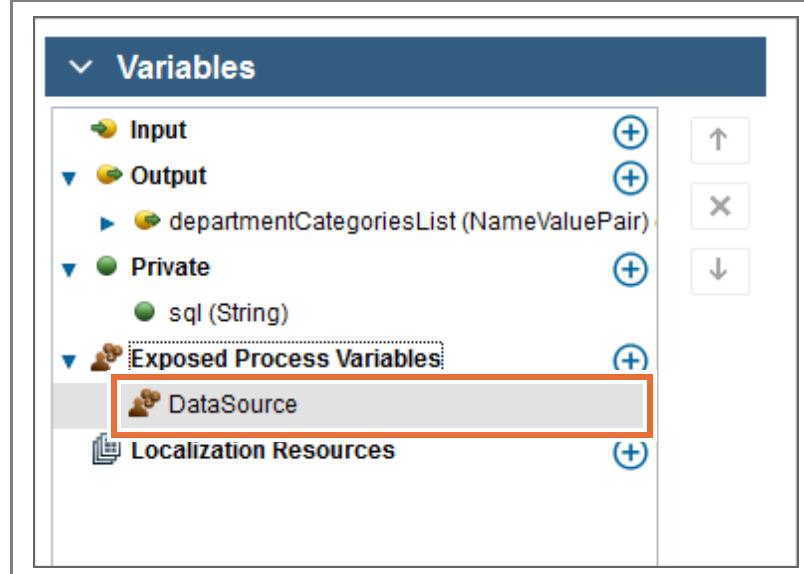
- __ a. At the upper left of the designer, open the History menu and return to the **Retrieve Department Categories** service.



- __ b. Open the **Variables** tab.
 __ c. Click the (+) plus sign next to the **Exposed Process Variables** category in the Variables list, and then select the **DataSource** EPV.

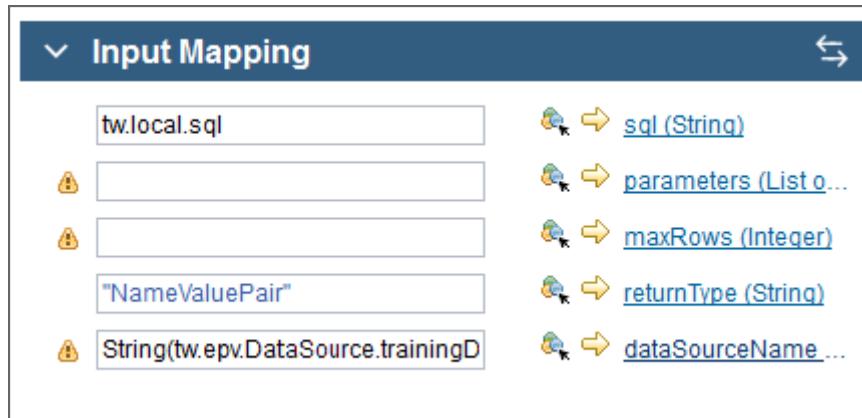


- __ d. Verify that you added DataSource to the Exposed Process Variables variables list.



- __ e. Click the **Diagram** tab to return to the service objects. Click the **SQL Execute Statement** step.
 __ f. In the properties section below, click the **Data Mapping** menu option.
 __ g. Map the `trainingDB` EPV to the `dataSourceName` variable. Enter as the **dataSourceName**: `String(tw.epv.DataSource.trainingDB)`

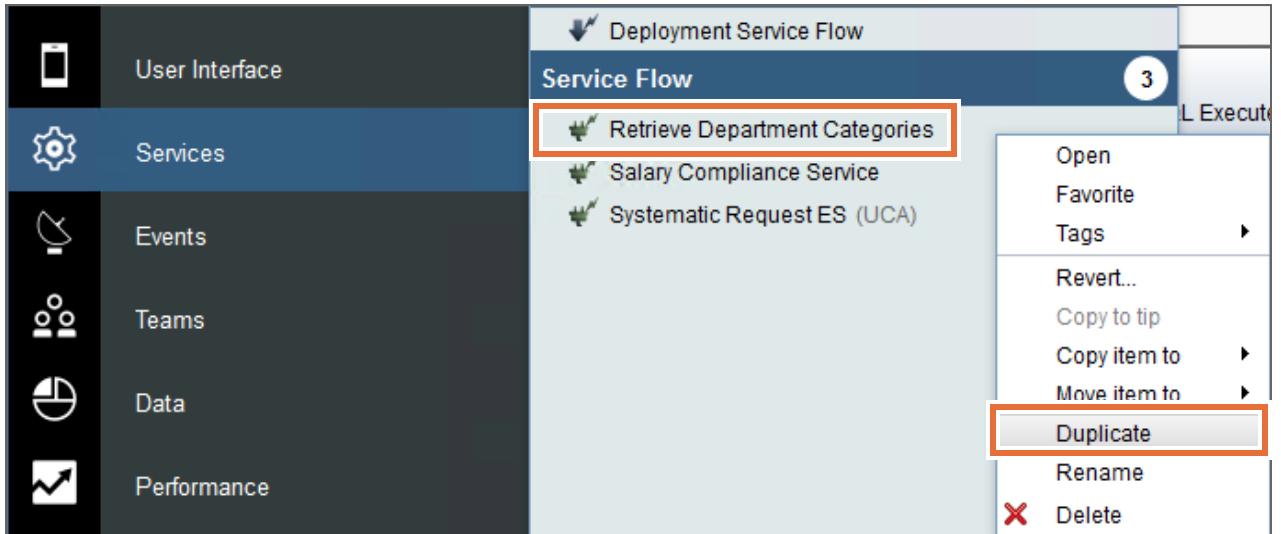
You must cast your EPV to a String. Replace the string that you previously mapped to the variable with `String(tw.epv.DataSource.trainingDB)`. Although you created the EPV as a String type, the system defines it as an EPV type when you map the variable. Therefore, you must cast the variable to String to use it.



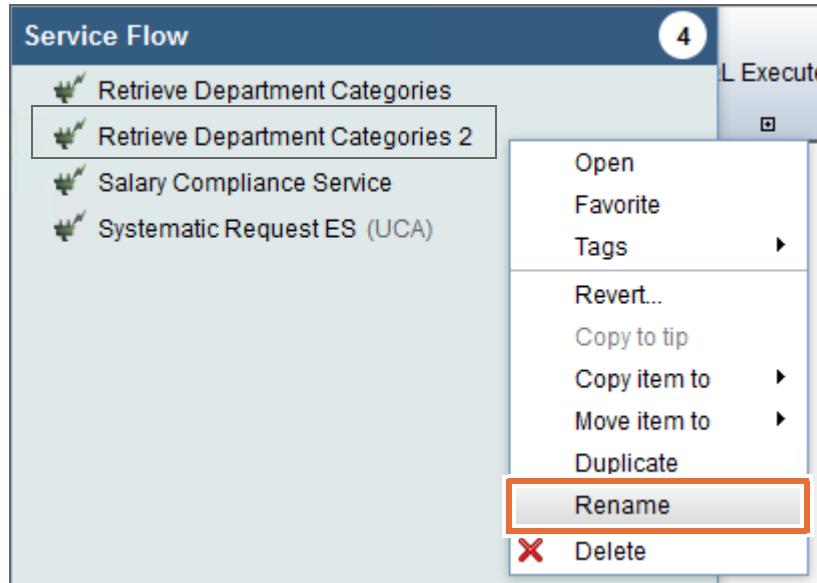
The warning indicator stays because it cannot validate the complex expression. This is normal.

- __ h. Save your work.

- ___ 8. Create a **Service Flow** to retrieve job levels; however, use an environment variable (ENV) to accomplish this implementation.
- ___ a. In the Designer library, click the **Services** category. Right-click the **Retrieve Department Categories** service flow and click **Duplicate**.



- ___ b. Right-click the new service and click **Rename**. Name the new service **Retrieve Job Levels**. Click **FINISH**.



- ___ 9. Change the new service to retrieve job levels.
- ___ a. Open the new **Services > Retrieve Job Levels** service flow. Because many of these database calls are similar, you just need to change a few things to retrieve the job levels from the database.
- ___ b. Click the **Variables** tab.
- ___ c. Click the **departmentCategoriesList Output** variable. Change the variable name to: **jobLevels**.
- ___ d. Ensure that the **Is List** checkbox is selected, and the Variable Type is **NameValuePair**.

- __ e. Save your work.
- __ f. Verify the private variable `sql` (`String`) is still present in the variables list.

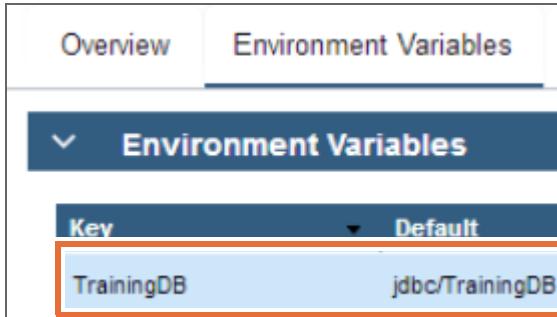
The screenshot shows the 'Variables' panel in a software application. The 'Variables' section is expanded, displaying categories: Input, Output, Private, Exposed Process Variables, DataSource, and Localization Resources. Under the 'Private' category, there is a single entry: 'sql (String)', which is highlighted with a red rectangular box. To the right of each entry are four circular icons: a plus sign (+), an upward arrow (↑), a downward arrow (↓), and a minus sign (X).

- __ 10. Add an environment variable to your process application.
- __ a. Open the **History** menu and select **Process App Settings**.

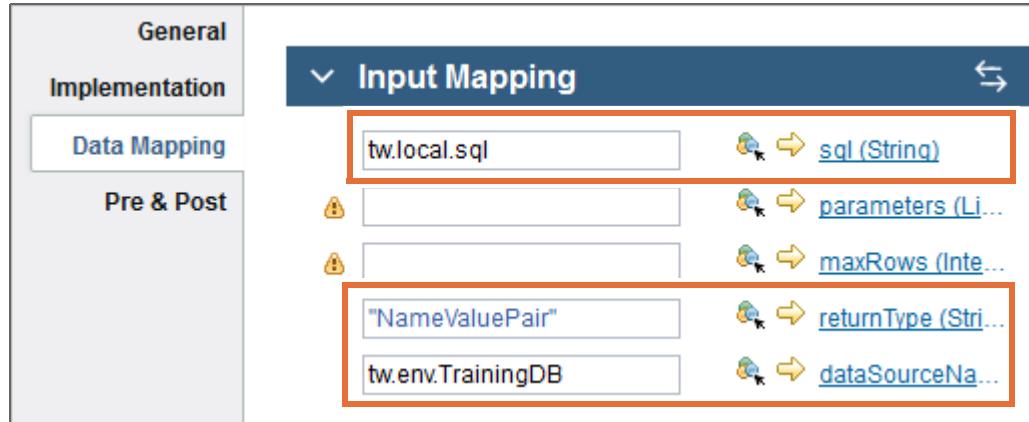
The screenshot shows the 'History' menu with several options listed. The 'Process App Settings' option is highlighted with a red box and has a checked checkbox icon to its left. Other options include 'Hiring Request Process', 'Systematic Request', 'Approve Hire Request', 'Salary Compliance Service', 'Systematic Request ES', 'Retrieve Department Categories', 'DataSource', and 'Retrieve Job Levels'. Each option has a small circular 'X' icon to its right.

- __ b. Click the **Environment Variables** tab and then click the (+) plus sign on the right side of the **Environment Variables** section.

The screenshot shows the 'Environment Variables' tab selected in a navigation bar. Below it is a table with columns: Key, Default, Development, Test, Stage, Production, and a large orange-bordered plus sign (+) button. The 'Key' column header is also highlighted with a red box. The table body is currently empty, showing only the column headers.

- ___ c. Complete the following values for the environment variable:
- **Key:** TrainingDB
 - **Default:** jdbc/TrainingDB
- 
- ___ d. Save your work.
- ___ 11. Implement the Set SQL server scriptlet.
- ___ a. Return to the **Retrieve Job Levels** service flow.
 - ___ b. Click the **Diagram** tab.
 - ___ c. Click the Set SQL server script.
 - ___ d. Click the **Script** properties menu option.
 - ___ e. Verify that the **Scriptlet** type is selected and that the scriptlet is bound to the `tw.local.sql` (String) variable.
 - ___ f. Change the **Implementation** script to:
- ```
SELECT JOBLEVELCODE as name, JOBLEVELNAME as value
FROM TWKS.JOBLEVELS
```
- 
- Alternatively, you can copy the script from the `Script3.txt` file that is located in the `C:\labfiles\Support files\Ex8` folder.
- \_\_\_ g. Save your work.
- \_\_\_ 12. Map the inputs and outputs of the SQL Execute Statement service.
- \_\_\_ a. Click **SQL Execute Statement** on the canvas.
  - \_\_\_ b. Click the **Data Mapping** properties menu option.
  - \_\_\_ c. Under the **Input Mapping** section, verify that the `sql` (String) input is mapped to the `tw.local.sql` variable.

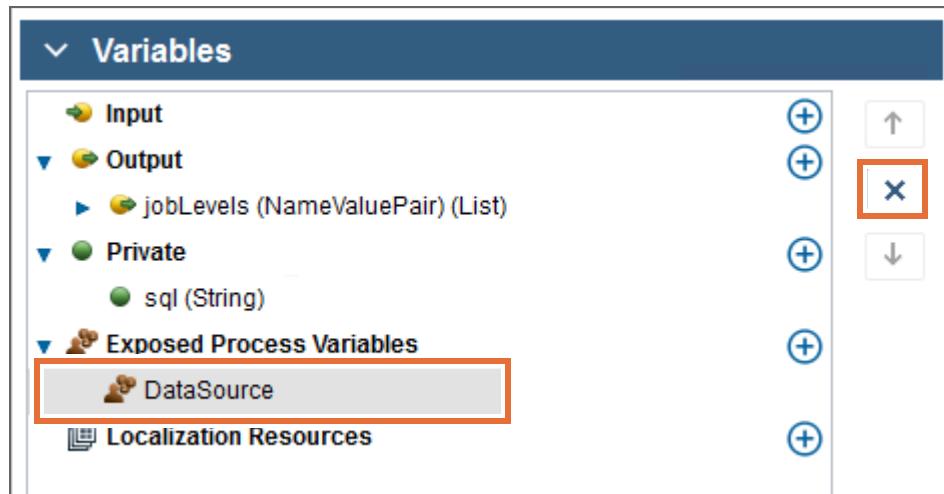
- \_\_\_ d. Verify the `returnType` (`String`) input is “`NameValuePair`” (include the quotation marks).
- \_\_\_ e. For the `dataSourceName` (`String`) input, enter: `tw.env.TrainingDB`



- \_\_\_ f. Change the **Output Mapping** section to the `tw.local.jobLevels` variable.



- \_\_\_ g. Now you can remove the DataSource Exposed Process Variable from the Variables tab that was added to the original service and duplicated for this service.



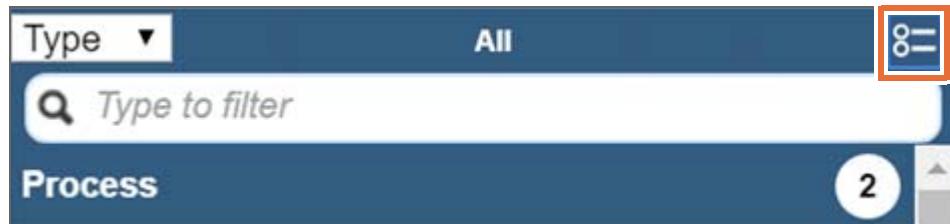
- \_\_\_ h. Save your work.
- \_\_\_ 13. Debug the service to determine whether the data is retrieved from the database correctly.
  - \_\_\_ a. Click the **debug** icon above the palette.

- \_\_\_ b. Click the **Step over** icon twice to step through the service and verify that the database call runs and returns data from the database.

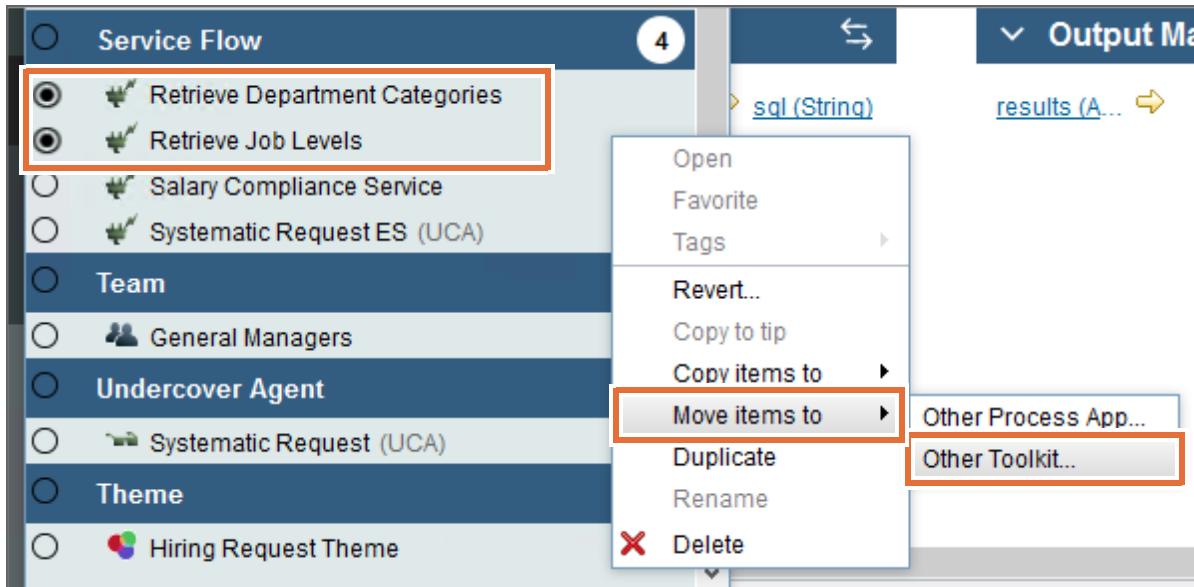
The screenshot shows the 'Retrieve Job Levels' process details. The 'Data' section displays a list of job levels. One item, 'item[0]', is highlighted with a red box. The details for 'item[0]' are: name(String) 5001 and value(String) Jr Associate. Below this, there is a SQL query: sql(String) SELECT JOBLEVELCODE as name, JOBLEVELNAME as value FROM TWKS.JOBLEVELS.

- \_\_\_ c. Click **DESIGNER** to return to the Designer view.
- \_\_\_ 14. To make these services available to other process applications, move the services to the toolkit.
- \_\_\_ a. At the top of the library, click **HR Recruitment Process** to view all the assets in your library.

- \_\_\_ b. On the top of the asset list, click the icon to change to **multi-selection mode**.



- \_\_\_ c. Radio buttons appear to the left of every asset in the library. Select both the **Retrieve Department Categories** and **Retrieve Job Levels** Service Flow services. Right-click and click **Move items to > Other Toolkit**.



- \_\_\_ d. Click **Hiring Requisition Toolkit**.

- \_\_ e. The system identifies the dependencies. Click **MOVE**.

### Move Artifacts to Toolkit: Hiring Requisition Toolkit

**Select artifacts to move:**  
Selected artifacts will be moved.

-  Retrieve Department Categories
-  Retrieve Job Levels

**Select dependent artifacts to move:**  
Broken references might occur if you do not move dependent artifacts.

-  DataSource

**Artifacts with dependencies on selected artifacts:**  
Selected items will be moved. Items that are not moved might cause broken references.

**MOVE** **CANCEL**



## Troubleshooting

If the Hiring Requisition Toolkit is not listed, verify **Let users update the toolkit** is selected.

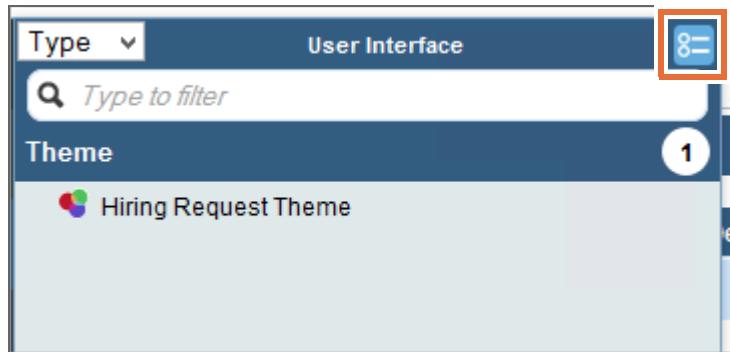
[Back to toolkits /](#) [Open in Designer](#)

### Hiring Requisition Toolkit

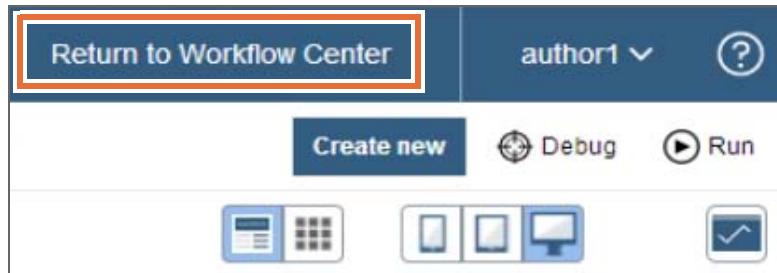
Overview    Snapshots    Permissions    History    Comments

|                                                                                     |                                                                                                                                 |                                              |                                                  |                                                                                                                                                            |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Acronym<br>HRT<br><br>Target environment<br> | Created<br>author1<br>11/3/2020, 10:47:36 AM | Last updated<br>author1<br>11/4/2020, 4:34:55 PM | <br><input checked="" type="checkbox"/> Let users update the toolkit. |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

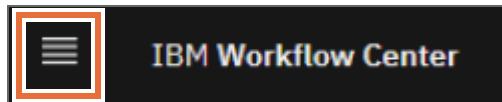
- \_\_\_ 15. In the last exercise, you created a Hiring Request Theme. To make this theme available to other process applications, move the theme to the Hiring Requisition Toolkit.
- \_\_\_ a. Open the **User Interface** category in the library. Turn the **multi-select** off for this next move.



- \_\_\_ b. Right-click **User Interface > Hiring Request Theme** in the library. Click **Move item to > Other Toolkit**, and select the **Hiring Requisition Toolkit**.
- \_\_\_ c. In the Move Artifacts to Toolkit window, click **MOVE**.
- \_\_\_ d. Finally, because the ENV variables are specific to the toolkit or Process App, you need to re-create the environment variable for this toolkit. Click the **Return to Workflow Center** link.



- \_\_\_ e. Click the navigation icon in the upper left.



- \_\_\_ f. Select **Toolkits**.
- \_\_\_ g. Click the **Hiring Requisition Toolkit** tile.
- \_\_\_ h. At the top, click the **Environment Variables** tab.

- \_\_\_ i. In the Environment Variables section, add the TrainingDB key with the default value of jdbc/TrainingDB. This setting is needed to run the Retrieve Job Levels service. Click the (+) Add icon from the right to add a variable.

**Key:** TrainingDB

**Default:** jdbc/TrainingDB



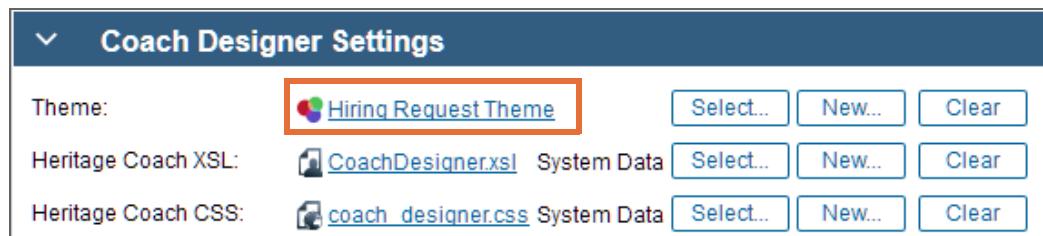
| Key        | Default         |
|------------|-----------------|
| TrainingDB | jdbc/TrainingDB |

- \_\_\_ j. Save your work.

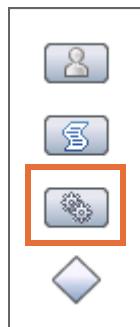
## Part 5: Change an input to a single select on a coach

Until now, any value would be accepted for the Department input on the page. If only certain values are allowed, you must then approach the control in one of the following ways. The first way is to limit the input to only acceptable values by using a selection like the single select on a coach. The second way is through validation of the input data. Many times the solution is a combination of both of these approaches. This section demonstrates how to create a single selection to limit the values that are allowed.

- \_\_\_ 1. Apply the Hiring Request Theme to the Hiring Requisition Toolkit.
  - \_\_\_ a. In the **Overview > Coach Designer Settings** section, click **Select** and select the **Hiring Request Theme**.



- \_\_\_ b. Save your changes.
- \_\_\_ 2. Add the **Retrieve Department Categories** service to the **Hiring Form** human service.
  - \_\_\_ a. Select **User Interface** from the library and click **Hiring Form**.
  - \_\_\_ b. The **Hiring Form** service opens. In the **Diagram** tab, drag a **Service** from the palette onto the **Hiring Form** sequence flow between the **Start** event and the **Hiring Form** coach.

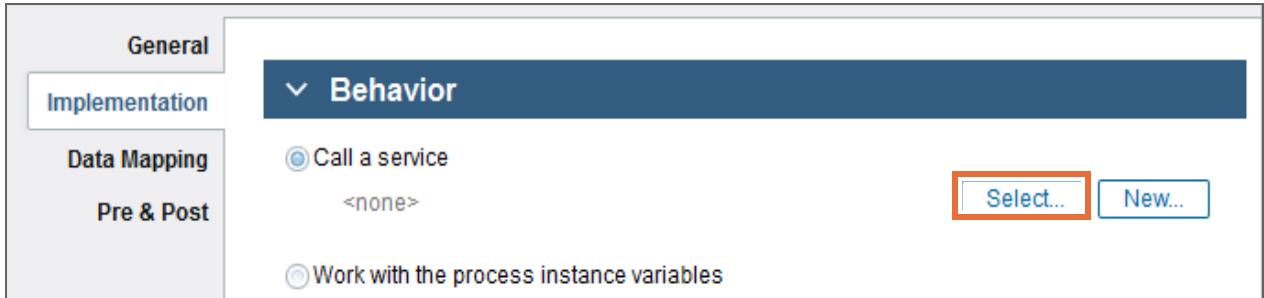


- \_\_\_ c. Verify the flow so that the new service is part of the flow.

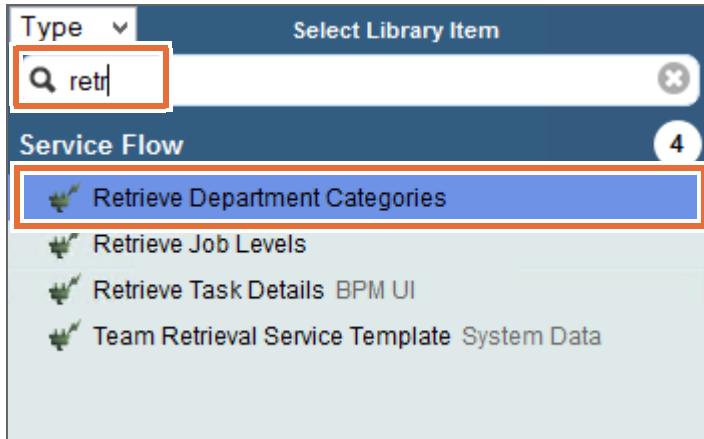


- \_\_\_ d. Rename the new service step to: **Retrieve Department Categories**

- \_\_\_ e. In the **Implementation** properties menu, click **Select** for the **Call a service** setting.

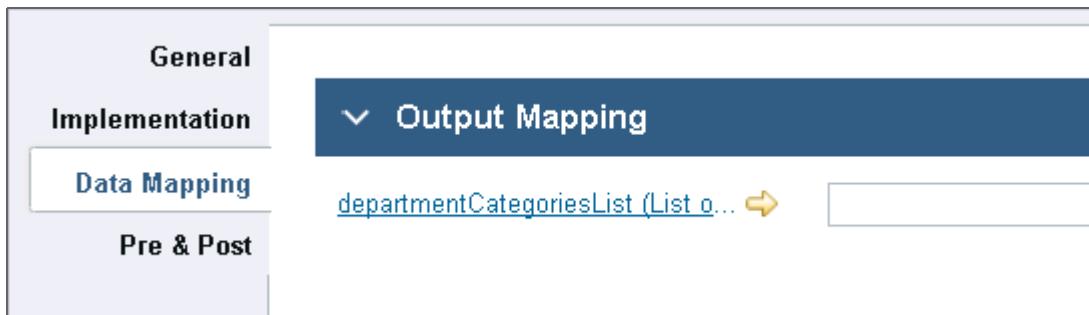


- \_\_\_ f. Select the **Retrieve Department Categories** service that you created.



- \_\_\_ g. Save your work.

- \_\_\_ 3. Map the output variable of the service and allow the system to create the private variable.
- \_\_\_ a. In the **Data Mapping > Output Mapping** properties section, note that the service requires you to map a variable to the `departmentCategoriesList` variable.



- \_\_\_ b. Click the **Variables** tab. Create a private variable: departmentCategoriesList (NameValuePair) (List). Make sure the variable is configured as a List.

The screenshot shows the 'Variables' tab selected in a BPMN tool interface. The 'Variables' section lists several items under categories: Input, Output, Private, Exposed Process Variables, Environment Variables, and Localization Resources. The 'departmentCategoriesList (NameValuePair) (List)' variable is located in the 'Private' category and is highlighted with a red box.



### Reminder

Select **Show all business objects** to choose the NameValuePair variable type.

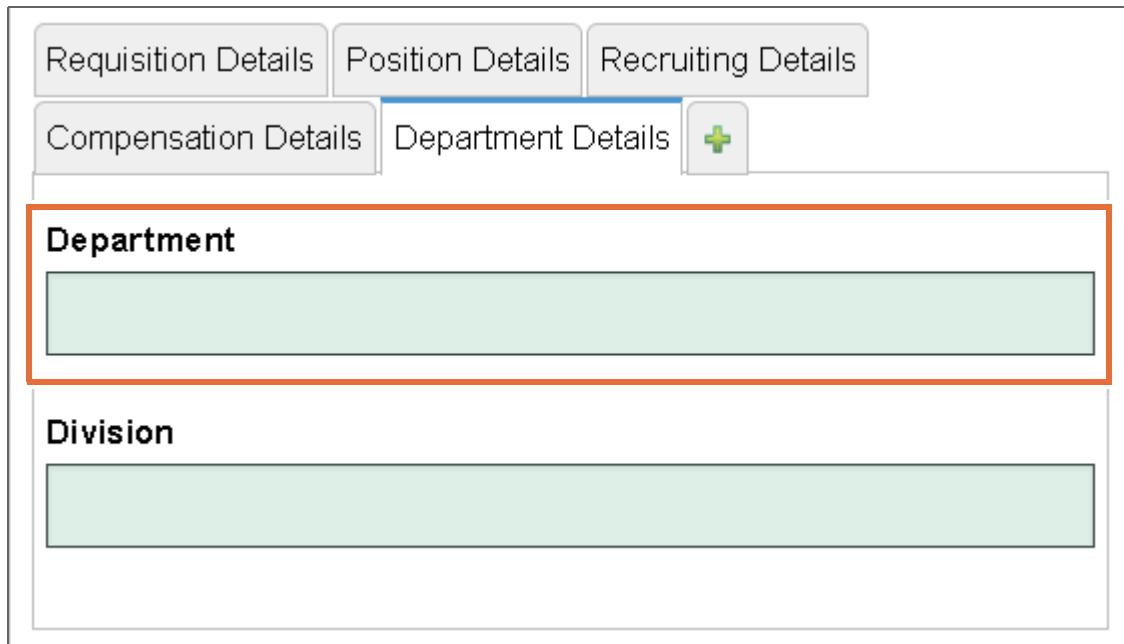
The screenshot shows a dropdown menu listing several business objects, such as BPMAgentDefinitionStatusResponse, BPMBOPropertyChange, and BPMBOSaveFailedError. At the bottom of the list, there is an option labeled 'Show all business objects' which is checked and highlighted with a red box.

- \_\_\_ c. Return to the **Diagram** tab. Verify that the **Retrieve Department Categories** step on the canvas is selected. In the **Data Mapping** properties section, map the departmentCategoriesList variable to the output tw.local.departmentCategoriesList.

The screenshot shows the 'Output Mapping' section in the Data Mapping properties. It displays a mapping between the source variable 'departmentCategorie...' and the target variable 'tw.local.departmentCategoriesList'. The target variable is highlighted with a red box.

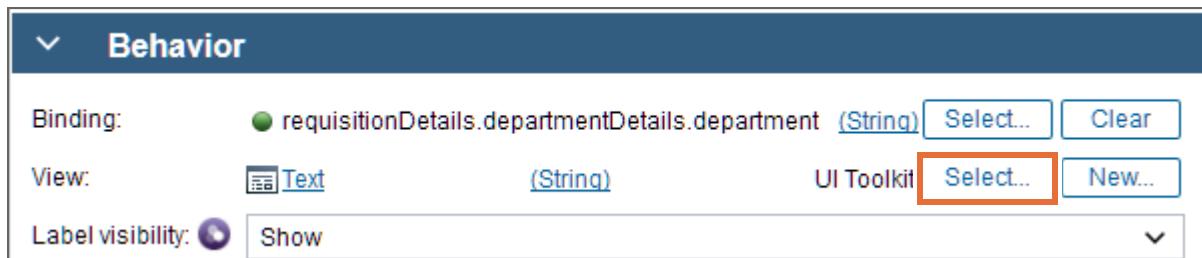
- \_\_\_ d. Save your work.

- \_\_\_ 4. Change the control on the **Hiring Form** coach to a single select.
- \_\_\_ a. Click the **Coaches** tab and the **Hiring Form** coach. Initially when you view the coach, the server must generate the CSS and render the new coach on the screen, so a delay might occur before you see the effect.
- \_\_\_ b. From the **Department Details** tab, select the control **Department** that is bound to the department variable.



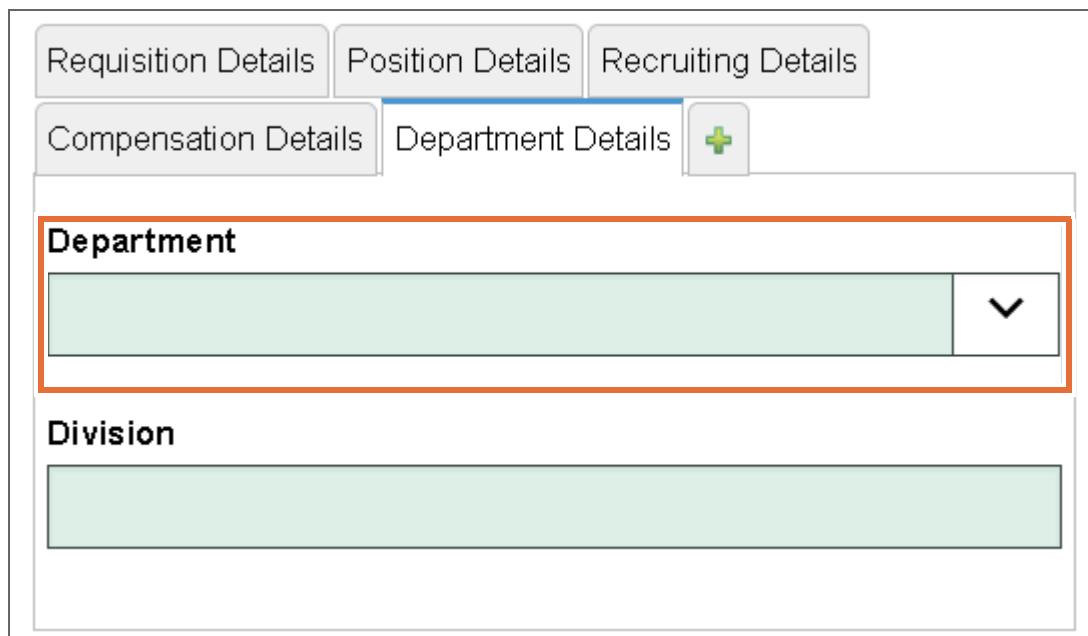
Change the view that the control is bound to. Right now, all the input boxes on the coach are text views. These controls are all variables that are bound to String data types.

- \_\_\_ c. In the **General > Behavior** properties section, click **Select** next to View.

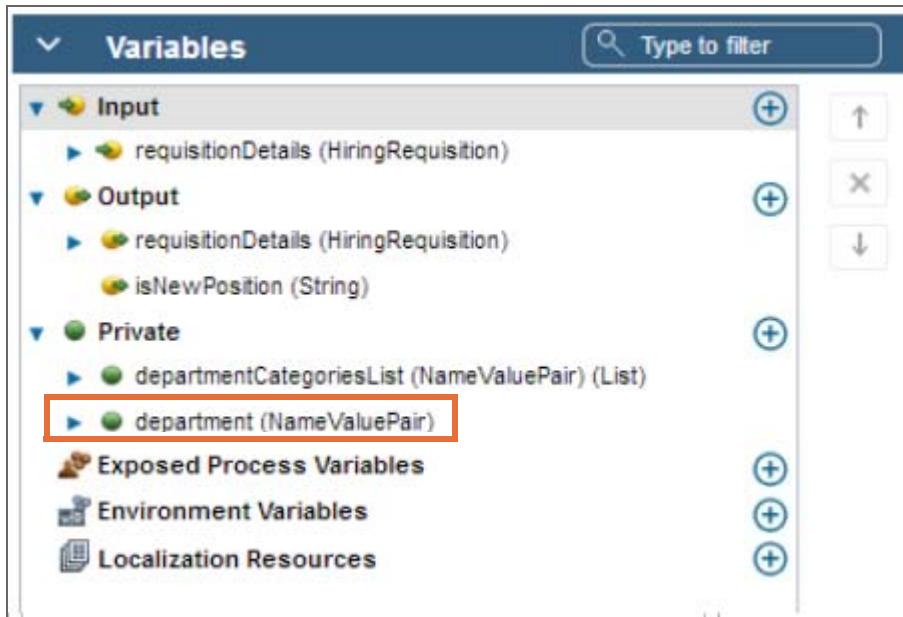


- \_\_\_ d. Select the option **Single Select (UI Toolkit)** coach view.

- \_\_\_ e. After you change the view binding, the control on the palette has a different look. It is now a **Single Select** view control.

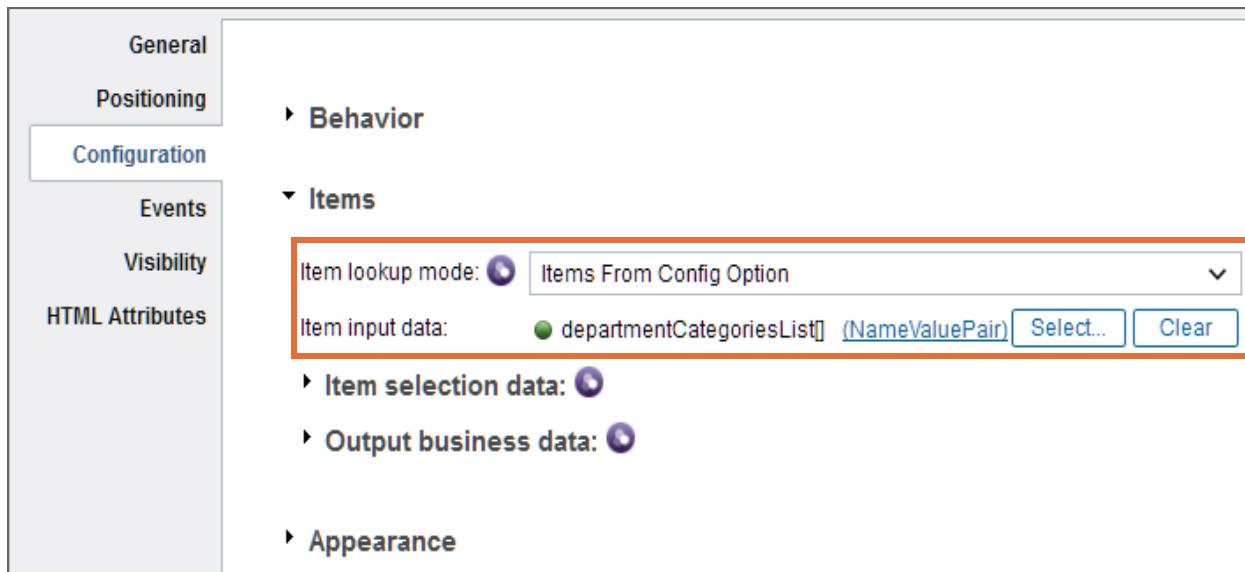


- \_\_\_ f. Save your work.
- \_\_\_ 5. The select control returns the single element that is chosen from the list, and you must create a simple list variable to store the selection.
- \_\_\_ a. Click the **Variables** tab of the **Hiring Form** service.
- \_\_\_ b. Create a private variable: department (NameValuePair)



- \_\_\_ c. Save your work.
- \_\_\_ 6. Store the selected department to the department variable.
- \_\_\_ a. Return to the Coach.

- \_\_\_ b. Select the **Department** control under the **Department Details** tab.
  - \_\_\_ c. In the **General > Behavior** properties section, click **Select** next to the **Binding** property.
  - \_\_\_ d. Click the new Private variable **department (NameValuePair)** from the variable list.
7. Configure the setting that determines the list of selections available to the user on the **Department** select control.
- \_\_\_ a. Click the **Configuration** properties menu. Expand the **Items** section.
  - \_\_\_ b. Change the Item lookup mode to **Items from Config Option**. Click **Select** next to the Item input data setting and click the **departmentCategoriesList (NameValuePair)** variable.



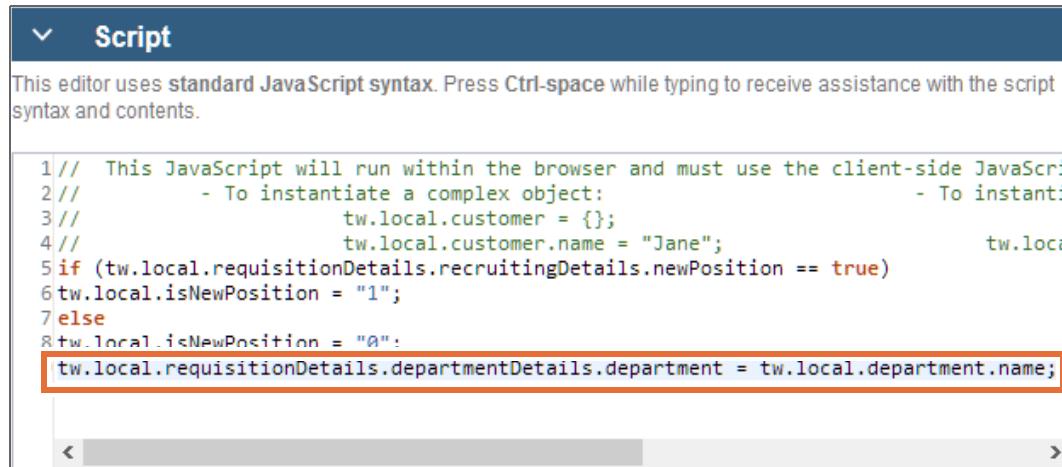
### Information

Because the control uses the `.name` variable for the **Display Name Property** and the `.value` of the **Value** property settings by default, you do not have to set those settings. If you are using a variable other than a `NameValuePair`, you must assign the values that populate the name and value of the control.

- 
- \_\_\_ c. Save your work.
8. The Department selection that is stored in the private `department (NameValuePair)` variable must be mapped to the parameter of the business object. Bind the selection value back to the business object parameter.
- \_\_\_ a. In the **Hiring Form** service, click the **Diagram** tab.
  - \_\_\_ b. Change the name of the **Map New Position Var** script to: `Map Vars`

- \_\_\_ c. In the **Script** properties menu, add the following code to the bottom of the script:

```
tw.local.requisitionDetails.departmentDetails.department =
tw.local.department.name;
```



The screenshot shows a code editor window titled "Script". It contains the following JavaScript code:

```

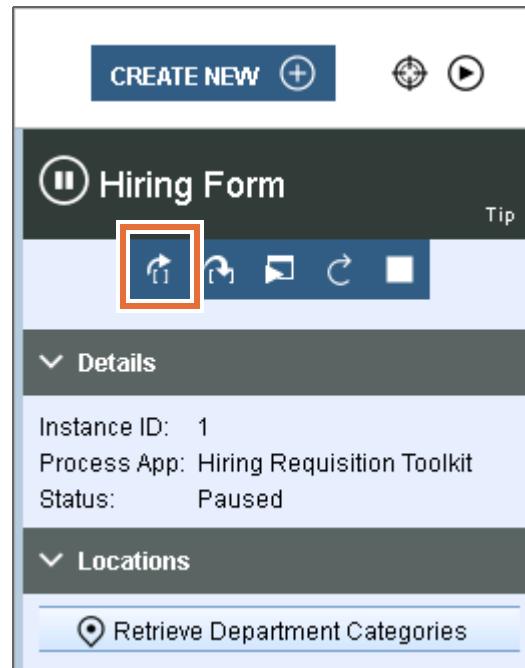
1 // This JavaScript will run within the browser and must use the client-side JavaScript
2 // - To instantiate a complex object: - To instanti
3 // tw.local.customer = {};
4 // tw.local.customer.name = "Jane"; tw.loca
5 if (tw.local.requisitionDetails.recruitingDetails newPosition == true)
6 tw.local.isNewPosition = "1";
7 else
8 tw.local.isNewPosition = "0";
 tw.local.requisitionDetails.departmentDetails.department = tw.local.department.name;

```

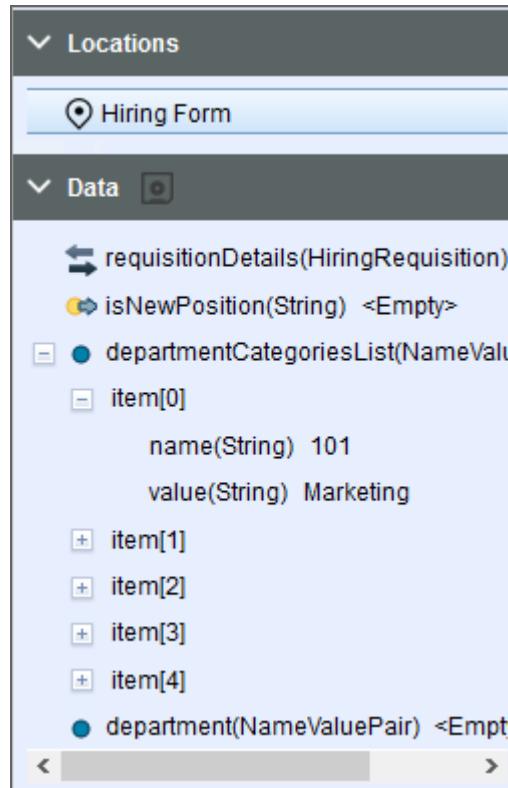
The last line of the code, `tw.local.requisitionDetails.departmentDetails.department = tw.local.department.name;`, is highlighted with a red rectangular box.

Optionally, you can copy the script from the `Script4.txt` file that is located in the `C:\labfiles\Support files\Ex8` folder

- \_\_\_ d. Save your work.
- \_\_\_ 9. Debug the coach, and verify that the **Department** single select works and binds the data to the variable.
- \_\_\_ a. Click the **Debug** icon.
- \_\_\_ b. When the debug window opens, minimize the new window.
- \_\_\_ c. Click **Step over**.



- \_\_\_ d. Verify that the `departmentCategoriesList` variable is populated with data from the integration that you created in the last exercise.



- \_\_\_ e. Click **Step over** to show the coach. The coach is displayed in the Playback window.  
 \_\_\_ f. Click the **Department Details** tab. The database retrieval service now drives the selections that are presented to the user.

| Marketing | Finance | Engineering | Professional Services | HR |
|-----------|---------|-------------|-----------------------|----|
| Marketing | Finance | Engineering | Professional Services | HR |

- \_\_ g. In the **Department** field, select **HR** and click **Submit**.

Requisition Details Position Details Recruiting

Department Details

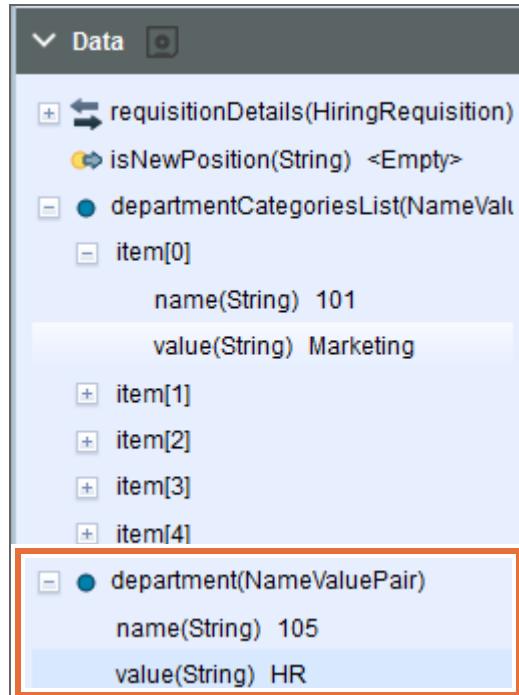
Department

HR

Division

Submit

- \_\_ h. Verify that the data is bound to the variable inside the debugger.



- \_\_\_ i. Click **Step Over**. Verify that the data was mapped into the `requisitionDetails.departmentDetails.department` variable. The `Map Vars` script stores the key for the object, not the value.

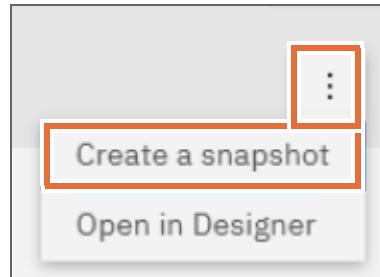


- \_\_\_ j. Close the **Debug Hiring Form** browser window.
- \_\_\_ 10. Update the process application HR Recruitment Processes dependency to the most recent version of the Hiring Requisition Toolkit.
  - \_\_\_ a. Return to Process Designer and click **Return to Workflow Center**.

- \_\_ b. Click **Details** on the Hiring Requisition Toolkit tile.

The screenshot shows the 'Toolkits (10)' page in the IBM Workflow Center. At the top, there are search and filter options ('Recently updated', 'All'), an 'Import' button, a 'Create' button, and a user profile for 'author1'. Below this, a message says 'Displaying 10/10 projects'. There are three toolkit tiles: 'Hiring Requisition Toolkit', 'UI Toolkit', and 'SAP Guided Workflow (deprecated)'. The 'Hiring Requisition Toolkit' tile is highlighted with a red box around its 'Details' button. The tile contains a wrench and screwdriver icon, the toolkit name, a brief description ('This toolkit contains all common artifacts that are associated with the HR Recruitment Processes.'), and a 'Details' button.

- \_\_ c. Select the **Snapshots** tab.  
 \_\_ d. Open the list of options by clicking the **three dots** on the right of the **Current** snapshot tile and select **Create a snapshot**.



- \_\_ e. Name the snapshot **Playback 2** and provide the following documentation: This snapshot includes integrations. Click **Create**.  
 \_\_ f. Click the navigation icon in the upper left and select **Process apps**.

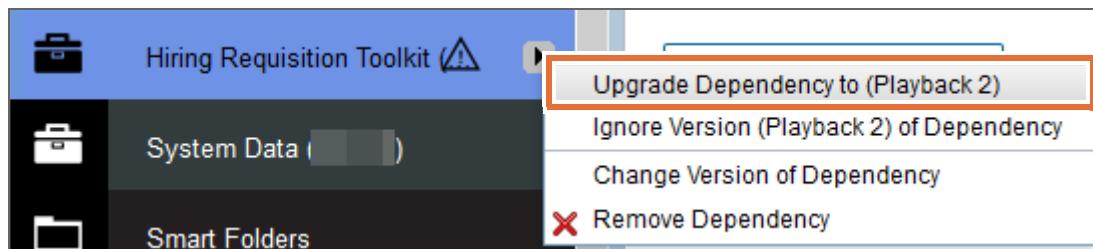


### Troubleshooting

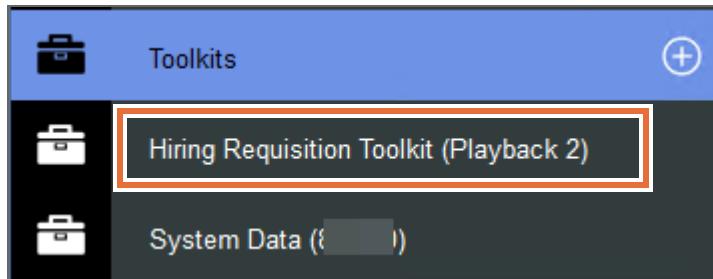
If the Process apps list does not display and still displays the Toolkits, wait a minute for the refresh to occur.

- \_\_ g. Open the **HR Recruitment Process** process application.  
 \_\_ h. Expand the **Toolkits** category in the library.

- \_\_\_ i. Click the arrow next to **Hiring Requisition Toolkit (Baseline)** and click **Upgrade Dependency to Playback 2**.



- \_\_\_ j. The dependency is now upgraded to Playback 2.



- \_\_\_ k. Playback 2: Integrations, is now complete.

- \_\_\_ l. Click **Return to Workflow Center**.

You have completed this exercise.

The first part of this exercise looked at creating a decision service. You created rules in Business Action Language (BAL) to support the service and then implemented the new service in a process. Next, you created a message start event as a trigger to run the process. You organized assets with tags. Finally, you built a query service to read information from a database. The query service is built with environment variables and exposed process variables. You then implemented the query service as a single-select control on the user interface.

## End of exercise

---

# Exercise 9. Playback 3: Handling errors and deploying your process application

## Estimated time

01:00

## Overview

This exercise covers how to implement error handling in a service.

## Objectives

After completing this exercise, you should be able to:

- Harden a service with a catch exception component
- Create a snapshot for deployment

## Introduction

In this exercise, inside the Process Designer service diagram, you drag a Catch Exception and attach it to a generic service or a human service. You then create a flow from the Catch Exception. Finally, you create a snapshot for deployment.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Implement exception handling in a service](#)

[Part 2: Prepare for final snapshot](#)

[Part 3: Check the history of the process application](#)

[Part 4: Create a snapshot of the process application](#)

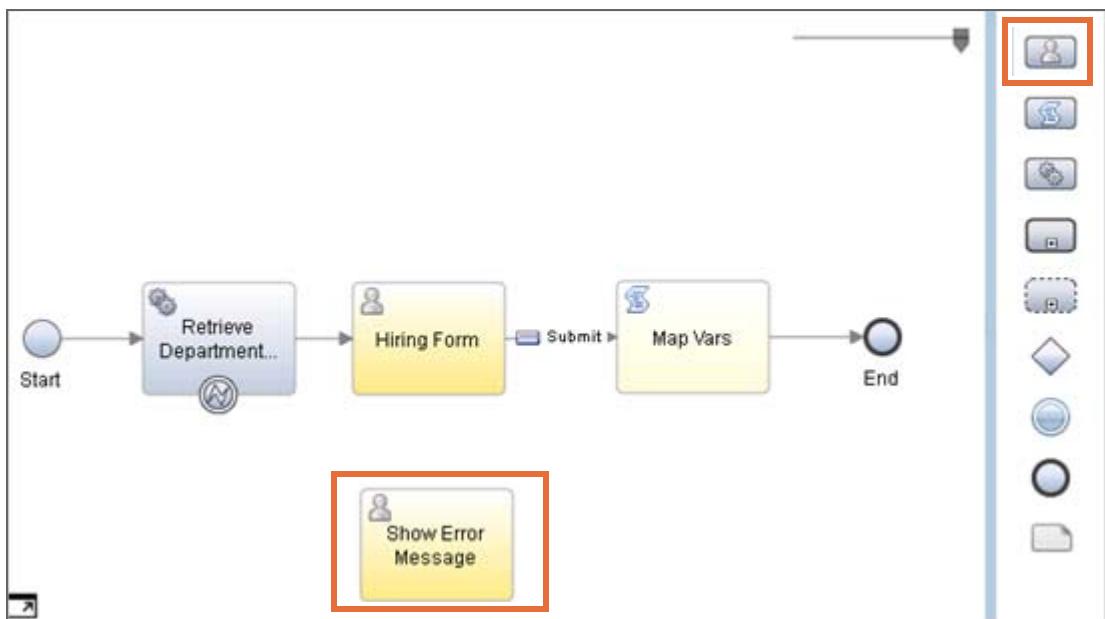
## Exercise instructions

### Part 1: Implement exception handling in a service

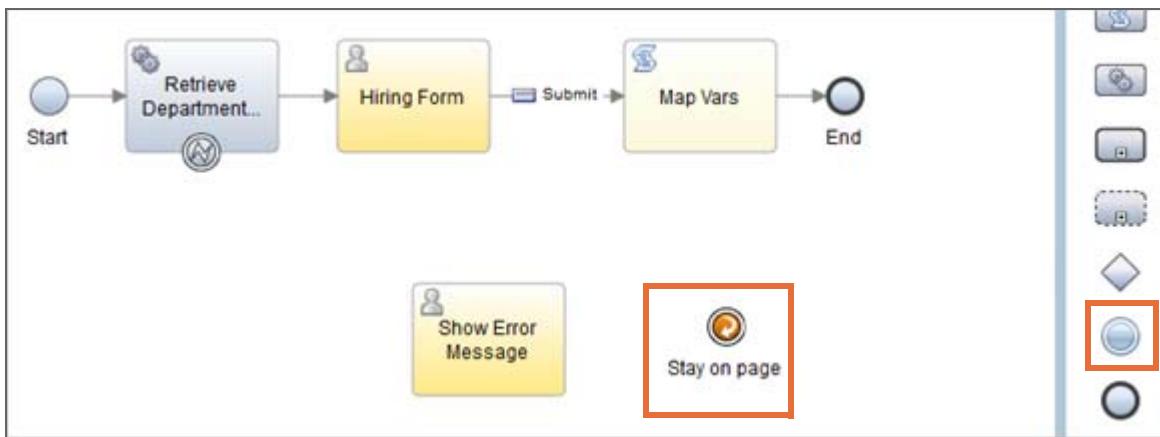
- \_\_\_ 1. Add a Catch Exception event to the Hiring Form service.
- \_\_\_ a. Open the **Hiring Requisition Toolkit**.  
Be sure to open the toolkit and not the process application.
- \_\_\_ b. In the Hiring Requisition Toolkit, open the **User Interface > Hiring Form** human service.
- \_\_\_ c. On the **Diagram** tab, drag an **Intermediate Event** from the service palette on to a control point on the Retrieve Department Categories service.



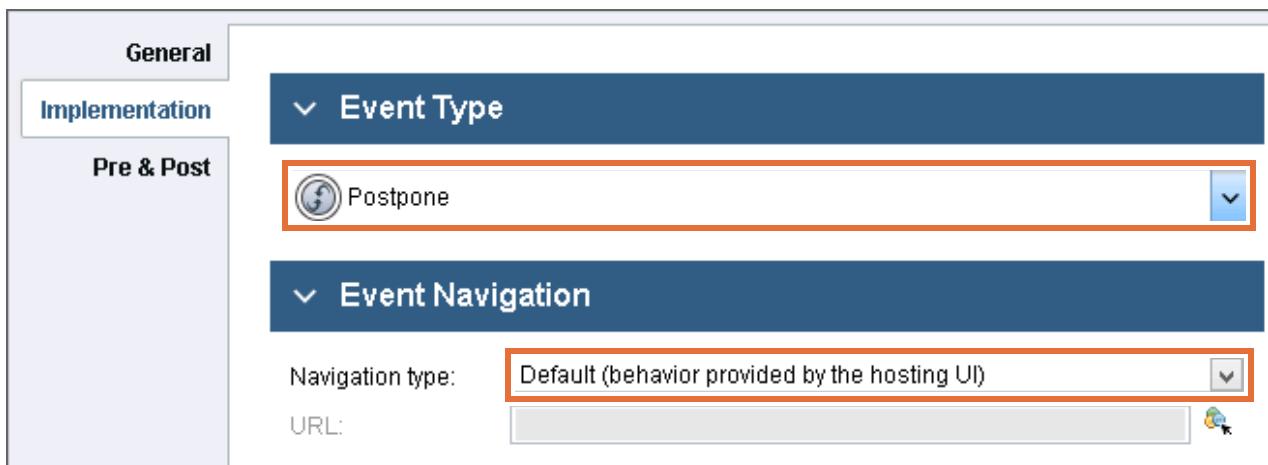
- \_\_\_ 2. Add a coach to display an error message.
- \_\_\_ a. Drag a **Coach** from the service palette onto the canvas. Rename the coach: Show Error Message



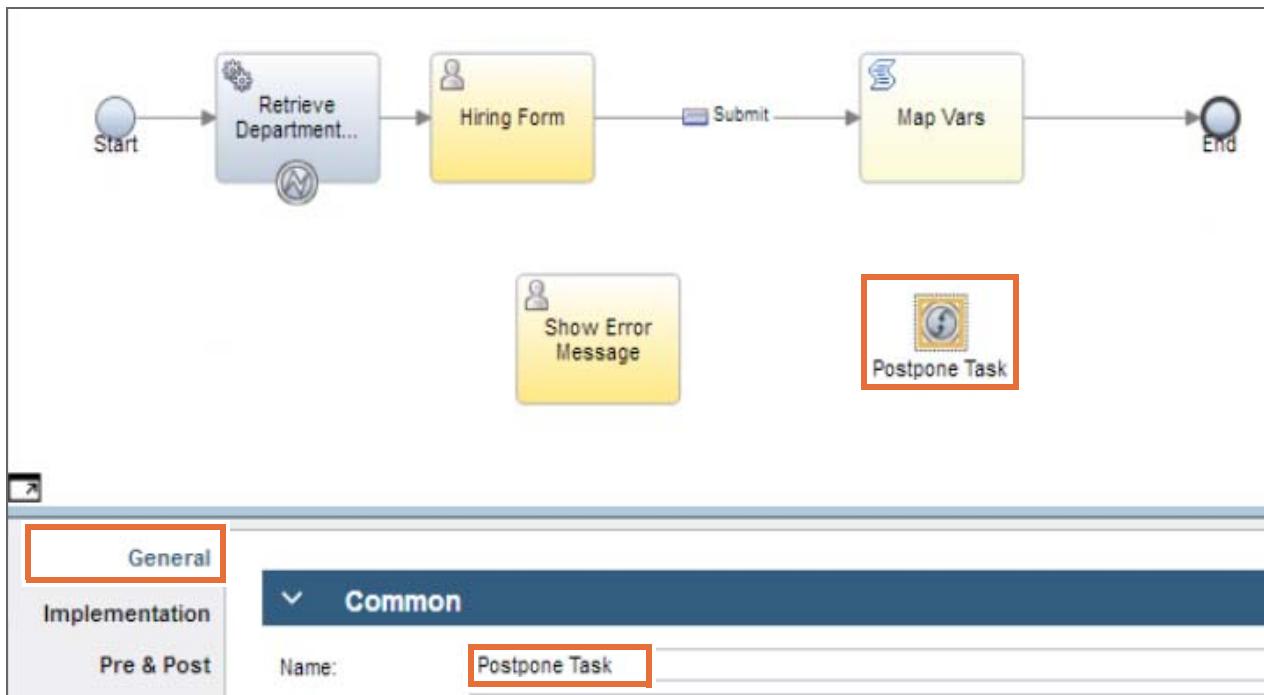
- \_\_\_ 3. Add a postpone task service.
- \_\_\_ a. Drag an **Intermediate Event** from the palette to the right of the **Show Error Message** coach.



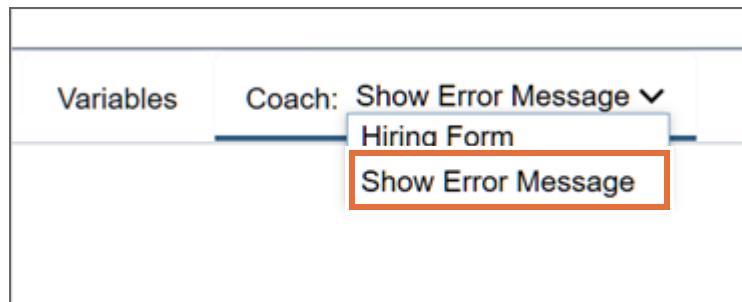
- \_\_\_ b. With the new Intermediate Event selected on the canvas, open the **Implementation** properties menu. Change the Event Type to **Postpone**. Leave the Event Navigation as **Default (behavior provided by the hosting UI)**.



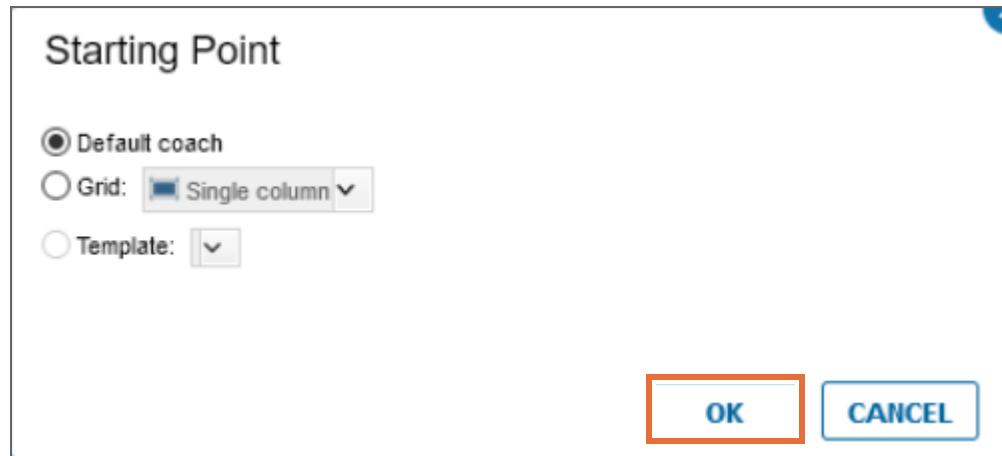
- \_\_\_ c. Open the **General > Common** properties section and rename the new event: Postpone Task



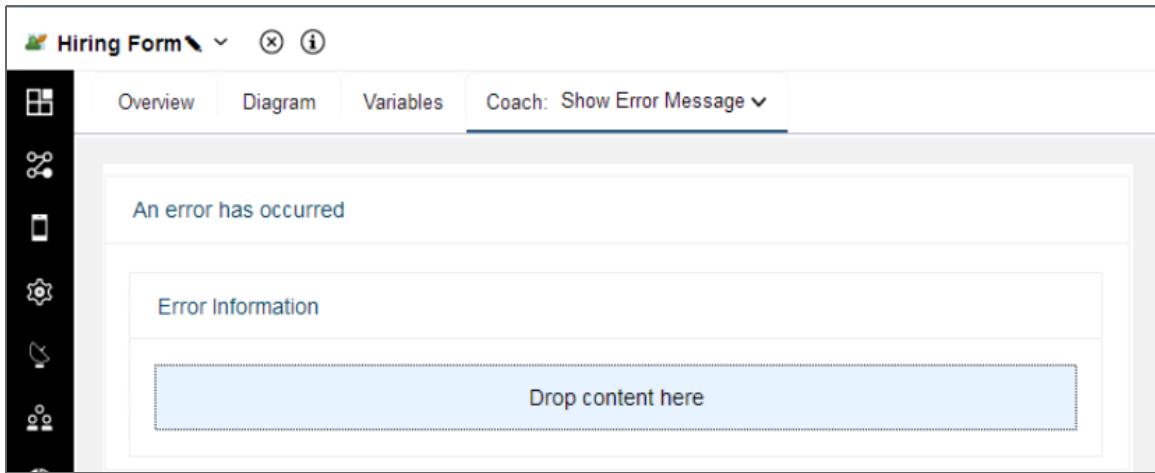
- \_\_\_ d. Save your work.
- \_\_\_ 4. Create the error message coach.
- \_\_\_ a. Click the **Coaches** tab and select the **Show Error Message** coach.



- \_\_ b. In the **Starting Point** window, ensure that the **Default coach** is selected, and click **OK**.



- \_\_ c. Select **Layout** in the **Building blocks** panel and select **Panel**  
 \_\_ d. While holding down the left mouse button, drag the **Panel** above the **OK** button. Let go of the mouse button when the top border displays a gold highlight  
 \_\_ e. Enter **An error has occurred** for the name.  
 \_\_ f. Add another panel by dragging a panel to the location on the existing panel that displays **Drop content here** and label the panel **Error information**

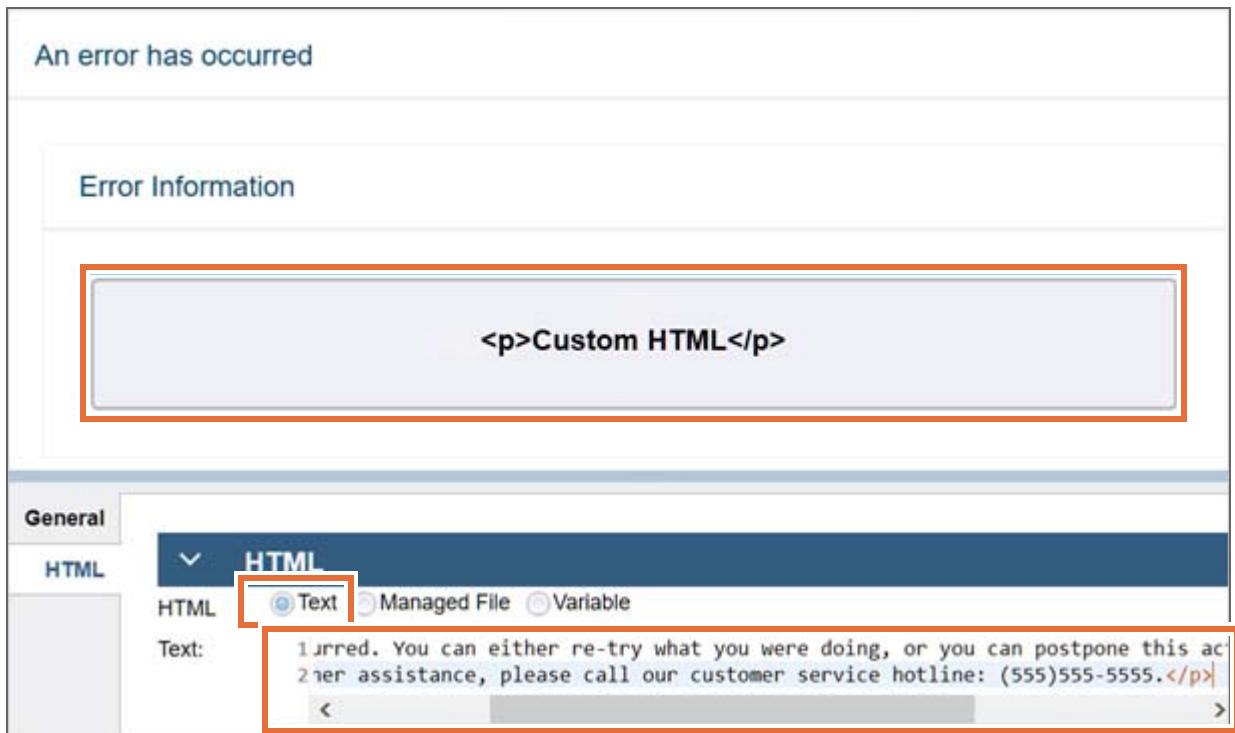


- \_\_ g. Select **Uncategorized** in the **Building blocks** panel and select **Custom HTML**  
 \_\_ h. While holding down the left mouse button, drag the **Custom HTML** control and let go of the mouse button over the section that displays **Drop content here**.

- \_\_\_ i. Under the **HTML** properties menu, select **Text** and add error information to the HTML text block that is provided to the user. Alternatively, you can copy and paste the text that is provided in the file `Script1.txt` at the location: `C:\labfiles\Support files\Ex9`

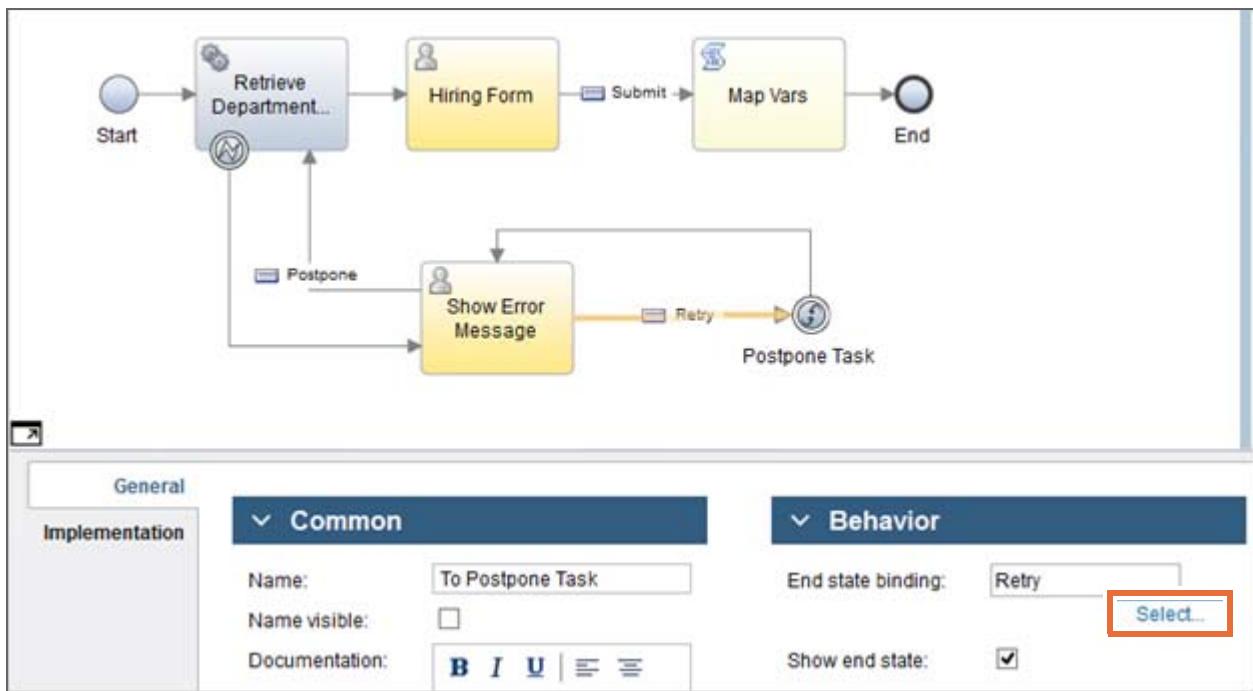
`<p>An error has occurred. You can either retry what you were doing, or you can postpone this activity until a later date.</p>`

`<p>If you need further assistance, please call our customer service hotline: (555)555-5555.</p>`

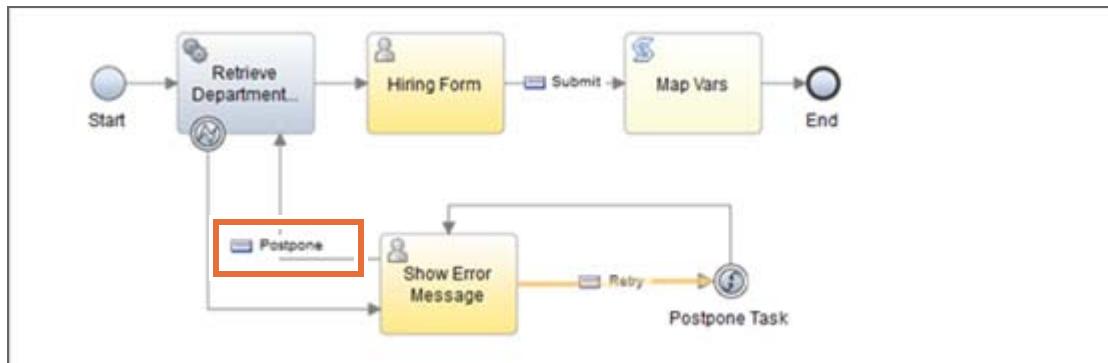


- \_\_\_ j. Select **Input** in the **Building blocks** panel and select **Button**
- \_\_\_ k. While holding down the left mouse button, drag the **Button** below the **OK** button. Let go of the mouse button when the bottom border displays a gold highlight.
- \_\_\_ l. Change the **OK** button label to: **Retry**.
- \_\_\_ m. Change the second button label to: **Postpone**.
- \_\_\_ n. Save your work.
- \_\_\_ 5. Reconnect the flow on the **Hiring Form** canvas.
- \_\_\_ a. Click the **Diagram** tab.
- \_\_\_ b. Connect the new elements on the canvas. First, connect the **Error Intermediate Event** event to the **Show Error Message** coach.
- \_\_\_ c. Next, connect the **Show Error Message** coach to the **Postpone Task** event.
- \_\_\_ d. Third, connect **Postpone Task** back to **Show Error Message**.

- e. Finally, connect **Show Error Message** to **Retrieve Department Categories** by the **Retry** control.



- f. Notice that the label on the flow is incorrect. The instructions intentionally wired the flows up to the wrong event. In this flow, the **Postpone** button retries the service while the **Retry** button postpones the task. That would be incorrect. After you connect the Show Error Message coach to other activities, the flow editor automatically uses the button controls from the coach. If the controls do not match the flow, you can either reroute the connection arrows or change the end state binding. To change the end state binding, select the **Postpone** flow.



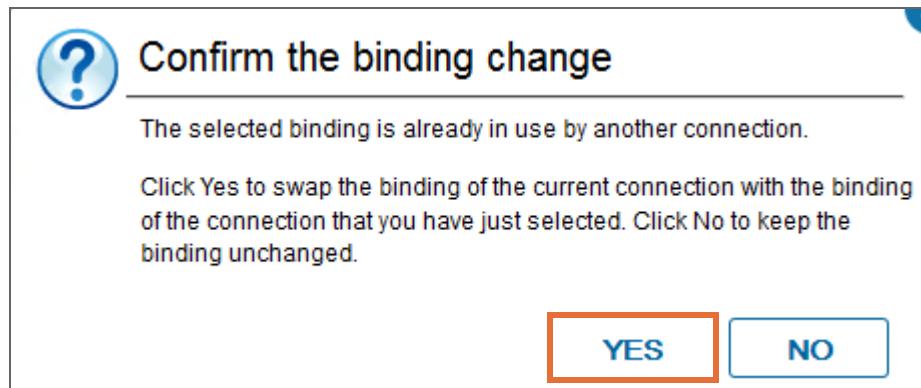
- g. In the **General > Behavior** properties section, click **Select** next to End state binding.

|                    |                                     |
|--------------------|-------------------------------------|
| Behavior           |                                     |
| End state binding: | Postpone <span>Select...</span>     |
| Show end state:    | <input checked="" type="checkbox"/> |

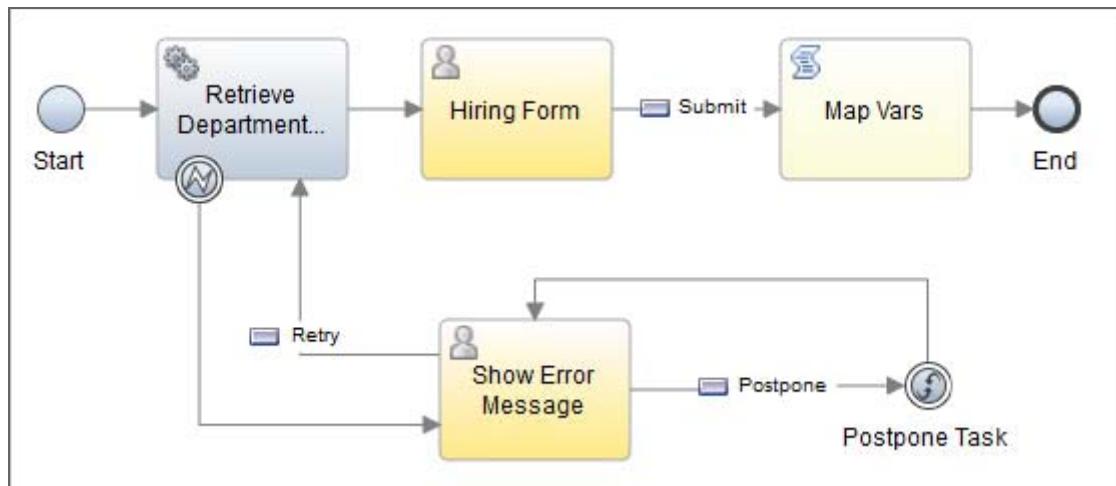
- \_\_ h. From the coach dialog box, select the **Retry** button and click **FINISH** at the bottom.



- \_\_ i. Click **Yes** to confirm the binding change.



- \_\_ j. Ensure that your buttons on your coach correspond to the correct flows on the coach.



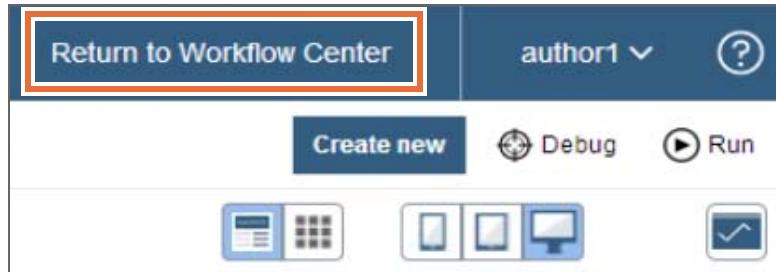
- \_\_ k. Save your work.

## Part 2: Prepare for final snapshot

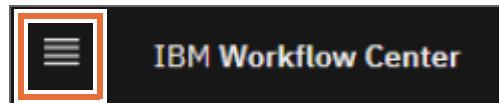
- \_\_\_ 1. Check for any validation errors in the process application.

Before taking the final snapshot of the process application for deployment, it is a good practice to check for any validation errors in the process application.

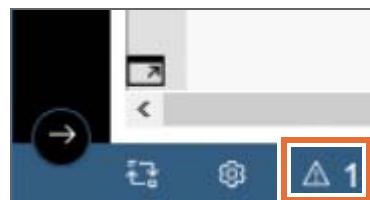
- \_\_\_ a. Click **Return to Workflow Center**.



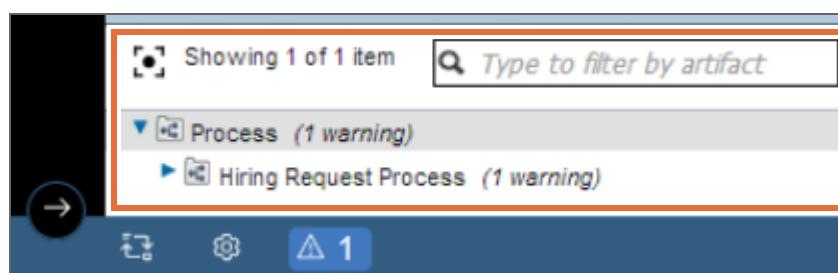
- \_\_\_ b. Click the navigation icon in the upper left.



- \_\_\_ c. Select **Process apps**
- \_\_\_ d. Open the **HR Recruitment Process** process application.
- \_\_\_ e. Validation errors must be resolved before you deploy a process application to a different environment. The instructions thus far introduced an error in the process application. You correct this error before the deployment of your process application. In this example, the Hiring Request Process contains an error. To resolve this error, in the Process Designer footer, click the **Validation errors and warnings** icon to check the name of the process that contains the error. Note that there is a number 1 next to the icon indicating one warning or error.



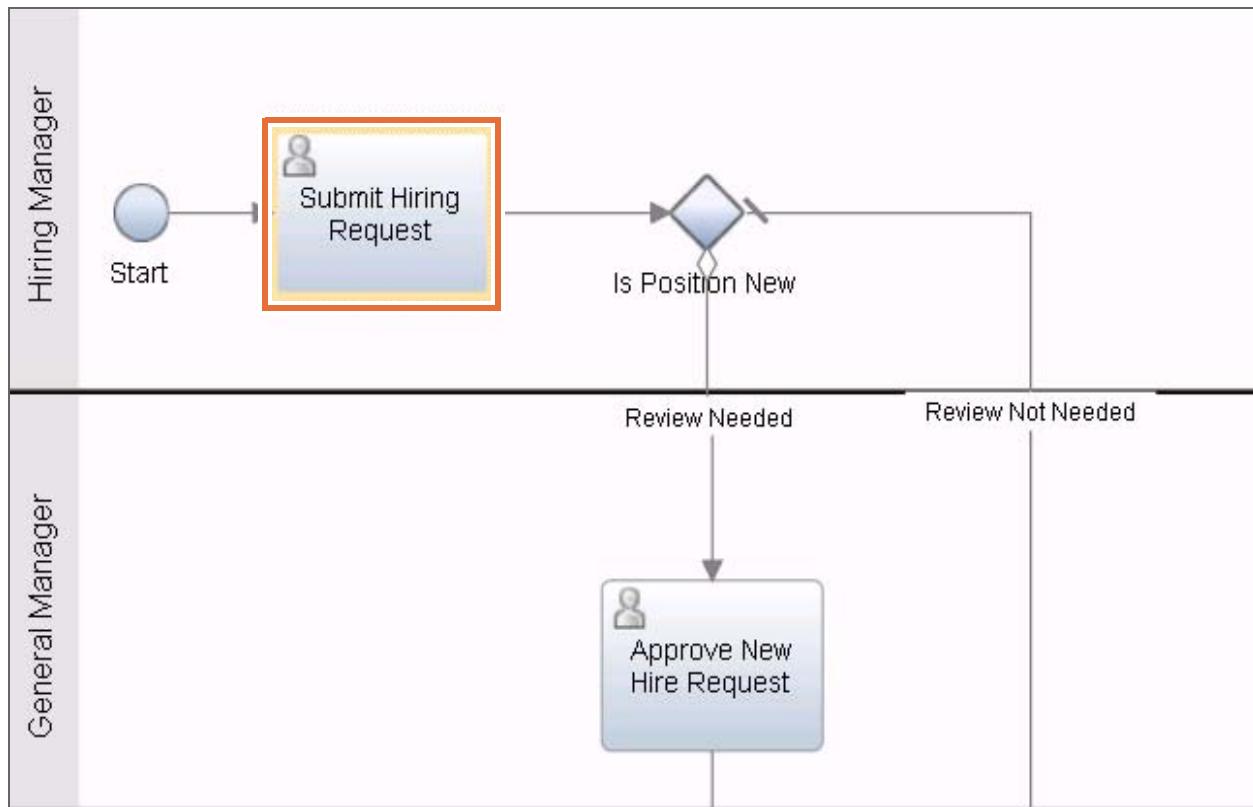
- \_\_\_ f. Clicking the **Validation errors and warnings** icon displays the process that contains the problem.



- g. Expand the **Hiring Request Process** to view the error message.

The screenshot shows a search bar at the top with the placeholder "Type to filter by artifact". Below it is a list titled "Process (1 warning)". Underneath is a single item: "Hiring Request Process (1 warning)". A tooltip below the item states: "⚠ No input parameter mapping found for parameter without default: requisitionDetails". The entire "Hiring Request Process" row is highlighted with a red border.

- h. To resolve the parameter mapping errors, click the error message to directly open **Hiring Request Process** and then click the **Submit Hiring Request** activity.



- i. Open the **Data Mapping** properties menu.  
j. In the **Input Mapping** section, map the input to the `requisitionDetails` (`HiringRequisition`) variable by typing `tw.local.requisitionDetails`.

The screenshot shows the "Input Mapping" properties menu. A text input field contains the value "tw.local.requisitionDetails", which is highlighted with a red border. To the right of the input field is a tooltip: "requisitionDetails (Hiring...)" with a small blue arrow icon pointing to the text.

**Hint**

Because the names are consistent, you can use the auto-map feature for these steps.

- \_\_\_ k. In the **Output Mapping** section, map the first output to the `requisitionDetails` (`HiringRequisition`) variable by typing `tw.local.requisitionDetails`.
- \_\_\_ l. In the **Output Mapping** section, map the second output to the `isNewPosition` (`String`) variable by typing `tw.local.isNewPosition`.

**Output Mapping**

|                                                        |   |                             |
|--------------------------------------------------------|---|-----------------------------|
| <a href="#">requisitionDetails (HiringRequisition)</a> | → | tw.local.requisitionDetails |
| <a href="#">isNewPosition (String)</a>                 | → | tw.local.isNewPosition      |

- \_\_\_ m. Select the **Approve New Hire Request** activity.
- \_\_\_ n. Open the **Data Mapping** properties menu.
- \_\_\_ o. In the **Input Mapping** section, map the input to the `requisitionDetails` (`HiringRequisition`) variable by typing `tw.local.requisitionDetails`.

**Input Mapping**

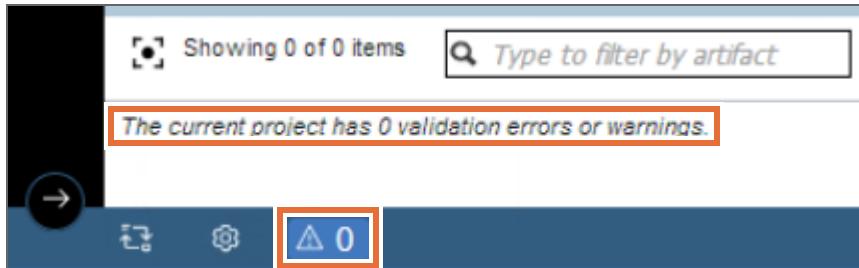
|                             |   |                                               |
|-----------------------------|---|-----------------------------------------------|
| tw.local.requisitionDetails | → | <a href="#">requisitionDetails (Hiring...</a> |
|-----------------------------|---|-----------------------------------------------|

- \_\_\_ p. In the **Output Mapping** section, map the first output to the `requisitionDetails` (`HiringRequisition`) variable by typing `tw.local.requisitionDetails`.
- \_\_\_ q. In the **Output Mapping** section, map the second output to the `isNewPosition` (`String`) variable by typing `tw.local.isNewPosition`.

**Output Mapping**

|                                                        |   |                             |
|--------------------------------------------------------|---|-----------------------------|
| <a href="#">requisitionDetails (HiringRequisition)</a> | → | tw.local.requisitionDetails |
| <a href="#">isNewPosition (String)</a>                 | → | tw.local.isNewPosition      |

- \_\_\_ r. Save your changes.
- \_\_\_ s. Verify that there are no more any errors or warnings. The Validation and errors icon indicates a zero next to it. Close the validation panel.



### **Part 3: Check the history of the process application**

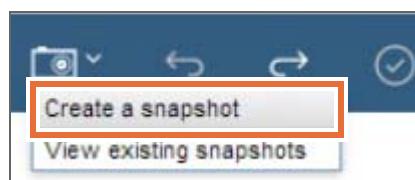
- \_\_\_ 1. Before taking a snapshot of the process application for deployment, you can check the history of the actions that you performed on the process application.
  - \_\_\_ a. Return to the Workflow Center.
  - \_\_\_ b. Click **Details** in the **HR Recruitment Process** process application tile.
  - \_\_\_ c. Click **History**.
  - \_\_\_ d. Review the set of actions that you performed on the process application.

### **Part 4: Create a snapshot of the process application**

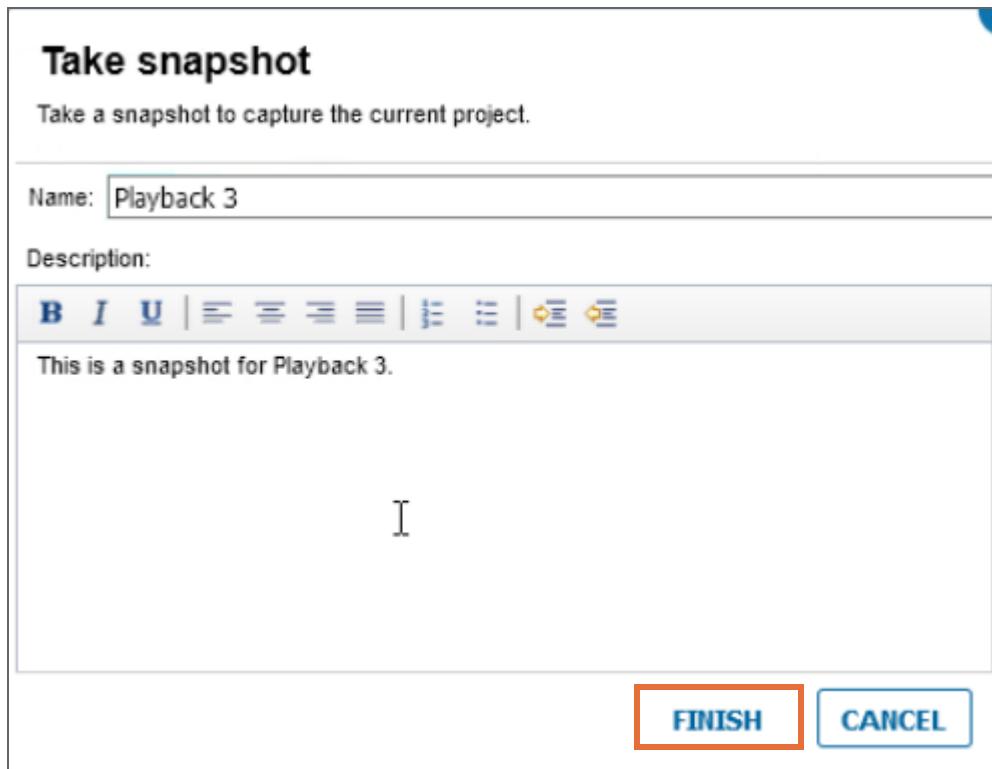
- \_\_\_ 1. Create a snapshot of the **Hiring Request Process** in IBM Process Designer.
  - \_\_\_ a. Click the **Back to process Apps** link.

The screenshot shows the "HR Recruitment Process" application in the Workflow Center. The top navigation bar includes "IBM Workflow Center" and a "Back to process apps /" link, which is highlighted with a red box. The main content area displays the "HR Recruitment Process" title. Below the title is a navigation bar with tabs: Overview, Snapshots, Permissions, History (which is underlined in blue), and Comments. The "History" tab is active, showing a single entry: "History ⓘ".

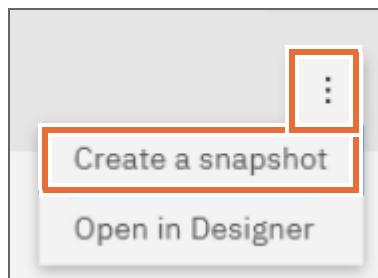
- \_\_\_ b. Open the **HR Recruitment Process** process application.
- \_\_\_ c. Expand the **Snapshot** menu on the top of the Process Designer window. Select **Create a snapshot**.



- d. Give your snapshot the name **Playback 3** and description: This is a snapshot for Playback 3. Click **FINISH**.



- e. Click **OK** when the pop-up window is shown.
- f. Return to **Workflow Center**.
- g. Click **Details** in the **HR Recruitment Process** process application tile.
- h. Select the **Snapshots** tab
- i. Verify your snapshot is displayed in the list of snapshots.
2. Create a snapshot of the new toolkit and upgrade the dependency to the new snapshot.
- a. Click the navigation icon in the upper left and expand the **Process apps** section.
- b. Click **Toolkits**.
- c. Click **Details** on the Hiring Requisition Toolkit tile.
- d. Select the **Snapshots** tab.
- e. Open the list of options by clicking the **three dots** on the right of the **Current** snapshot tile and select **Create a snapshot**.



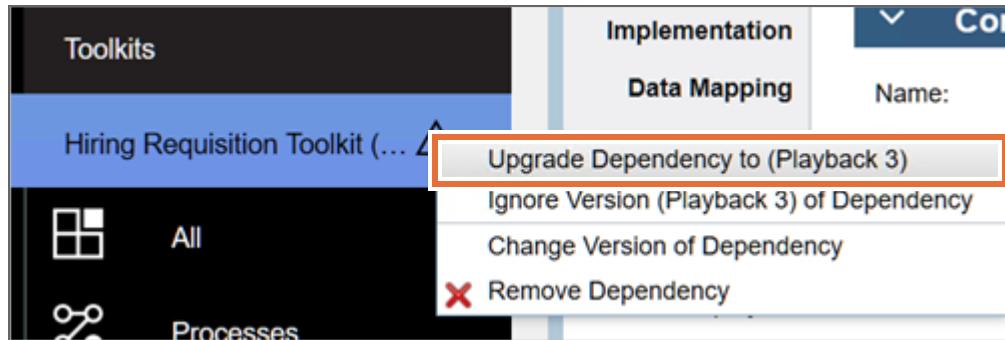
- \_\_\_ f. Name the snapshot Playback 3 and provide the following documentation: This snapshot includes error handling for a service. Click **Create**.
- \_\_\_ g. Open the menu in the top left and select **Process apps**.



## Troubleshooting

If the Process apps list does not display and still displays the Toolkits, wait a minute for the refresh to occur.

- \_\_\_ h. Open the **HR Recruitment Process** process application in Process Designer.
- \_\_\_ i. Click **Toolkits** in the library. Click the arrow next to **Hiring Requisition Toolkit** and click **Upgrade Dependency to (Playback 3)**.



- \_\_\_ j. Playback 3: Creating error handling for a service is now complete.

You have completed this exercise.

In this exercise, you added a Catch Exception component to capture and manage errors in the process. If any validation errors were present in the process application, you resolved any problems. Next, you looked at the history of the actions that were performed on the process application and created the snapshot for deployment.

## End of exercise

# Exercise 10. Implementing a custom routing solution

## Estimated time

01:00

## Overview

In this exercise, you learn how to implement a custom routing solution.

## Objectives

After completing this exercise, you should be able to:

- Implement custom routing

## Introduction

The company is working hard to meet a deadline and some employees are working on the weekend. General Managers who approve the new hire request do not work on the weekend. Those designated as Weekend Managers should receive work if the date of the request is on the weekend.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Add code to determine whether it is the weekend](#)

[Part 2: Add custom routing to the HR Recruitment process](#)

[Part 3: Test the custom routing solution](#)

## Exercise instructions

### Part 1: Add code to determine whether it is the weekend

Before adding a custom routing option to the **Approve New Hire Request**, the determination of whether the date falls on a weekend needs to be added. To do this, you add a Boolean variable and JavaScript to the **Hiring Form**.

- 1. Add a Boolean variable
  - a. Open the **Hiring Requisition Toolkit** toolkit
  - b. Open the **Hiring Form** user interface
  - c. Under the **Variables** tab, click the **plus sign (+)** next to Output.
  - d. Enter `isWeekend` for the name
  - e. Click **Select** next to Variable type and select **Boolean** for the type.
  - f. Save your work



- 2. Enter the JavaScript code to determine if it is the weekend
  - a. Click the **Diagram** tab to switch to the diagram
 

Because there is already a script task before the End event, you can place the JavaScript in this task.
  - b. Highlight the **Map Vars** script task and select the **Script** tab under the Properties section.

- \_\_\_ c. Enter the following JavaScript at the end of the existing script block

```
//Determine if request date is on the weekend
var dt = new Date(tw.local.requisitionDetails.dateOfRequest);
if(dt.getDay() == 6 || dt.getDay() == 0) {
 tw.local.isWeekend = true;
} else {
 tw.local.isWeekend = false;
}
```

Alternatively, you can paste the script from the file Script1.txt under the folder  
C:\labfiles\Support files\Ex10

- \_\_\_ d. Verify your JavaScript code

```
1// This JavaScript will run wi
2// thin the browser and must use the client-side JavaScript syntax supported by the browser. For example,
3// - To instantiate a complex object: - To instantiate a list:
4// tw.local.customer = {}; tw.local.addresses = [];
5// tw.local.customer.name = "Jane"; tw.local.addresses[0] = {};
6if (tw.local.requisitionDetails.recruitingDetails newPosition == true)
7 tw.local.isNewPosition = "1";
8else
9 tw.local.isNewPosition = "0";
10tw.local.requisitionDetails.departmentDetails.department = tw.local.department.name;
11
12//Determine weekend
13var dt = new Date(tw.local.requisitionDetails.dateOfRequest);
14if(dt.getDay() == 6 || dt.getDay() == 0) {
15 tw.local.isWeekend = true;
16} else {
17 tw.local.isWeekend = false;
}
```

- \_\_\_ e. Save your work

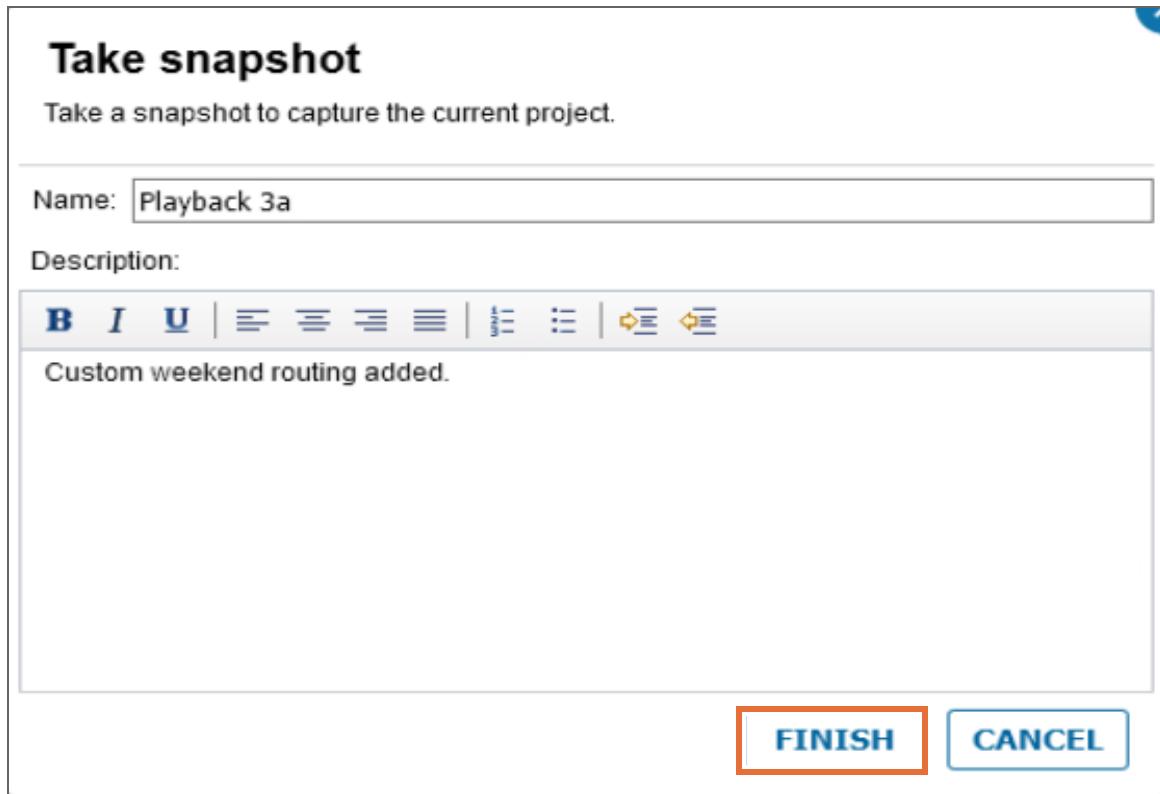
- \_\_\_ 3. Save a new snapshot

This was a last-minute addition. As such, it is considered part of the Playback 3 solution.

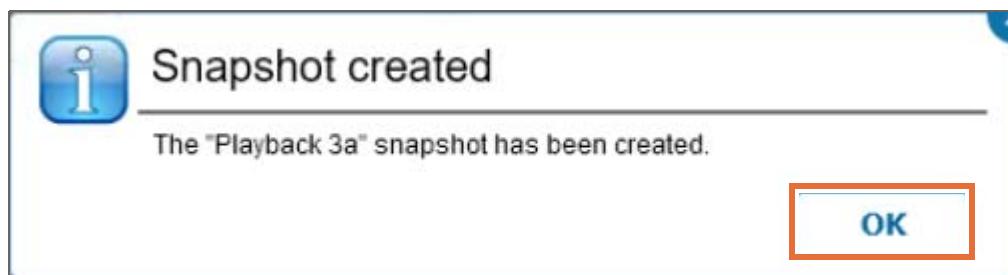
- \_\_\_ a. Click the **camera icon** in the upper right corner and select **Create a snapshot**.



- \_\_\_ b. Enter Playback 3a as the name, enter a description, and click **FINISH**



- \_\_\_ c. Click **OK** on the Snapshot Created dialog.



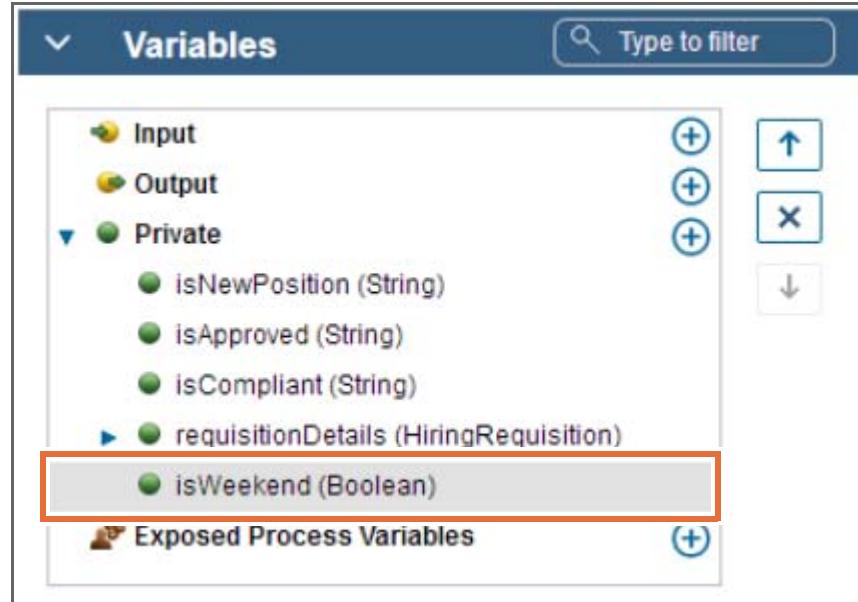
- \_\_\_ d. Click **Return to Workflow Center**.

## **Part 2: Add custom routing to the HR Recruitment process**

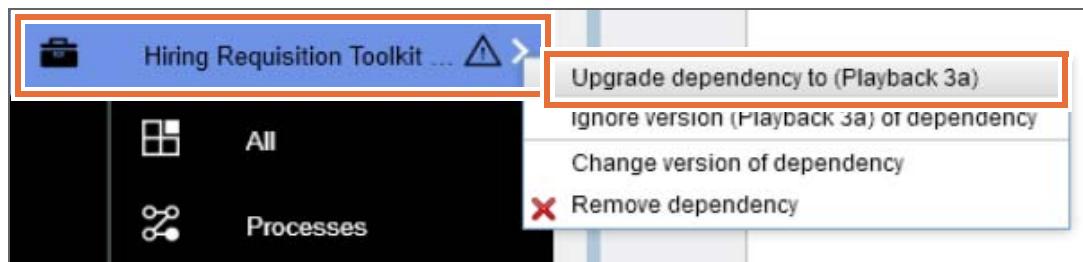
In this section, you update **Approve New Hire Request** activity to use custom routing. Before you do that, you add the variable to the process, perform data mapping, and add the **Weekend Managers** team. Before being able to use the updated **Hiring Form**, you must upgrade the dependency to the new snapshot of the **Hiring Requisition Toolkit**.

- \_\_\_ 1. Add a Boolean variable to the process
  - \_\_\_ a. Open the **HR Recruitment Process** process application
  - \_\_\_ b. Open the **Hiring Request Process**
  - \_\_\_ c. Open the **Variables** tab

- \_\_\_ d. Click the **plus sign (+)** next to Private to add a variable
- \_\_\_ e. Enter `isWeekend` for the name and set the variable type to Boolean
- \_\_\_ f. Verify the addition of the private variable.



- \_\_\_ g. Save your work.
- \_\_\_ 2. Upgrade to the new **Hiring Request Toolkit** snapshot
- \_\_\_ a. Expand the **Toolkits** category in the Library section. The **Hiring Request Toolkit** shows a warning because the process application is not pointing to the most recent snapshot version.
  - \_\_\_ b. Click the **arrow** next to the Toolkit and select **Upgrade Dependency to (Playback 3a)**



- \_\_\_ 3. Perform data mapping

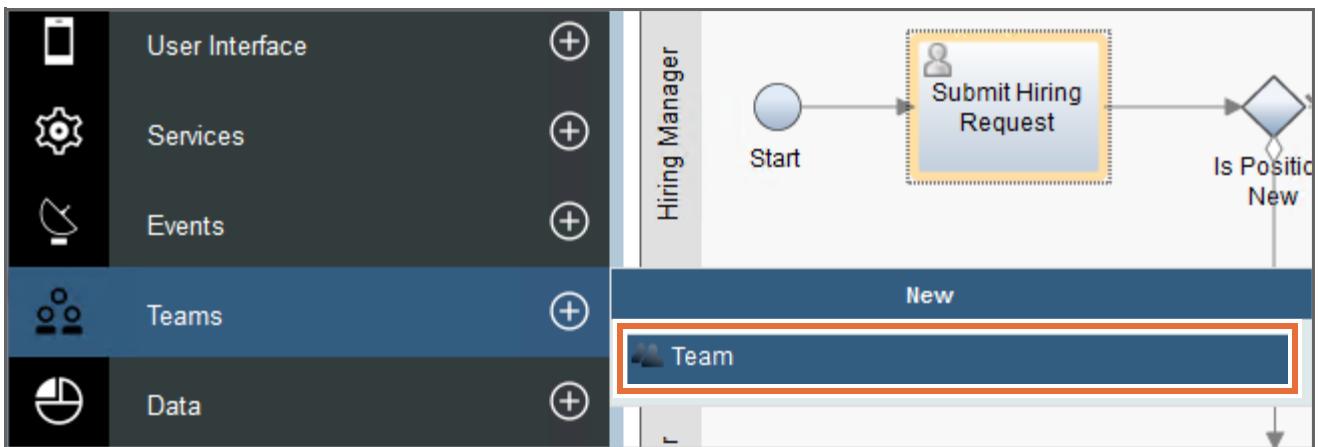
Now that the Boolean variable is added to the process and the toolkit dependency upgraded, you can map the variable from the **Submit Hiring Request** to the process.

- \_\_\_ a. From the Hiring Request Process, click the **Definition** tab.
- \_\_\_ b. Select the **Submit Hiring Request** activity
- \_\_\_ c. Select the **Data Mapping** tab in the properties section below

- \_\_ d. Under the Output Mapping section, click the **Variable Picker** next to the `isWeekend` variable and select **isWeekend (Boolean)**



- \_\_ e. Save your work
- \_\_ 4. Add the Weekend Managers team and add a member
- The Weekend Managers team contains the managers that are able to work on weekends.
- \_\_ a. In the process library, click the (+) plus sign next to **Teams**. Click **Team**.



- \_\_\_ b. Name the team **Weekend Managers** and click **FINISH**.

**New Team**

Teams represent the teams of users in your enterprise that perform tasks. After you create a team, you can add users or teams as members, or define an expression to determine membership.

Name: Weekend Managers

**FINISH**   **CANCEL**

- \_\_\_ c. In the **Members** section, click the (+) plus sign next to **Users**.

**Common**   **Behavior**

Name: Weekend Managers  
Modified: author1 (Aug 27, 2020, 4:45:43 PM)

Documentation:

Specify members by using:  
 Users or groups  
 A service

**Members**

Users   **(+)**

User Groups   **(+)**

- \_\_\_ d. Select **user2** from the list.

You now have a populated list in the Members section with **user2** for the **Weekend Managers** team.

**Members**

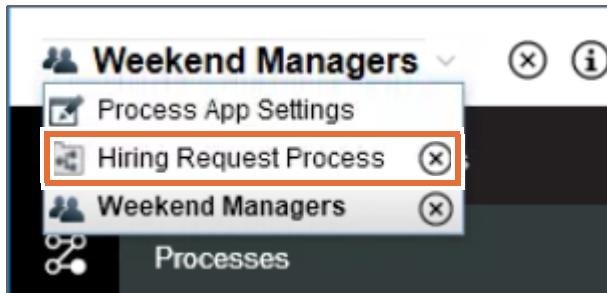
Users   **(+)**

**user2**

User Groups   **(+)**

- \_\_\_ e. Save your work.

- \_\_\_ 5. Apply custom routing to the **Approve New Hire Request** activity assignment
  - \_\_\_ a. Return to the Hiring Request Process by opening the **History** menu at the top and selecting it from the drop-down menu



- \_\_\_ b. Select the **Approve New Hire Request** activity
- \_\_\_ c. In the Properties section below, select the **Assignments** tab
- \_\_\_ d. Select **Custom** next to **Assign to**.
- \_\_\_ e. Enter the following expression in the **Expression** field
 

```
tw.local.isWeekend?"TEAM:Weekend Managers":"TEAM:General Managers"
```

 Alternatively, you can paste the script from the file `Script2.txt` under the folder  
`C:\labfiles\Support files\Ex10`

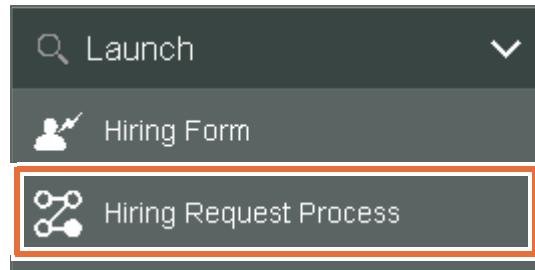
A screenshot of the 'Assignments' configuration dialog. The title bar says 'Assignments'. There are four input fields: 'Assign to:' set to 'Custom', 'Expression:' containing the script `tw.local.isWeekend?"TEAM:Weekend Managers":"TEAM:General Managers"`, 'User distribution:' set to 'None', and 'Experts team:' set to 'General Managers'. Below these fields are 'Select...', 'New...', and 'X' buttons.

- \_\_\_ f. Save your work

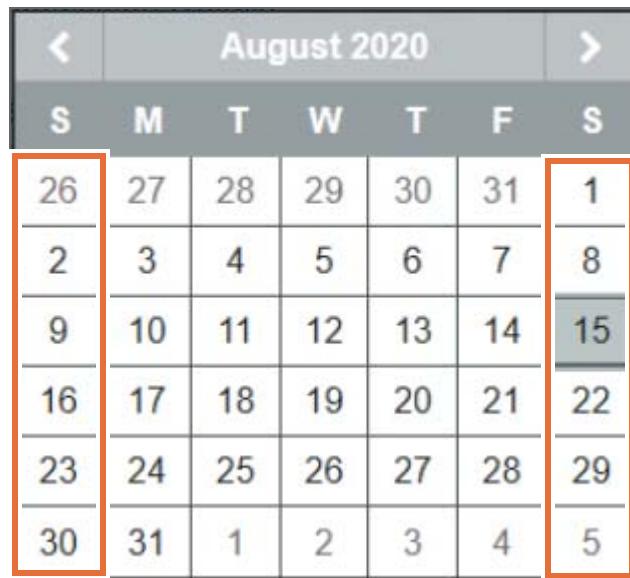
### **Part 3: Test the custom routing solution**

- \_\_\_ 1. Submit a new request
  - \_\_\_ a. Open **Process Portal** in another tab.
  - \_\_\_ b. The Process Portal opens in a new browser tab. Since you were already logged in to the Workflow Center as **author1**, the session continues and you are logged in to Process Portal as **author1** also.

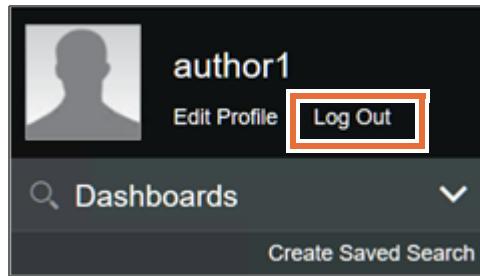
- \_\_\_ c. Create an instance of the process by clicking **Hiring Request Process** in the Launch section on the left. Wait for the first task to open in the main window.



- \_\_\_ d. To send the task to the Weekend Managers group, select a Saturday or Sunday for the **Date Of Request**



- \_\_\_ e. To ensure that the process follows the path to the **Approve New Hire Request** activity, click the **Recruiting Details** tab and place a check mark next to **New Position**  
 \_\_\_ f. Click **Submit**  
 \_\_\_ g. Log out as **author1**



- \_\_\_ 2. Verify that the weekend managers see the task

Recall that user2 was the only member of the **Weekend Managers** team.

- \_\_\_ a. Log in to Process Portal by using user2 in the **Username** field and passw0rd in the **Password** field

- \_\_\_ b. Verify the user, as a member of the Weekend Managers team, received the **Approve New Hire Request** task

The screenshot shows the Process Portal interface for User2. The top navigation bar displays "User2", "Edit Profile", and "Log Out". On the right, a summary bar shows "1 Total Open" tasks, all marked as "On Track". The main workspace contains a search bar and a list of tasks. One task is visible: "Step: Approve New Hire Request" from the "Hiring Request Process: 311" under the "Weekend Managers" team, due on Aug 27, 2020, at 6:48:38 PM. There are three small icons next to the task: a green play button, an orange flag, and a yellow exclamation mark.

- \_\_\_ c. Log out as **User2**
- \_\_\_ 3. Verify that the General Managers do not see the task  
Recall that user1 was the only member of the **General Managers** team.
- \_\_\_ a. Log in to Process Portal by using user1 in the **Username** field and password in the **Password** field
- \_\_\_ b. Verify the user, as a member of the General Managers team, did not receive the **Approve New Hire Request** task.

The screenshot shows the Process Portal interface for user1. The top navigation bar displays "User1", "Edit Profile", and "Log Out". On the right, a summary bar shows "0 Total Open" tasks, all marked as "On Track". The main workspace contains a search bar and a list of tasks. No tasks are visible, indicating that user1 did not receive the task assigned to User2.

- \_\_\_ c. Log out as **user1**.

## Part 4: Clean up the environment

When demonstrating mock scenarios, it is important to reset the environment back to the defaults before moving on.

- \_\_\_ 1. Delete snapshot status changes
- \_\_\_ a. To access the administration console, go back to the Quick Start window and click the **Process Admin Console** link.

**IBM Business Automation Workflow Quick Start**

**IBM Business Automation Workflow Enterprise Quick ProcessCenter with profile DmgrProfile**

**Deployment environment administration**

Stop the Workflow Center Deployment Environment. Stop all configured clusters  
[Stop the Workflow Center Deployment Environment](#)

**Administration consoles & tools**

Manage applications, buses, servers and resources within the administrative domain. Control managed nodes and their resources.  
[WebSphere Application Server Administrative Console](#)

Administer the workflow servers in your environment, including the users and instances for each server. Manage queues and caches, and view and manage process instances.  
[Process Admin Console](#)

Work with Performance Data Warehouse queues, manage data transfer errors and overall performance.

- \_\_\_ b. When the Process Admin console is displayed, note the various links in the left pane.

**IBM | Process Admin Console**

**SingleClusterMe...**

- ▶ Workflow Admin
- ▶ User Management
- ▶ Monitoring
- ▶ Event Manager
- ▶ Admin Tools
- ▶ Alert Definitions
- ▶ Performance

The Process Admin console provides configuration and management tools for the Workflow Servers in your IBM Business Automation Workflow environment.

**Process Status Summary**

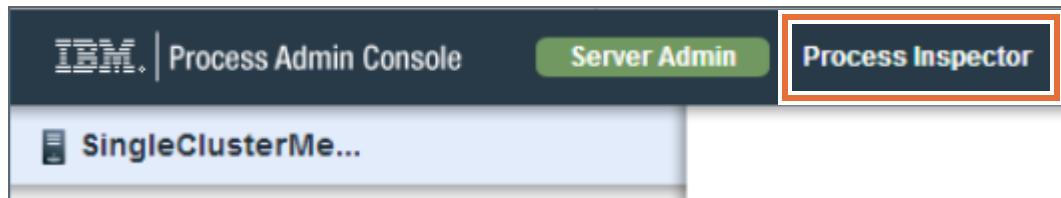
| Today | Week | Month | Year | All |
|-------|------|-------|------|-----|
| -     | -    | -     | -    | -   |

Active    Completed    Failed    Suspended

**Process Applications**

- [Heritage Process Portal \(deprecated\) \(TWP\)](#)
- [Process Portal \(SYSRP\)](#)
- [Performance \(SYSPERFDB\)](#)
- [Hiring Sample \(HSS\)](#)

- \_\_\_ c. Click the **Process Inspector** link in the navigation bar at the top.



The Process Inspector opens in a new tab.

- \_\_\_ d. Click **Search**

The **Filter process instances by** section is updated and all process applications in the repository are listed on the right with their status. The Search button changes to Refresh.

- \_\_\_ e. Highlight all the snapshot status changes and click **Delete**. The number of snapshots may be different for your environment.

The screenshot shows the Process Inspector interface. On the left, a list of process instances is displayed, each with a checked checkbox next to it. The instances are: 'Hiring Request Process:311', 'Default Snapshot Status Change:310', 'Default Snapshot Status Change:309', 'Default Snapshot Status Change:308', 'Default Snapshot Status Change:307', 'Default Snapshot Status Change:306', and 'Default Snapshot Status Change:305'. On the right, a sidebar displays the message '6 processes selected' and a 'Delete' button, which is highlighted with a red rectangular border.

- \_\_\_ f. Click **OK** on the Confirm Bulk Action window.

The snapshot status changes are deleted. Two active Hiring Request process applications are still listed. Your list might have more.

- \_\_\_ 2. Terminate and delete the active processes.

- \_\_\_ a. Highlight the remaining active process instances and click **Terminate**.

The screenshot shows the Process Inspector interface. On the left, a list of process instances is displayed, each with a checked checkbox next to it. The instances are: 'Hiring Request Process:311', 'Hiring Request Process:304', 'Advanced Employee Requisition NG (List) for Tom Miller (14)', and 'Standard Employee Requisition for Tom Miller (Standard HR Open New Position)'. On the right, a sidebar displays the message '2 processes selected' and a 'Terminate' button, which is highlighted with a red rectangular border.

- \_\_\_ b. Click **OK** on the Confirm Bulk Action window.

- The status of the processes change to Terminated.
  - c. With both terminated processes highlighted, click Delete.
  - d. Click **OK** on the Confirm Bulk Action window.
- Both processes are deleted.

You have completed this exercise.

In the first part of the exercise, you modified the Hiring Request Toolkit to add a Boolean variable and JavaScript to the Hiring Request form. Then you added the variable to the process, upgraded the toolkit dependency, and added the weekend managers team to the process. Once that was completed, you were able to configure a custom routing assignment for the Approve New Hire Request activity. Finally, to test that only Weekend Managers receive the task, you created a new request by using a weekend date. To verify that only the Weekend Managers see it, you logged in as a member of the Weekend Managers to verify you see the task. You then logged in as a member of the General Managers team to verify that you did not see the task.

## End of exercise

# Exercise 11. Implementing the “four eyes” policy by using a team filter

## Estimated time

01:30

## Overview

In this exercise, you learn how to implement a “four eyes” policy in a process by using a team filter.

## Objectives

After completing this exercise, you should be able to:

- Implement the “four eyes” policy by using a team filter

## Introduction

The company chief executive officer (CEO) is concerned about the activity in the HR Administrator lane during the approval linked process. The CEO wants to make sure that if an override is required, two people must look at the request, enabling a “four eyes” policy for activities in that lane. You must create a second activity in the lane, and use a routing policy to ensure that the user who completed the first activity in a lane does not claim the second activity in the lane.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Modify the process](#)

[Part 2: Create the team filter service](#)

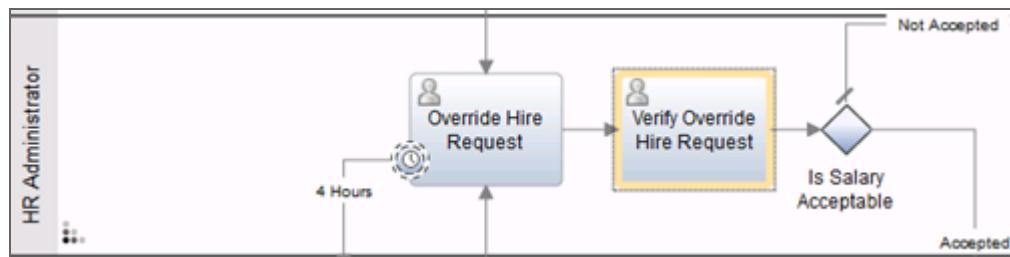
[Part 3: Apply the team filter service to the activity assignment](#)

[Part 4: Test the four eyes policy](#)

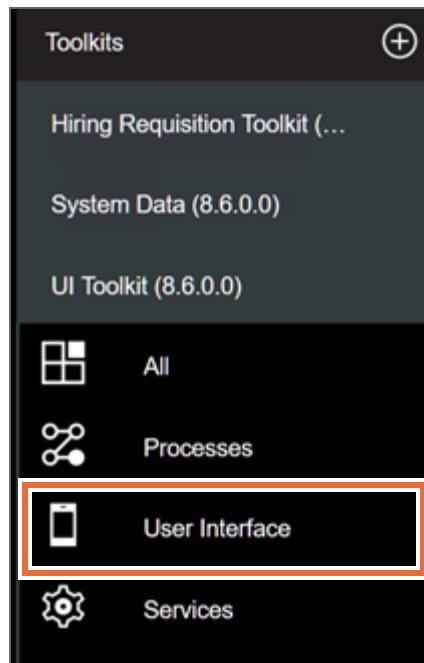
## Part 1: Modify the process

The first steps of this exercise are to change the process to meet the new requirements of the organization. After the process is changed, you will then implement the four eyes policy by using Team Filter services.

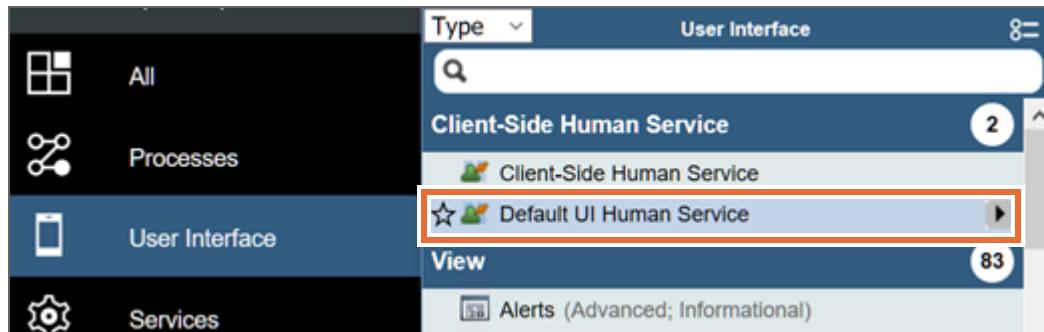
- \_\_\_ 1. Create the verify activity to make sure that if an override is required, two people must look at the request, enabling a “four eyes” policy for activities in that lane.
- \_\_\_ a. Open the **HR Recruitment Process** process application.
- \_\_\_ b. Open the **Approve Hire Request** process.
- \_\_\_ c. Drag a User Task to the right of the **Override Hire Request** activity and drop it on the sequence flow. Name this activity: Verify Override Hire Request



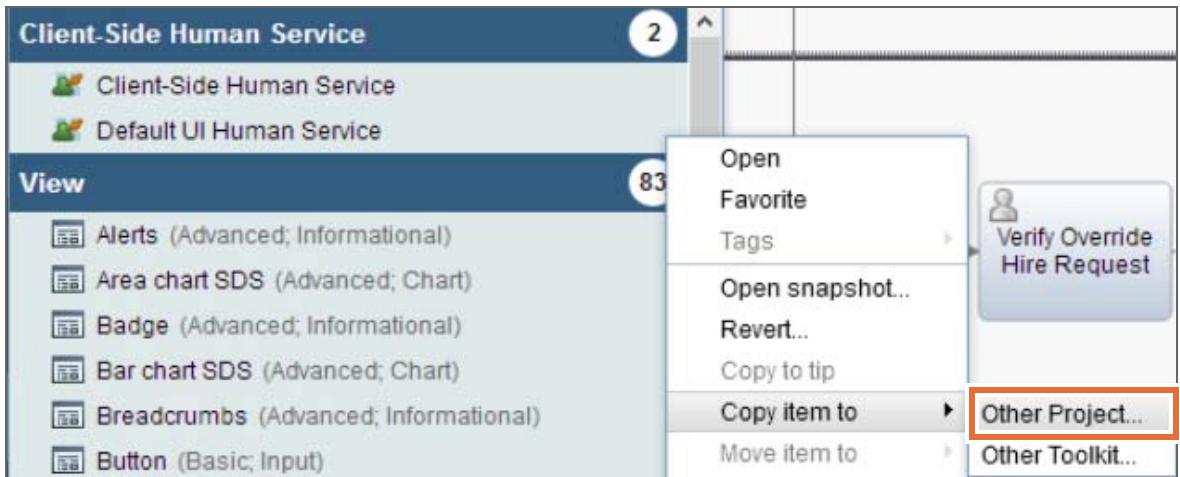
- \_\_\_ 2. Create a service to capture the username during the Override Hire Request step.
- \_\_\_ a. Expand the **Toolkits** category in the library, and then expand **UI Toolkit** and select **User Interface**.



- \_\_ b. Under the Client-Side Human Service category, right-click the **Default UI Human Service**.



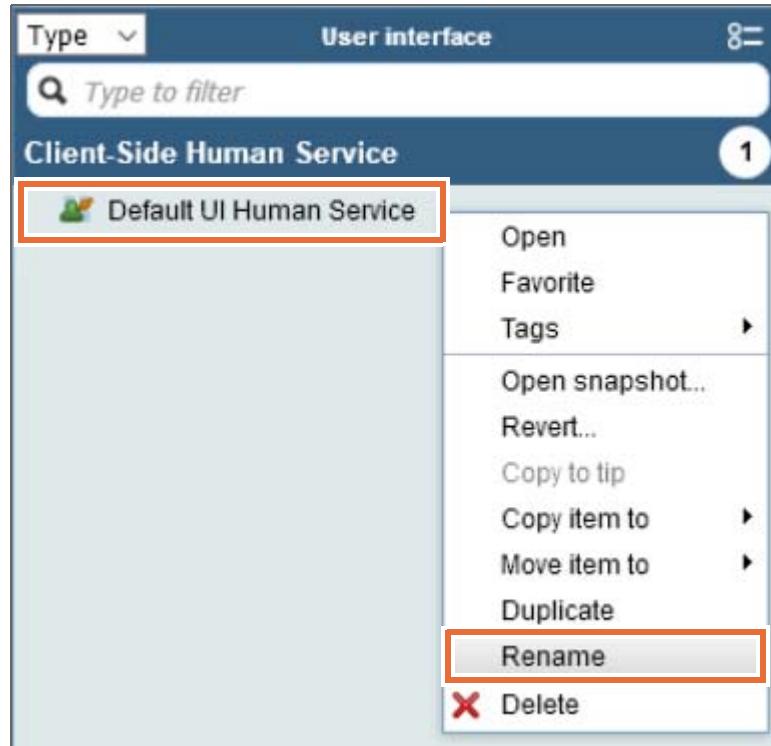
- \_\_ c. Select **Copy item to > Other Project**.



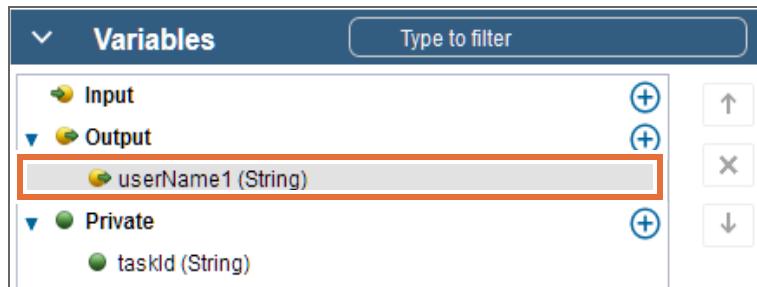
- \_\_ d. Select **HR Recruitment Process**.

- \_\_ e. Click the **Toolkits** menu to collapse the category in the library.

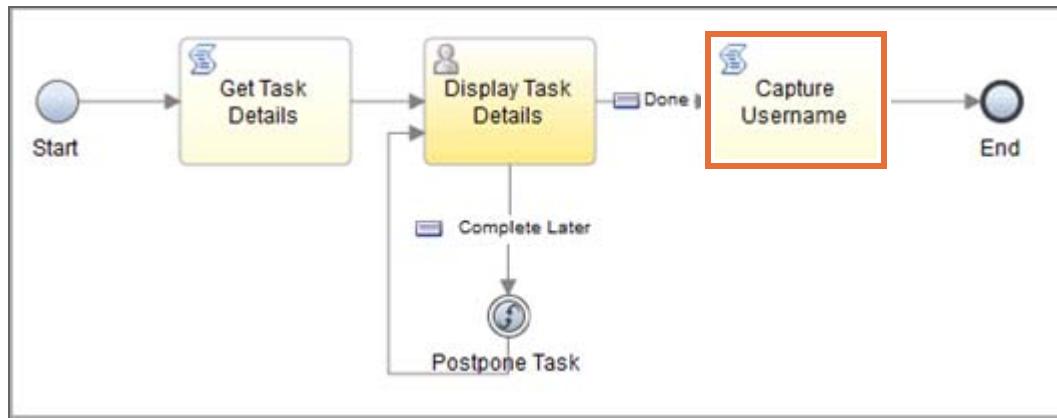
- \_\_\_ f. Scroll up in the library and this time click the **User Interface** menu in the library for the **HR Recruitment Process** process application and right-click the Default UI Human Service you just copied and select **Rename**.



- \_\_\_ g. Rename the **Default UI Human Service** to: **Override Hire Request** and click **FINISH**.  
 \_\_\_ h. Open the **Override Hire Request** human service.  
 \_\_\_ i. Click the **Variables** tab and add an output variable `userName1 (String)` to the Override Hire Request service to capture the username.

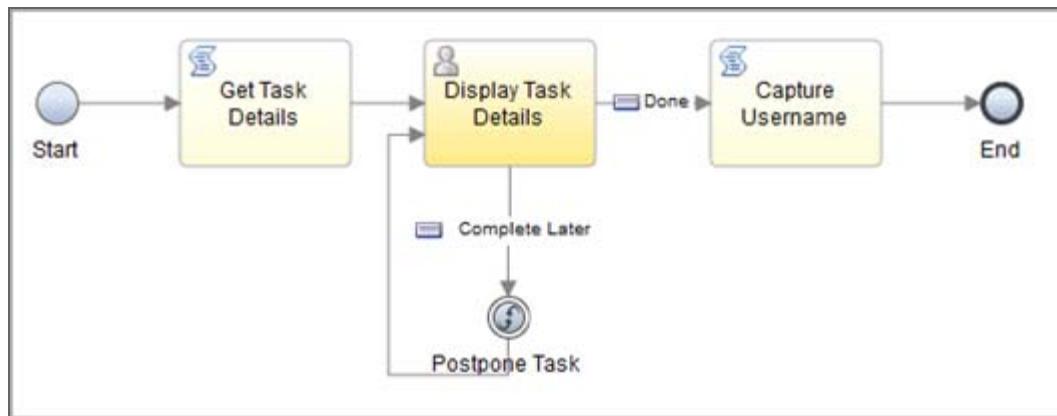


- \_\_ j. Click the **Diagram** tab and add a Client-Side Script that is named Capture Username to the sequence flow after the **Display Task Details** coach.



- \_\_ k. Click the **Capture Username** step.  
 \_\_ l. Enter the code in the **Script** properties menu to capture the username in the `userName1` variable:

```
tw.local.userName1 = tw.system.user.name;
```



- \_\_ m. Save your work.  
 \_\_ 3. Implement the new service on the process.  
 \_\_ a. From the history menu, open the **Approve Hire Request** linked process.

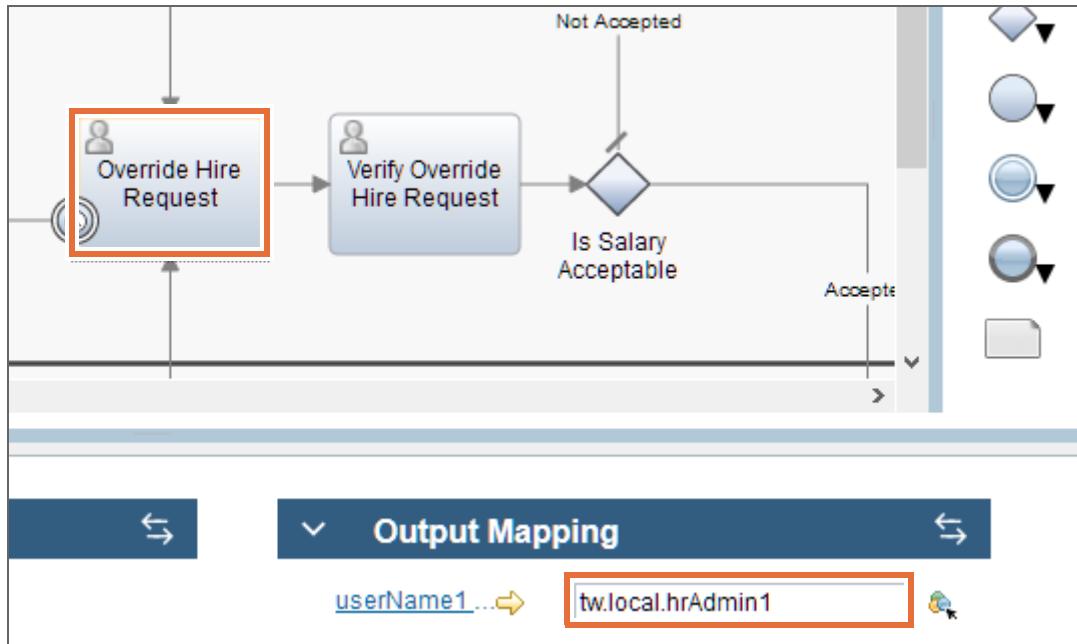
- \_\_ b. On the **Variables** tab, create a private variable: `hrAdmin1 (String)`

The screenshot shows the 'Variables' tab in a process editor. The variables are categorized into 'Input', 'Output', and 'Private'. The 'Private' section contains one item: 'hrAdmin1 (String)', which is highlighted with a red box. There are buttons for adding (+), removing (X), and reordering (up, down) variables.

- \_\_ c. Click the **Definition** tab.
- \_\_ d. Implement the activity with the **Override Hire Request** client-side human service in the **Implementation** section. Click the **Override Hire Request** in the HR Administrator lane and click the **Implementation** properties menu. Select the **Override Hire Request** human service for the implementation.

The screenshot shows a BPMN diagram with a task named 'Override Hire Request' in the 'HR Administrator' lane. A timer event is connected to the task with a duration of '4 Hours'. Below the diagram, the properties panel is open. The 'Implementation' tab is selected. Under 'Type:', 'User Task' is chosen. In the 'Implementation:' field, 'Override Hire Request' is selected, and this selection is highlighted with a red box.

- \_\_\_ e. Repeat the last step to implement the **Verify Override Hiring Request** activity with the same **Override Hire Request** client-side human service.
- \_\_\_ f. Click the **Override Hire Request** activity.
- \_\_\_ g. Open the **Data Mapping** properties menu and map the **userName1** output variable to the `tw.local.hrAdmin1(String)` private variable.

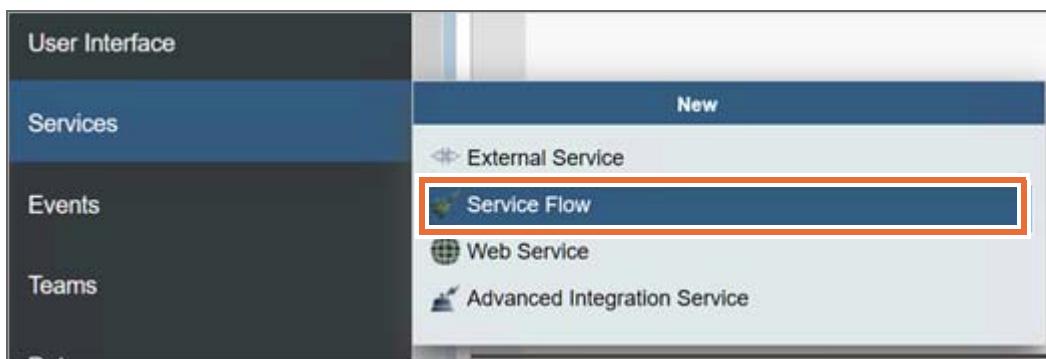


- \_\_\_ h. Map the **userName1** variable of the **Verify Override Hire Request** activity to the `tw.local.hrAdmin1 (String)` private variable.
- \_\_\_ i. Save your work.

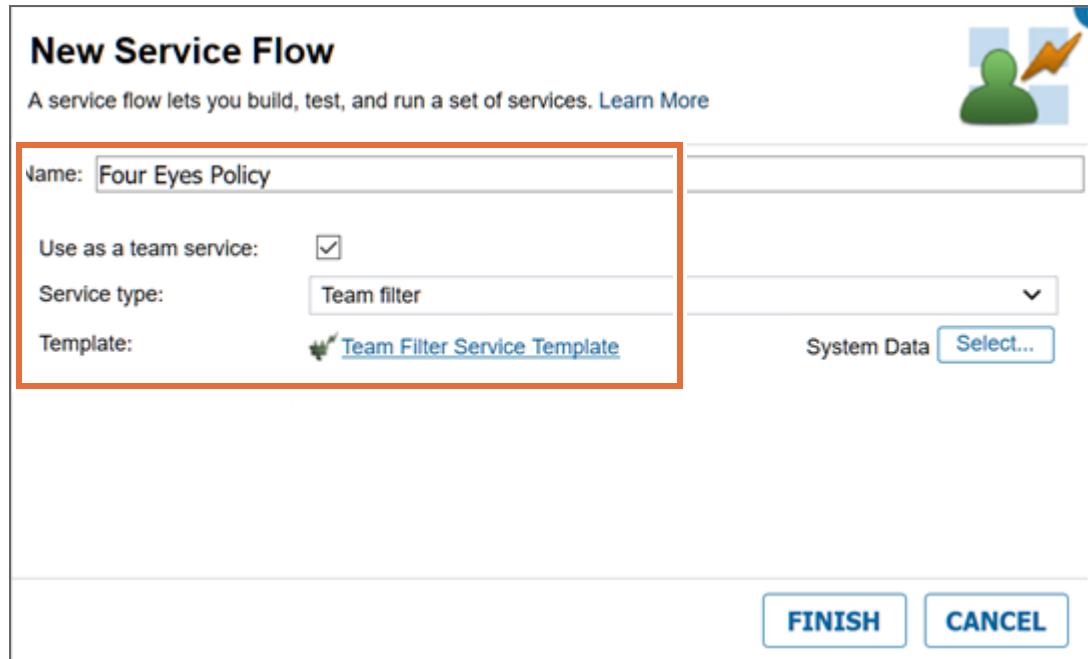
## Part 2: Create the team filter service

The four eyes principle requires that two people approve an activity. To implement the Four eyes policy, you need to know who the previous approver for the activity was, then filter that person out using Team Filter services.

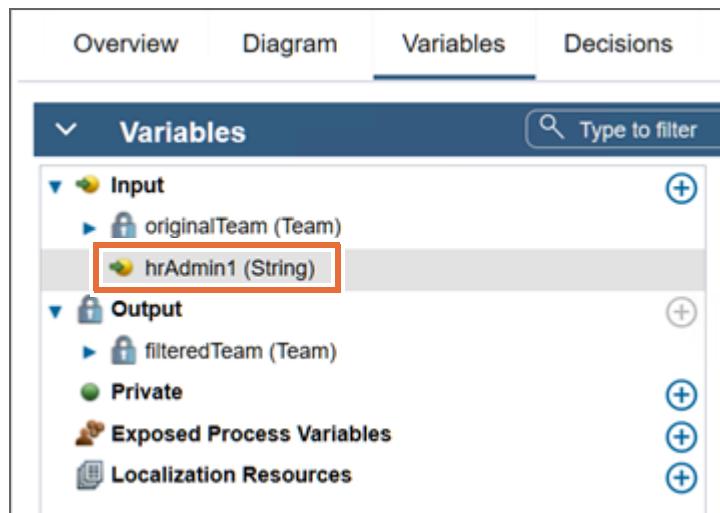
- \_\_\_ 1. Create a team filter service to assign the second task to the correct pool of users.
- \_\_\_ a. In the library, click the plus sign (+) next to the **Services** category, and then click **Service Flow**.



- \_\_\_ b. Name the service: Four Eyes Policy
- \_\_\_ c. Select the **Use as a team service** option.
- \_\_\_ d. Select **Team filter** for the **Service type** and verify that the **Team Filter Service Template** is set as the **Template**. This template is part of the System Data toolkit. The template defines the inputs and output variables that are required for this type of service.

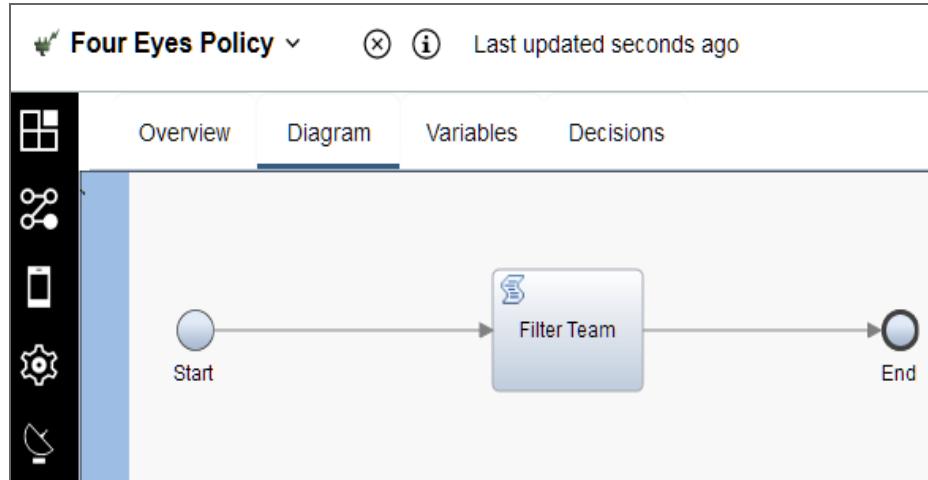


- \_\_\_ e. Click **FINISH**.
- \_\_\_ f. In the Four Eyes Policy service flow, click the **Variables** tab.
- \_\_\_ g. Add an input variable: hrAdmin1 (String)



- \_\_\_ h. Click the **Diagram** tab.

- \_\_ i. Drag a Server **Script** onto the sequence flow and name the it: Filter Team



- \_\_ j. Click the **Filter Team** server script, and then click the **Script** properties tab. Leave the default as **Script**. Do not use a Server **Scriptlet** for this task.

- \_\_\_ k. Filter the team by adding the following script. Alternatively, you can copy and paste the script that is provided in the file FilterJS.txt text file at the location:

C:\labfiles\Support files\Ex11

```

tw.local.filteredTeam = new tw.object.Team();
tw.local.filteredTeam.name = tw.local.originalTeam.name;
tw.local.filteredTeam.managerTeam = tw.local.originalTeam.managerTeam;
//copy the original team and then recreate the team members

tw.local.filteredTeam.members = new tw.object.listOf.String();
//iterate through original list

for (var i=0; i<tw.local.originalTeam.members.listLength; i++) {
 if (tw.system.org.findRoleByName(tw.local.originalTeam.members[i])) {
 //team
 var tempTeam =
 tw.system.org.findRoleByName(tw.local.originalTeam.members[i]).users;
 for (var j=0; j<tempTeam.length; j++) {
 if (tempTeam[j].name != tw.local.hrAdmin1)

tw.local.filteredTeam.members[tw.local.filteredTeam.members.listLength] = tempTeam[j].name;
 }
 }
 else { //individual user
 if (tw.local.originalTeam.members[i] != tw.local.hrAdmin1)
 //if member is not the hrAdmin member, add member to filtered
list

tw.local.filteredTeam.members[tw.local.filteredTeam.members.listLength] =
tw.local.originalTeam.members[i].toString();
 }
}

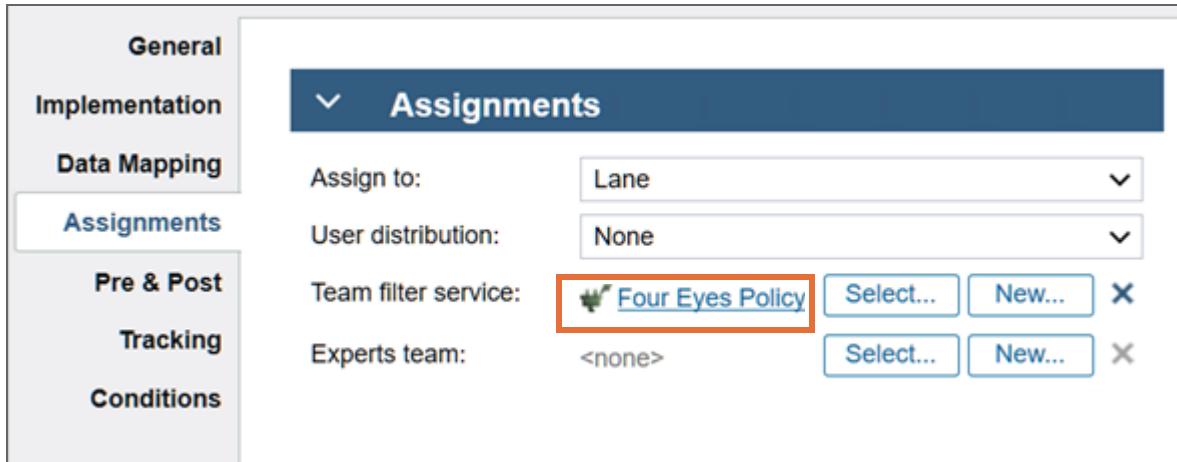
```



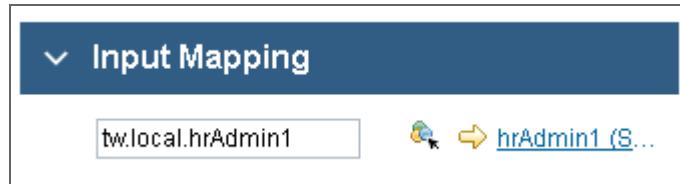
- \_\_\_ I. Save and close the Four Eyes Policy service.

### **Part 3: Apply the team filter service to the activity assignment**

- \_\_\_ 1. Apply the team filter service.
  - \_\_\_ a. Return to the **Approve Hire Request** process.
  - \_\_\_ b. Click the **Verify Override Hire Request** activity on the process.
  - \_\_\_ c. From the **Assignments** properties menu, select the **Four Eyes Policy** as the Team Filter service.



- \_\_\_ d. In the input mapping section for the team filter service on the right, map the `tw.local.hrAdmin1` variable.

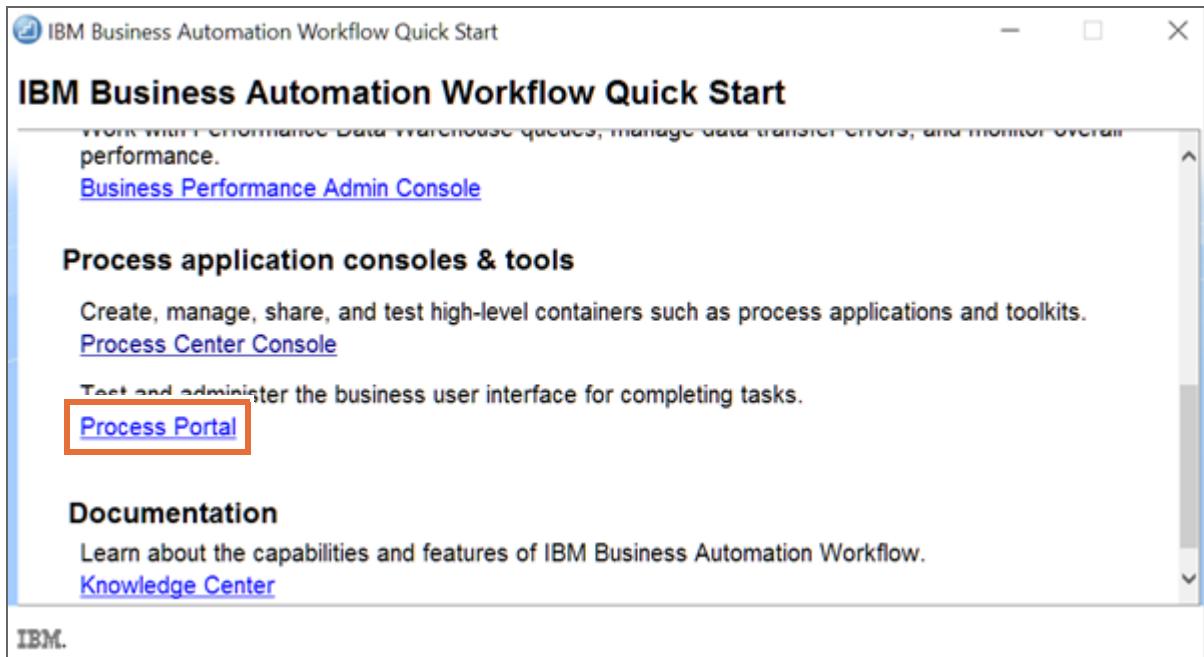


- \_\_\_ e. Save your work.

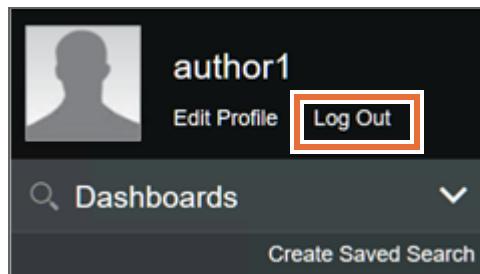
## Part 4: Test the four eyes policy

You now test the four eyes policy in the process portal.

- \_\_ 1. Log in as `user1` and start the process.
  - \_\_ a. If not already open, click the link to the **Process Portal** by using the Quick Start menu.



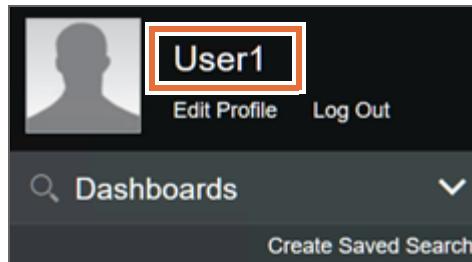
- \_\_ b. The Process Portal opens in a new browser tab. Since you were already logged in to the Workflow Center as **author1**, the session continues and you are logged in to Process Portal as **author1** also. For the team filtering to work successfully, you need to be logged in as a non-administrator user. If logged in as **author1**, log out of Process Portal by clicking **Log Out**.



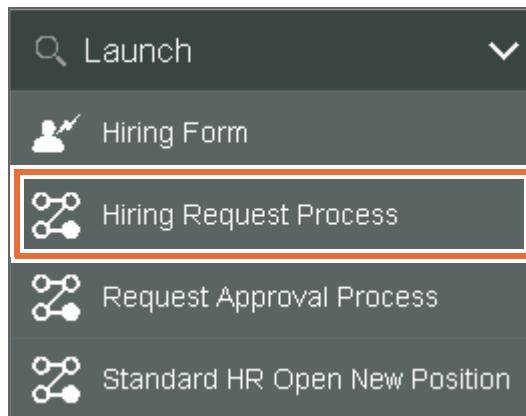
- \_\_\_ c. Log in to Process Portal using `user1` in the **Username** field and `passw0rd` in the **Password** field. Click **Continue**.

The screenshot shows the 'Sign in to Business Automation Workflow' page. It has two input fields: 'Username' containing 'user1' and 'Password' containing '\*\*\*\*\*'. Below the password field is a 'Continue' button with a right-pointing arrow icon.

- \_\_\_ d. Verify that **User1** is listed at the upper left corner.

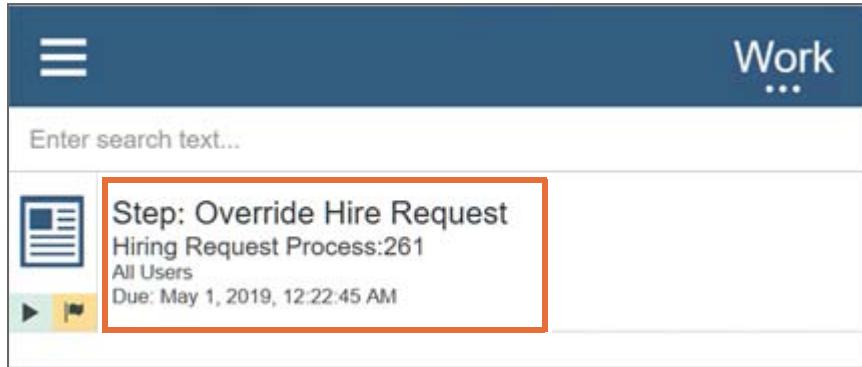


- \_\_\_ e. Create an instance of the process by clicking **Hiring Request Process** in the Launch section on the left. Wait for the first task to open in the main window.



- \_\_\_ f. Complete the first task by creating a non-compliant requisition. Use the following values for the variables on the form (settings inside the browser in parentheses):
- `tw.local.requisitionDetails.position.jobLevel` (**Position Details > Job Level**)  
enter: Manager
  - `tw.local.requisitionDetails.recruitingDetails newPosition` (**Recruiting Details > New Position**) set to: false (not selected)
  - `tw.local.requisitionDetails.compensationDetails.salaryToOffer` (**Compensation Details > Salary to Offer**) enter: 100000

- \_\_ g. Click **Submit**.
- \_\_ h. When the portal refreshes, the new task **Override Hire Request** activity is shown in the work dashboard. This task is the first task in the four eyes requirement. Click the **Step: Override Hire Request** link to run the task.



- \_\_ i. Click **Claim Task**.

A screenshot of a "Claim Task" dialog box. At the top left, it says "Claim Task". Inside the dialog, there is a task card identical to the one in the previous screenshot: "Step: Override Hire Request", "Hiring Request Process:261", "All Users", and "May 1, 2019, 12:22:45 AM". Below the card, a message reads "This task is still unclaimed. When you open this task, it will be assigned to you." There is a checkbox labeled "Don't show me this message again." At the bottom right, there are two buttons: "Cancel" and "Claim Task", with "Claim Task" highlighted by a red rectangular border.

- \_\_ j. In the Override Hire Request page, click **Done**.

This is a sample coach for the follow

Step: Override Hire Request

Done      Complete Later

This coach is just a placeholder, and would be replaced with an approval form. You are trying to test the assignment, so ignore the placeholder.

- \_\_ k. Wait for the work dashboard to refresh. No other tasks are shown in the dashboard even though a second task is created and assigned to the All Users team, but the current user user1 is filtered from the list. As a result, you do not see any task that is listed in the dashboard.
- \_\_ l. Click **Log Out** to logout of the Process Portal.
- \_\_ 2. Login as user2 and claim the **Verify Override Hire Request** task
- \_\_ a. Log back into the Process Portal, this time as user2. Enter user2 in the **Username** field and passw0rd in the **Password** field. Click **Continue**.

Sign in to Business Automation Workflow

Username

user2

Password

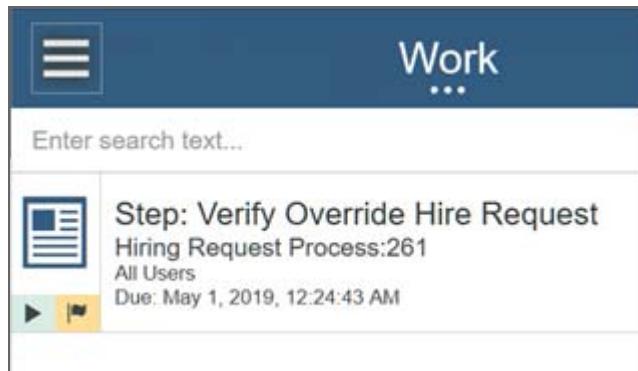
••••••••

Continue →

The task is shown in the Work page. The **Verify Override Hire Request** task is assigned to “All Users,” which is misleading. Although the role might seem to contain all

users, it is a filtered list. The original approver (user1) is filtered from the All Users list. The filter is the reason why the task is not listed in the user1’s work dashboard. The task is assigned to a team that filtered out the user who completed the override hire request task. In this example, user user1 completed the task of **Override Hire Request**, so user1 is filtered out and cannot complete the **Verify Override Hire Request**.

- \_\_\_ b. Click **Verify Override Hire Request**.



- \_\_\_ c. Click **Claim Task**.
- \_\_\_ d. In the Verify Hire Request coach that opens, click **Done**.



User2 successfully completed the **Verify Override Hire Request** task.

- \_\_\_ e. In the library on the left, click the link to **Log Out** of the process portal.
- \_\_\_ f. Close the Process Portal.

You have completed this exercise.

In the first part of the exercise, you modified the process and service to capture the first reviewer user ID. Next, you modified the activity on the process and created a routing policy. You tested the routing policy and verified that it filtered out the user from the list of participants available to claim the task.

## End of exercise

# Exercise 12. Building a cancellation pattern

## Estimated time

01:00

## Overview

In this exercise, you learn how to implement a cancellation pattern.

## Objectives

After completing this exercise, you should be able to:

- Implement a cancellation pattern in a process application
- Implement an undercover agent (UCA) to cancel the hiring request

## Introduction

A general manager requests the ability to cancel a hiring request while waiting for approval from the approvers. In this exercise, you create a cancellation message listener inside IBM Business Automation Workflow.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Create the event](#)

[Part 2: Create an undercover agent](#)

[Part 3: Implement the boundary event with the UCA](#)

[Part 4: Create the wrapper service](#)

[Part 5: Create a test harness](#)

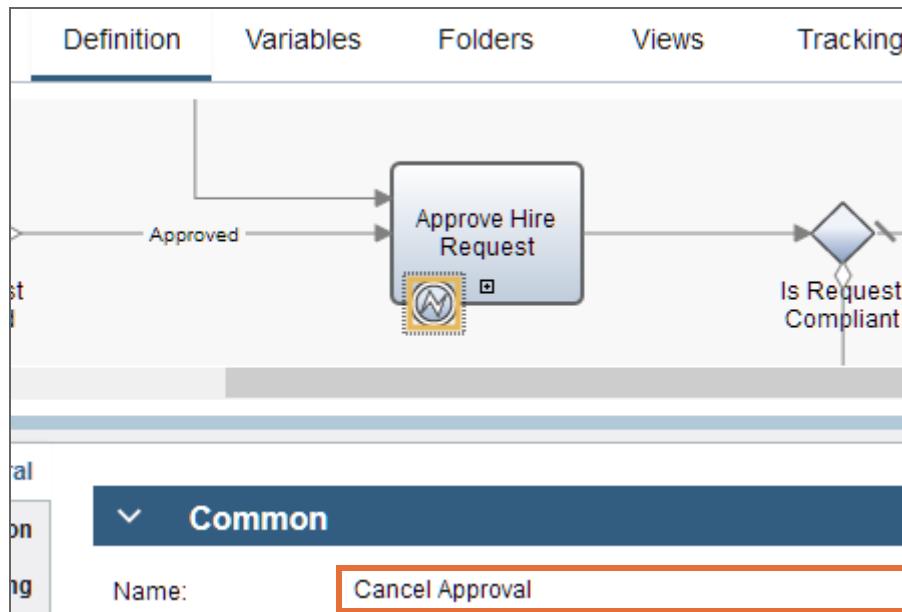
[Part 6: Test the cancellation UCA](#)

## Part 1: Create the event

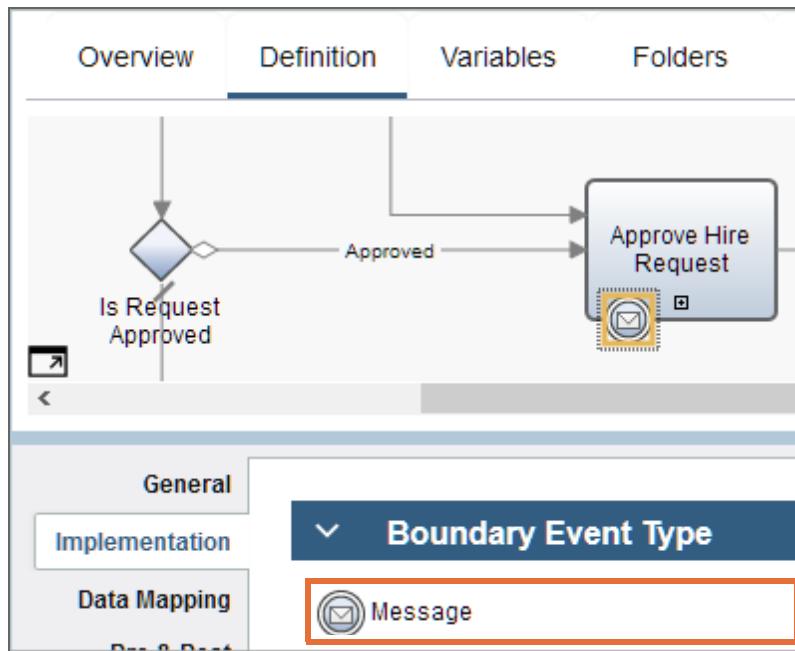
- 1. A cancellation event must be attached to the Approve Hire Request linked process.
  - a. Open the **HR Recruitment Process** process application.
  - b. Open the **Hiring Request Process**.
  - c. Drag an **intermediate event** from the palette and place it on one of the attachment points on the **Approve Hire Request** linked process. Make sure that you attach the intermediate event to the **Approve Hire Request** linked process and not the **Approve New Hire Request** activity.



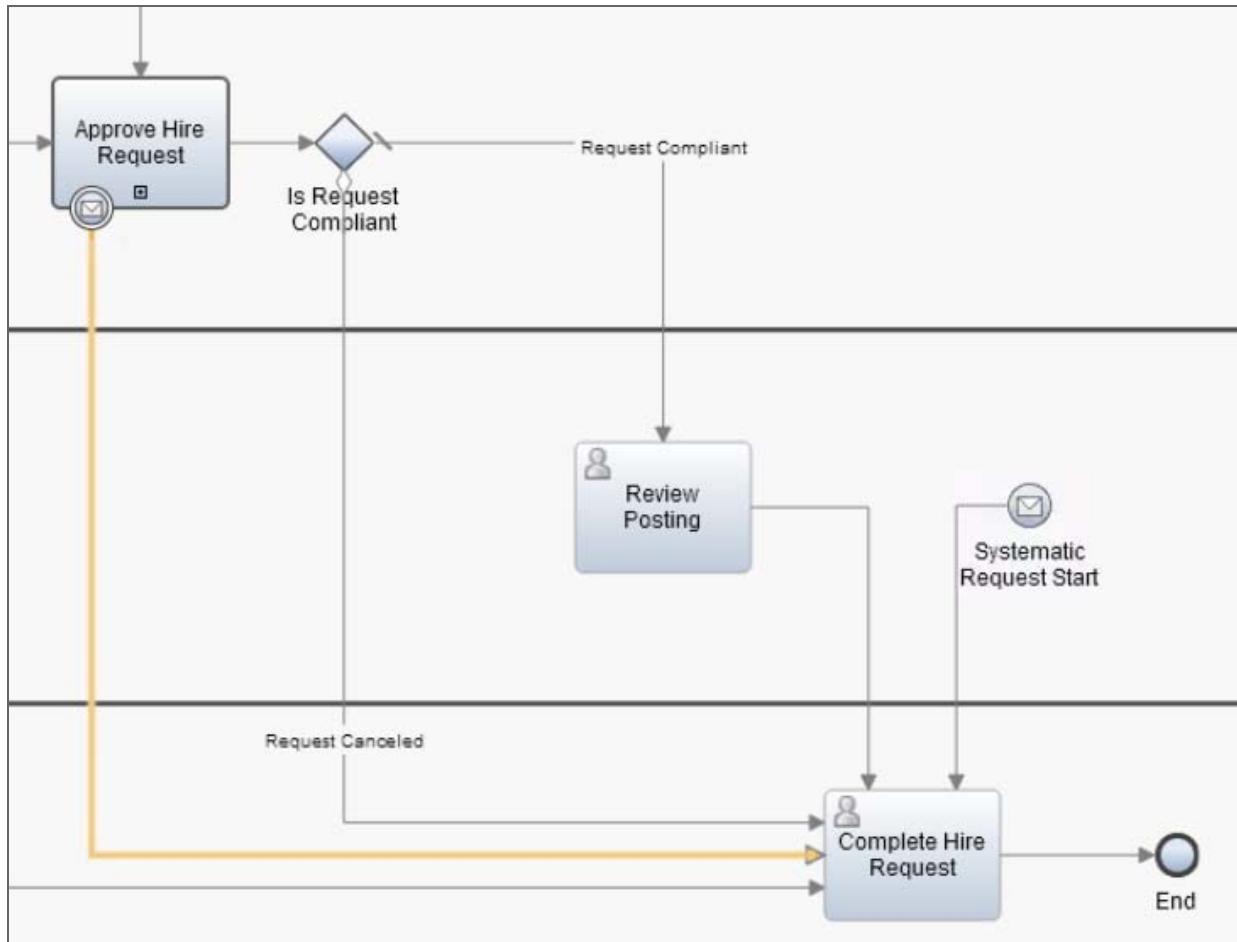
- d. Verify that the attached intermediate event is selected. Change the name of the event to: Cancel Approval.



- \_\_ e. From the **Implementation** properties menu, change the Boundary Event Type to **Message**.



- \_\_ f. Create a flow from the attached message intermediate event to the **Complete Hire Request** system lane activity.



- \_\_\_ g. Save your changes.



### Important

If an event changes the flow of a process similar to a cancellation event, you should also include some cleanup steps. This approach ensures that the process participants know about the event, and participants can react to the event as part of the process. Merely stopping or terminating an instance is not an ideal approach.

## Part 2: Create an undercover agent

- \_\_\_ 1. Add an undercover agent in the library.
  - \_\_\_ a. In the library, click the **plus sign (+)** next to the **Events** category, and click **Undercover Agent**.
  - \_\_\_ b. Name the undercover agent: Cancel Hiring Request
  - \_\_\_ c. Set the Schedule Type to: **On Event**.
  - \_\_\_ d. Click **FINISH**.

### New Undercover Agent

Undercover agents (UCAs) are started by an event and invoke a Process. UCAs pass data to the Process through an associated service or variable.

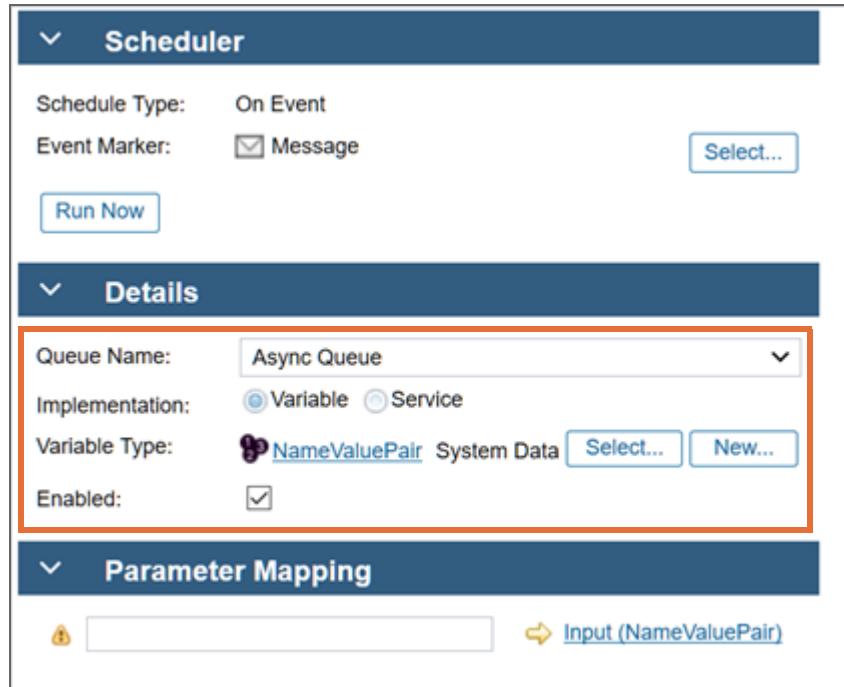
Name:

Schedule Type:

FINISH
CANCEL

- \_\_\_ 2. Configure the undercover agent.
  - \_\_\_ a. Under the **Details** section, keep all the default settings.
  - \_\_\_ b. Verify that the **Implementation** type is set to **variable**.
  - \_\_\_ c. Verify the default **Variable Type** is set to **NameValuePair**.

- \_\_\_ d. Verify that the **Enabled** checkbox is selected.



- \_\_\_ e. In the Event section at the bottom, change the Event Message to: CancelHiringRequest.



- \_\_\_ f. Save your work.

### **Part 3: Implement the boundary event with the UCA**

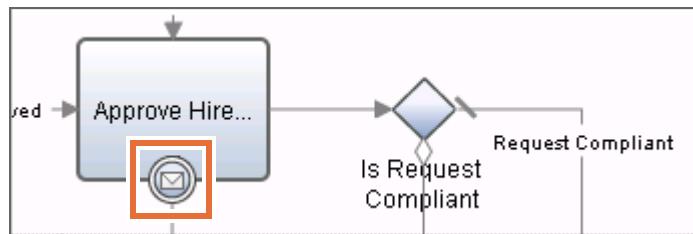
- \_\_\_ 1. Create a variable to correlate the UCA.  
 \_\_\_ a. Switch to the **Hiring Request Process** in the Designer.

- \_\_ b. On the **Variables** tab, create a private variable: cancellationId (String)

The screenshot shows the 'Details' tab of a variable configuration. The 'Name' field contains 'cancellationId'. Below it, there are buttons for bold (B), italic (I), underline (U), and various alignment and spacing options. Under 'Variable type', it says 'String' with a small icon. To the right are buttons for 'System Data', 'Select...', and 'New...'. At the bottom left is a 'List' button.

- \_\_ 2. Configure the boundary event.

- \_\_ a. Click the **Definition** tab.  
\_\_ b. Click the **Approve Hire Request** boundary event.



- \_\_ c. Click the **Implementation** properties menu.  
\_\_ d. Verify that the **Interrupt activity** checkbox is selected. The interrupt activity option consumes the token of the attached activity when the event is triggered. Because it is a linked process, even if numerous tokens exist inside the linked process, all the tokens are closed when the event is triggered.

The screenshot shows the 'Implementation' tab for a boundary event. In the 'Boundary Event Type' section, there is a list with 'Message' selected. Below it, the 'Interrupt activity:' checkbox is checked (indicated by a red box). There is also a 'Repeatable:' checkbox which is unchecked.

- \_\_\_ e. In the **Event Properties** section, click **Select** next to the **Attached message UCA** and select **Cancel Hiring Request**.



- \_\_\_ f. In the **Event Properties** section, select the **Consume message** checkbox. Consume message allows correlation with only a single process instance. If you configure multiple event listeners to correlate on the same message, then all events trigger if the **Consume message** checkbox is not selected.

**Event Properties**

Attached message UCA: **Cancel Hiring Request**

Condition: 1

Consume message:

Durable subscription:

- \_\_\_ g. Save your changes.
- \_\_\_ 3. Map the elements of the **Output (NameValuePair)** to simple variables in the process.
- \_\_\_ a. Click the **Data Mapping** properties menu.
- \_\_\_ b. From the **Correlation variable** list, select the **value (String)** variable.

Correlation variable:

Output (NameValuePair)

value (String)

- \_\_\_ c. Map the **Output.name (String)** value to the `tw.local.cancellationId` variable.

- \_\_\_ d. Map the Output.value (String) to the tw.system.process.instanceId variable.

**Correlation and Output Mapping**

Specify the variable that will be used to correlate the incoming event to the process, so that the process instance. Then, map the variable to a local variable. You can also create additional mappings from the output of intermediate events.

Correlation variable: value (String)

Output (NameValuePair)

|                |                              |
|----------------|------------------------------|
| name (String)  | tw.local.cancellationId      |
| value (String) | tw.system.process.instanceId |

The instance must know which message is targeting it. The messages sent to the UCA must contain a value that matches the instance ID of this process instance in order for this instance to receive the message. When the Output.value and the instance ID correlate, the intermediate event processes the message, and a token flows down the path. The user ID that canceled the process is also captured and mapped to a local variable.

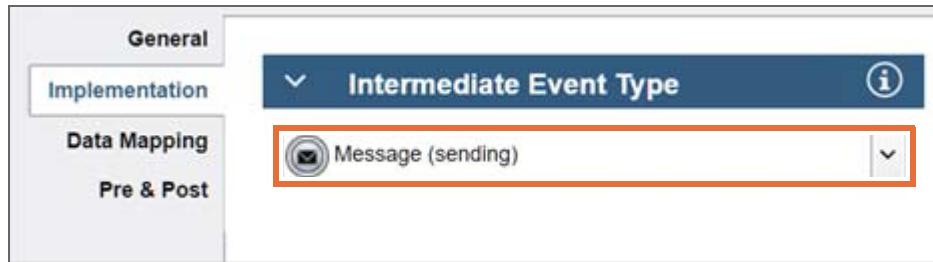
- \_\_\_ e. Save your work.

#### Part 4: Create the wrapper service

- \_\_\_ 1. Create a service to trigger the UCA.
- \_\_\_ a. In the library, click the **plus sign (+)** next to the **Services** menu and create a **Service Flow**.
- \_\_\_ b. Name the service: Cancel Hiring Request Trigger
- \_\_\_ c. Click **FINISH**.
- \_\_\_ d. Drag a **Message (Sending) Intermediate Event** on to the sequence flow.



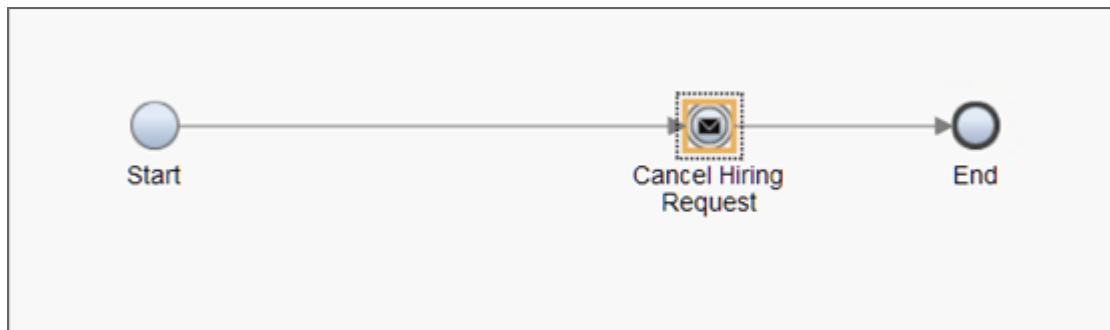
- \_\_ e. Select the **Implementation** properties section below.
- \_\_ f. Verify the implementation type is set to **Message (sending)**.



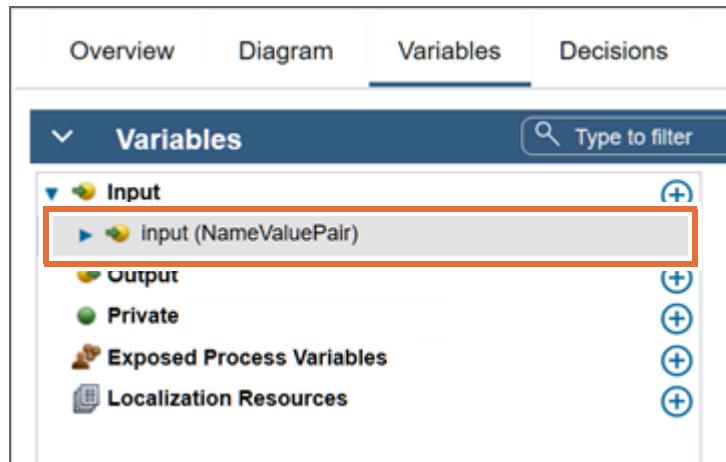
- \_\_ g. In the **Event Properties** section on the right, click **Select** next to **Attached message UCA** and select **Cancel Hiring Request**.



- \_\_ h. Rename the intermediate message event: Cancel Hiring Request.

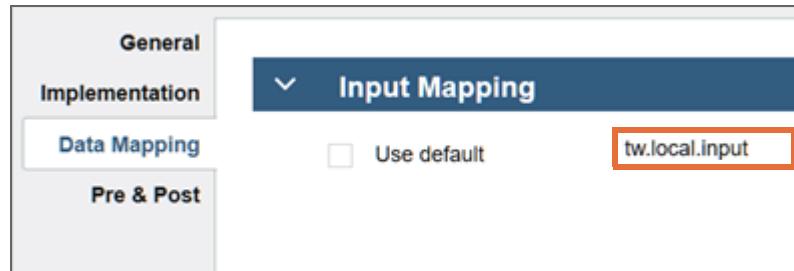


- \_\_ i. On the **Variables** tab, create an input variable: `input (NameValuePair)`



- \_\_ j. Save your changes.

- \_\_\_ 2. Map the variable.
  - \_\_\_ a. Click the **Diagram** tab.
  - \_\_\_ b. Click the **Cancel Hiring Request** event.
  - \_\_\_ c. Click the **Data Mapping** properties menu.
  - \_\_\_ d. Map the input of the UCA to the `tw.local.input` variable.



- \_\_\_ e. Save your work.

## **Part 5: Create a test harness**

- \_\_\_ 1. Add a client-side human service to test the UCA.
  - \_\_\_ a. In the library, click the **plus sign (+)** next to **User Interface**. Click **Client-Side Human Service**.
  - \_\_\_ b. Name the service: **Cancel Hiring Request Test**. Select the **Intended for use on multiple devices** checkbox. Click **FINISH**.

The screenshot shows a dialog box titled "New Client-Side Human Service". It contains a brief description of what a client-side human service is. Below the description, there is a "Name:" field containing "Cancel Hiring Request Test". There are two checkboxes: one checked and highlighted with a red box, and another unchecked. The checked checkbox is labeled "Intended for use on multiple devices". The unchecked checkbox is labeled "Use as a nested service". At the bottom right of the dialog are two buttons: "FINISH" (highlighted with a red box) and "CANCEL".

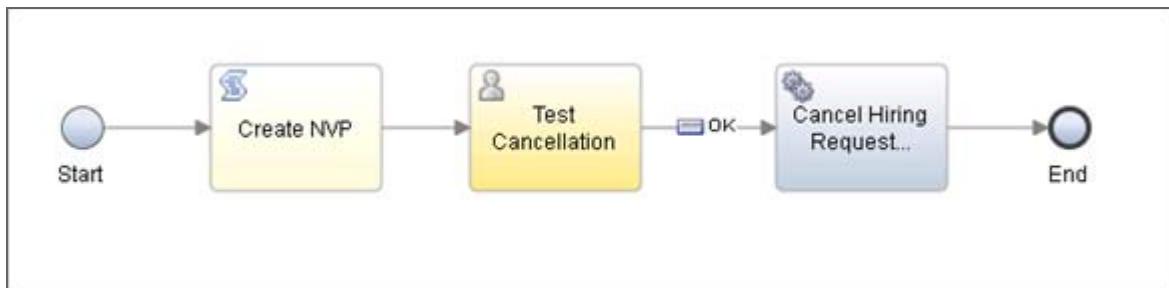
- \_\_\_ c. Drag a Client-side script from the palette and place it next to the Start event on the sequence flow. Name it: Create NVP



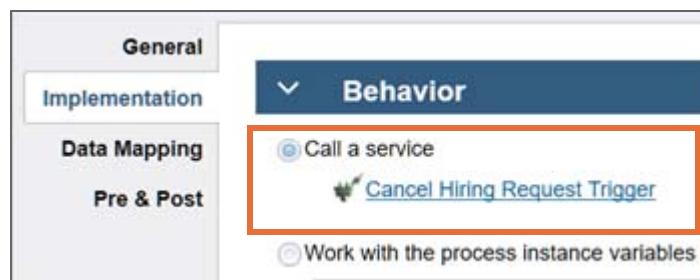
### Information

The abbreviation for name-value pair is NVP for this step.

- \_\_\_ d. Select the coach on the canvas and name it: Test Cancellation  
 \_\_\_ e. Drag a service task from the palette onto the sequence flow to the right of the coach and name it: Cancel Hiring Request Trigger  
 \_\_\_ f. Connect the flows.

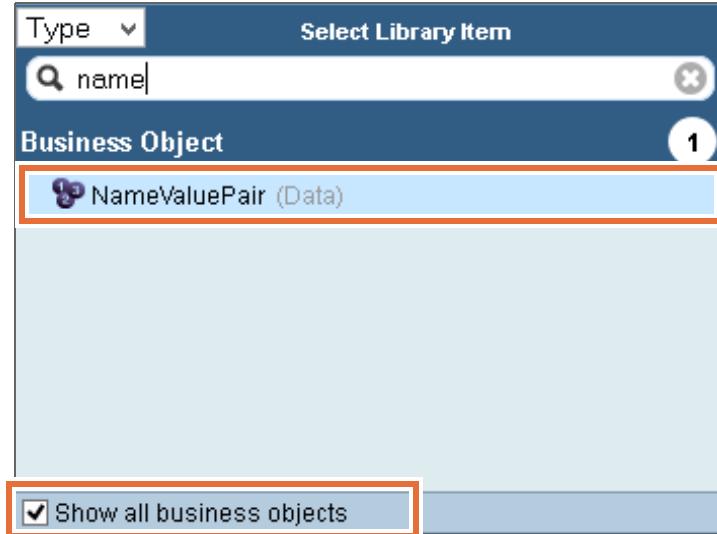


- \_\_\_ g. Select the **Implementation** properties menu.  
 \_\_\_ h. For the **Call a service**, click **Select**, and choose the **Cancel Hiring Request Trigger** service.

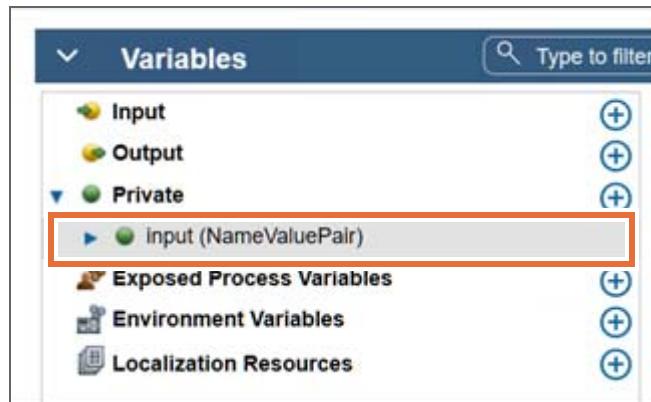


- \_\_\_ i. Save your changes.

- \_\_\_ 2. Implement the Create NVP step on the canvas.
- \_\_\_ a. On the **Variables** tab, create a private variable: `input`.
- \_\_\_ b. Assign the variable type to **NameValuePair**.



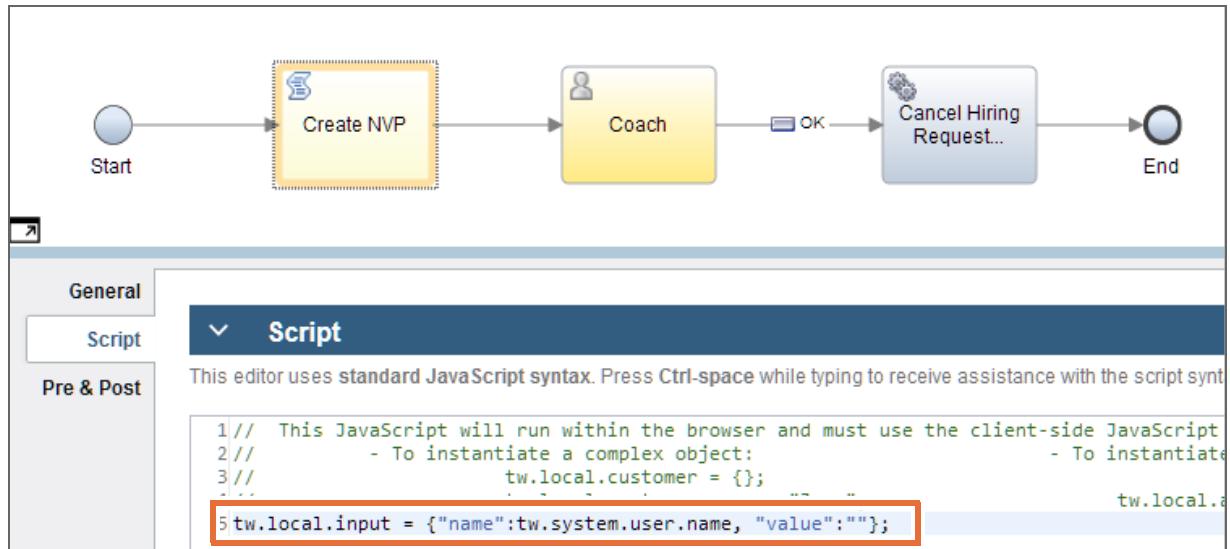
- \_\_\_ c. Verify that the variable is created with the **NameValuePair** type and save your work.



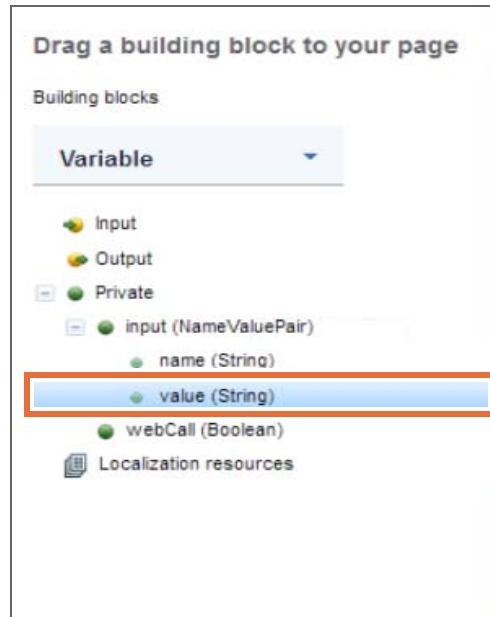
- \_\_\_ d. Click the **Diagram** tab.
- \_\_\_ e. Select the **Create NVP** script, and click the **Script** properties menu.

- \_\_\_ f. Copy the following code and paste it into the code block. Alternatively, you can copy and paste the script that is provided in the file CreateNVP.txt text file at the location:  
C:\labfiles\Support files\Ex12.

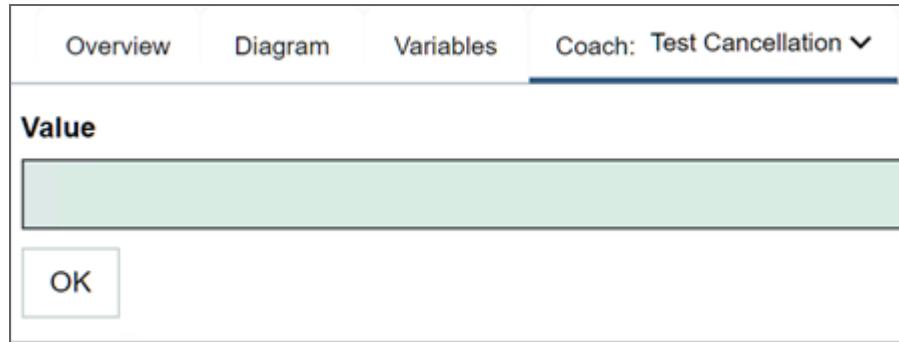
```
tw.local.input = { "name":tw.system.user.name, "value":"" };
```



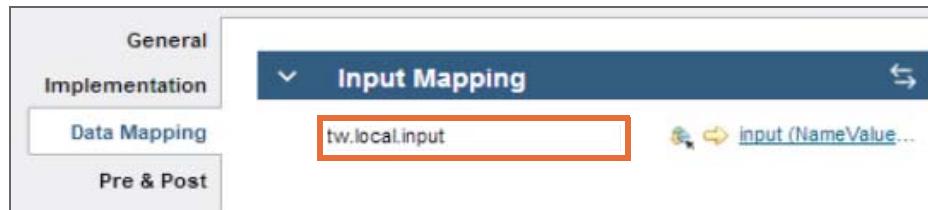
- \_\_\_ g. Save your changes.  
 \_\_\_ 3. Build the Test Cancellation coach.  
 \_\_\_ a. On the **Coaches** tab, click the **Test Cancellation** coach.  
 \_\_\_ b. Select **Default coach** as the starting point and click **OK**.  
 \_\_\_ c. Select **Variable** in the **Building blocks** panel and select **Private > input > value**



- \_\_\_ d. While holding down the left mouse button, drag the `input.value` variable to the top of the **OK** button. Let go of the mouse button when the top border displays a gold highlight



- \_\_\_ e. Change the input field label to: Instance ID
- \_\_\_ 4. Map the input variable for the Cancel Hiring Request Trigger step on the canvas.
- Open the **Diagram** tab.
  - Click the **Cancel Hiring Request Trigger** service.
  - Click the **Data Mapping** properties menu, and map the `tw.local.input` private variable to the input.



- \_\_\_ d. Save your work.

## **Part 6: Test the cancellation UCA**

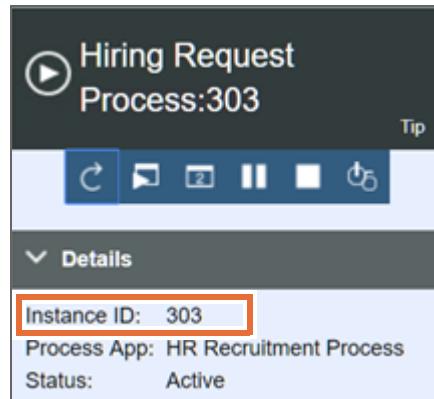
- Create an instance of the Hiring Request Process.

  - Open the **Hiring Request Process**
  - Click **Run** to create an instance of the process.



- \_\_\_ c. The perspective changes to the **Inspector** tab, and the information bar opens on the right.

- \_\_\_ d. The **Instance ID** in the Process Inspector is listed on the right. Make a note of this value. You use this number in the next step to cancel the instance. Your instance ID will be different than the one in the screen capture.



- \_\_\_ 2. Complete the first activity in the process.
- \_\_\_ a. To complete the first activity and get to the linked process, make sure that the **Locations** section is expanded.
- \_\_\_ b. Click **Start** to run the Submit Hiring Request task from the Inspector information bar.

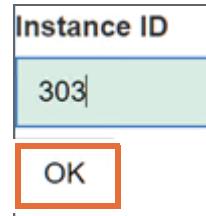


- \_\_\_ c. Use the following values for the variables on the form:
- Position Details > Job Level set to: Manager
  - Recruiting Details > New position set to **false** (not selected)
  - Compensation Details > Salary to offer set to 100000
- \_\_\_ d. Click **Submit**.
- \_\_\_ e. When the service completes, close the coach browser window.

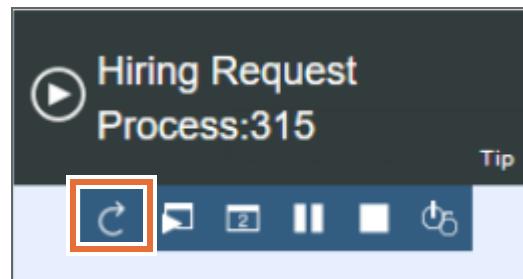
When the Inspector refreshes, a token is associated with the Approve Hire Request linked process. Next, you cancel the linked process and close all the tokens that are associated with the linked process by sending a message to the attached intermediate message event.

- \_\_\_ 3. Trigger the cancellation event on the linked process.
- \_\_\_ a. Open the **User Interface > Cancel Hiring Request Test** client-side human service.
- \_\_\_ b. Click **Run**.

- \_\_\_ c. Input the instance ID that you noted earlier. Click **OK**.



- \_\_\_ d. When the service completes, close the window.
- \_\_\_ 4. Verify that the linked process is canceled.
- \_\_\_ a. Return to the **Hiring Request Process** in the Inspector view.
- \_\_\_ b. Refresh the Process Inspector.



- \_\_\_ c. Verify that the message consumed the token for the **Verify Override Hire Request** linked process and all tokens that are associated with the linked process are closed.
- \_\_\_ d. In the Locations section on the right, click **Start** to run the Complete Hire Request activity.



- \_\_\_ e. Click **Done**
- \_\_\_ f. Close the browser window.
- \_\_\_ g. Return to the **Designer** tab.

All the tasks complete and the process consumes all the tokens in the linked process. The process flow travels out of the attached message event to the system lane activity. After the Complete Hire Request activity completes, the token moves to the End event and the process is complete. The instance status changes to **Completed**.

You have completed this exercise.

In the first part of the exercise, you created an attached intermediate event on the process. You implemented it as an intermediate message event. You created the enabling service, created the on-demand UCA, and then implemented the message service. You then created a service to wrap the UCA and created a test harness to trigger the UCA on the coach.

## End of exercise

# Exercise 13. Building web service connections

## Estimated time

01:00

## Overview

In this exercise, you learn how to build inbound and outbound web service connections.

## Objectives

After completing this exercise, you should be able to:

- Create an event-based undercover agent
- Build an inbound web service connection
- Build an outbound web service to message the inbound web service

## Introduction

The project sponsor identified a requirement to expose a cancellation service to an external system. The ability to cancel a request must be enabled internally, but also must be available to systems that integrate externally with IBM Business Automation Workflow.

You must create a cancellation message listener as an inbound web service inside IBM Business Automation Workflow. Then, create a web service to test or simulate an external system, which connects to the inbound web service. This activity gives you practice in creating web services and calling external SOAP services. Even though the message is targeting the localhost, the approach and concepts that are used in this activity are the same for an external system.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Create an inbound web service to trigger the UCA](#)

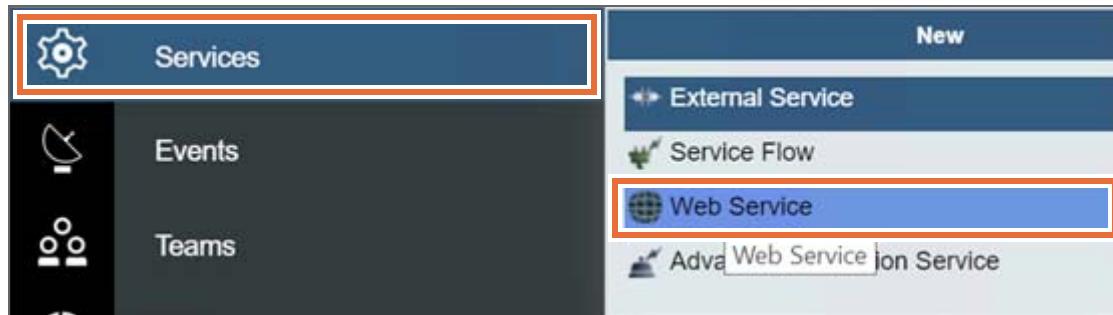
[Part 2: Test the inbound integration by creating an outbound integration](#)

[Part 3: Test the integrations](#)

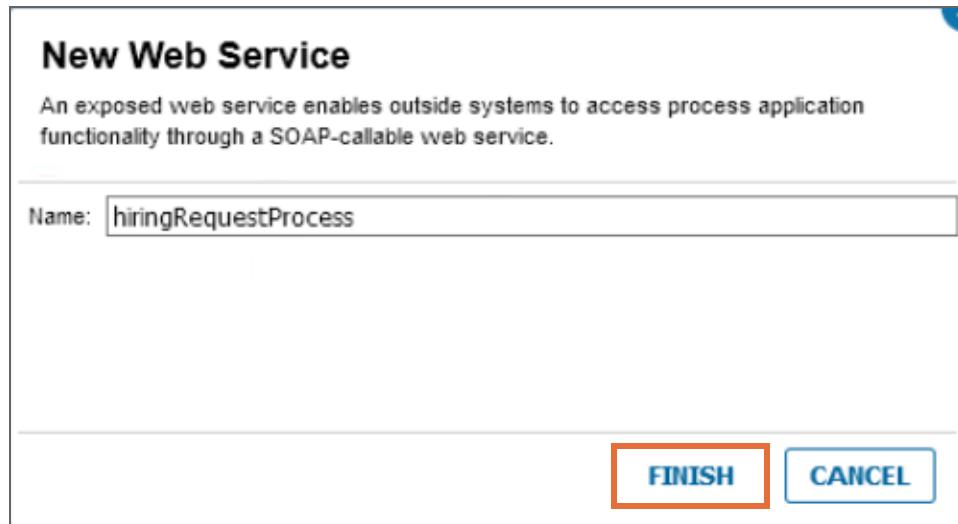
## Exercise instructions

### Part 1: Create an inbound web service to trigger the UCA

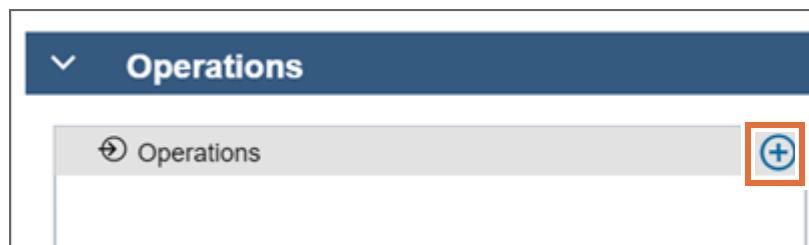
- \_\_\_ 1. Create an inbound web service.
- \_\_\_ a. Open the **HR Recruitment Process** process application.
- \_\_\_ b. In the Process Designer library, click the **plus sign (+)** next to the **Services** category.
- \_\_\_ c. Click **Web Service**.



- \_\_\_ d. Name the web service: `hiringRequestProcess` and click **FINISH**.



- \_\_\_ e. Click **Add** (plus icon) under the **Operations** section and select the **Cancel Hiring Request Trigger** service.



- \_\_ f. Under the **Operation Detail** section, change the **Operation Name** to: cancel

The screenshot shows the 'Operation Detail' configuration window. The 'Operation name' field contains the value 'cancel', which is highlighted with a red rectangular border. Below it, the 'Attached service' field shows 'Cancel Hiring Request Trigger' with 'Select...' and 'New...' buttons. The 'Documentation' section includes a rich text editor toolbar.

- \_\_ g. Under the **Behavior section**, verify that the **Protected** checkbox is not selected.

The screenshot shows the 'Behavior' configuration window. The 'Protected' checkbox is unselected (unchecked). Other fields include 'WSDL URI' set to <https://WS2016X64:9443/teamworks/webservices/HRR/hiringRequestProcess.tws?WSDL>, 'Target namespace scheme' set to 'Use process app or toolkit settings', 'Target namespace' set to <http://HRR/hiringRequestProcess.tws>, and 'SOAP version' with options for 1.1 and 1.2.

- \_\_ h. Save your work.

- \_\_ i. Click the **WSDL URI** link.

The screenshot shows the 'Behavior' configuration window again. The 'WSDL URI' field contains the URL <https://WS2016X64:9443/teamworks/webservices/HRR/hiringRequestProcess.tws?WSDL>, which is highlighted with a red rectangular border.

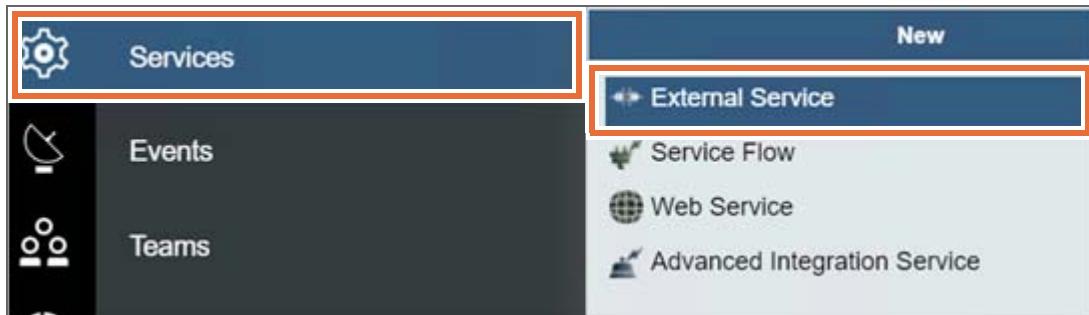
- \_\_ j. The WSDL is loaded in your browser. The information about the web service you created is shown to include the methods available, the variable inputs and outputs, and metadata about the service call. Highlight the address in the browser address bar, right-click, and click **Copy**.

The screenshot shows a right-click context menu over a browser address bar. The address bar displays the URL <https://ws2016x64:9443/teamworks/webservices/HRR/hiringRequestProcess.tws?WSDL>. The context menu includes options like Undo (Ctrl+Z), Cut (Ctrl+X), Copy (highlighted with a red box and labeled Ctrl+C), Paste (Ctrl+V), and others.

The URL is used to define the server setting for the **Web Service Integration** component in the outbound integration in the following steps.

## Part 2: Test the inbound integration by creating an outbound integration

- 1. Create an outbound integration.
- a. In the library, click the plus sign (+) next to the **Services** category, and select **External Service**.



- b. Leave the **Java REST or Web service** option selected and click **NEXT**.

**New External Service**

An external service lets you call a service or application that is external to IBM Business Integration. [Learn More](#)

Java, REST or Web service

Create an external service by discovering a Java, REST or Web service.

External Implementation

Create an external service with an External Implementation binding.

- c. Select **Web Service from URL** as the method to discover the service.
- d. For the **External service name**, enter `Cancel Hiring Request WS`

- \_\_\_ e. In the URL field, enter the URL that you copied earlier. If needed, switch to the browser tab and copy it.

## New External Service

An external service lets you call a service or application that is external to IBM Business Automation Workflow.  
[Learn More](#)

Select a method to discover the service.

Web service from URL

External service name:

URL:

User name:

Password:

- \_\_\_ f. Click **NEXT**.

- \_\_\_ g. Leave the default operation selected and click **NEXT**.

## New External Service

An external service lets you call a service or application that is external to IBM Business Automation Workflow.  
[Learn More](#)

Select the operations to include in the generated external service.

Operation Name

cancel

- \_\_\_ h. Leave the defaults for the request and response wrappers (unchecked) and the business objects to be created and click **NEXT**.

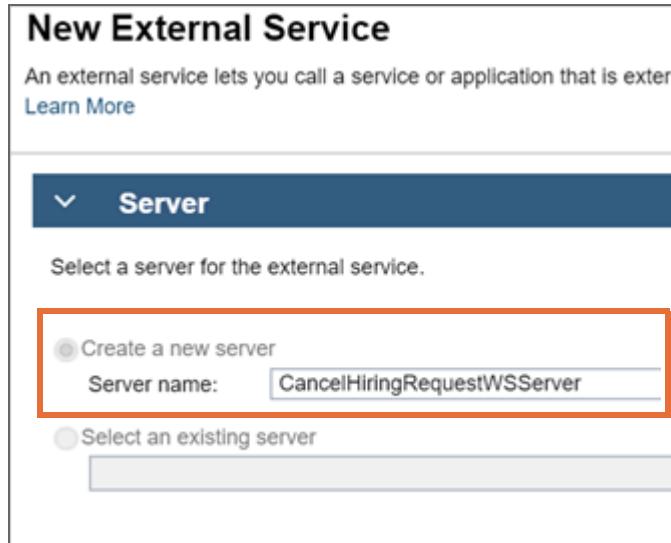
## New External Service

An external service lets you call a service or application that is external to IBM Business Automation Workflow.  
[Learn More](#)

The following business objects will be created for the corresponding XML Schema Type.

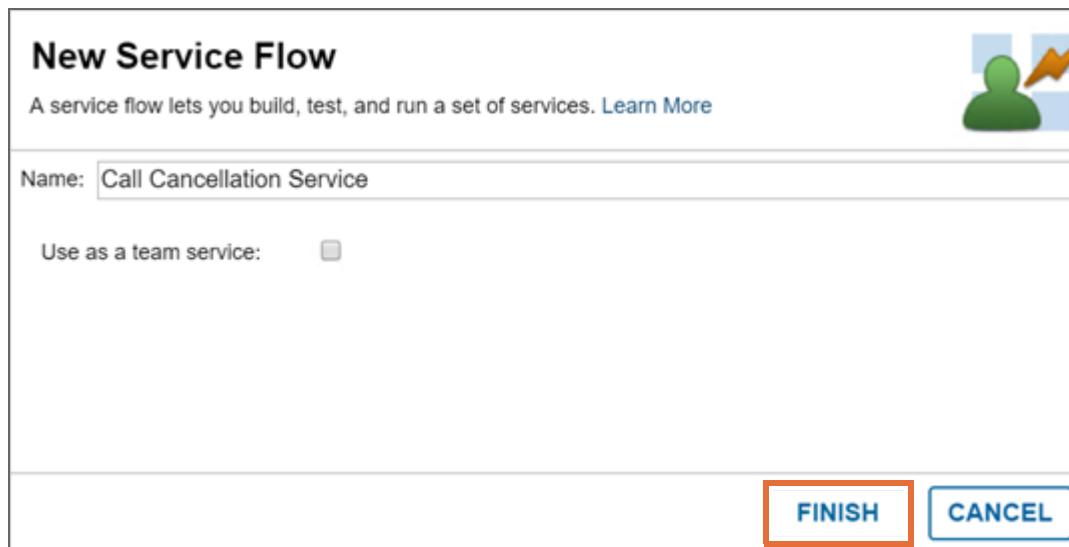
| XML Schema Type                                | Business Object |
|------------------------------------------------|-----------------|
| {http://lombardi.ibm.com/schema/]NameValuePair | NameValuePair   |

- \_\_\_ i. Leave the default new server name and click **FINISH**.

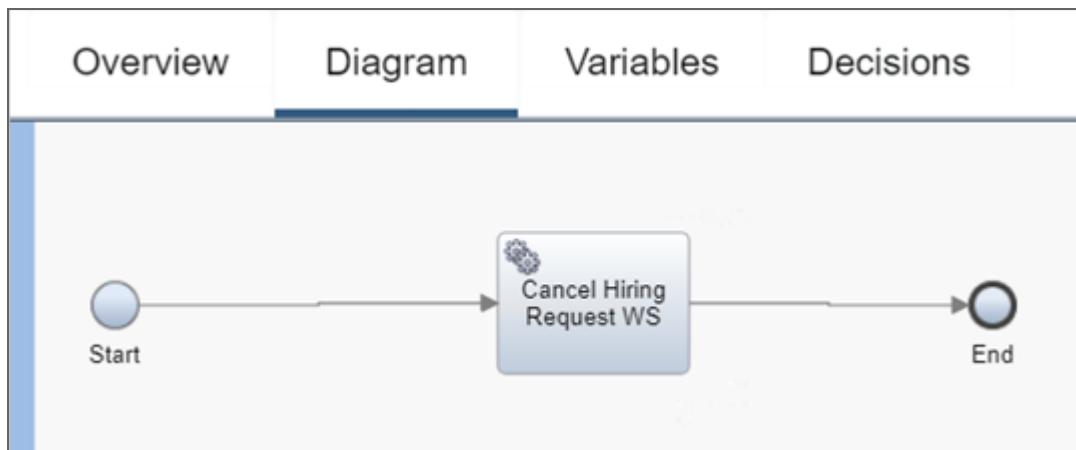


The external service configuration is displayed after you click **FINISH**.

- \_\_\_ 2. Create the service flow for the web service.
- \_\_\_ a. In the library, click the **plus sign (+)** next to the **Services** category, and select **Service Flow**.
- \_\_\_ b. Name the Service flow: Call Cancellation Service
- \_\_\_ c. Click **FINISH**.



- \_\_\_ d. In the library, click **Services**, locate **Cancel Hiring Request WS** and drag it to the sequence flow for the **Call Cancellation Service** flow.



### Troubleshooting

If you cannot drag the service to the canvas, close the browser and reopen the process application.

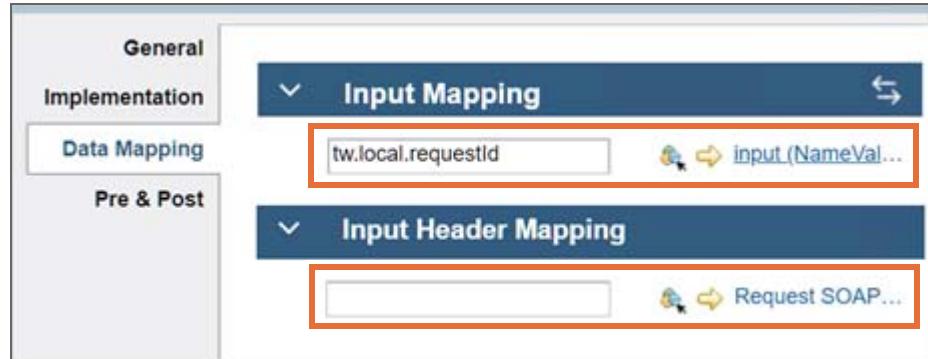
- \_\_\_ e. With the **Cancel Hiring Request WS** service highlighted, click the **Implementation** properties menu.  
 \_\_\_ f. In the Implementation section, change the **Operation** value to **cancel**.

|                       |                                                                          |
|-----------------------|--------------------------------------------------------------------------|
| <b>Implementation</b> |                                                                          |
| Implementation:       | <input type="button" value="Cancel Hiring Request WS"/> Select... New... |
| Operation:            | cancel                                                                   |
| Port:                 | soap12hiringRequestProcessSoap (SOAP 1.2)                                |

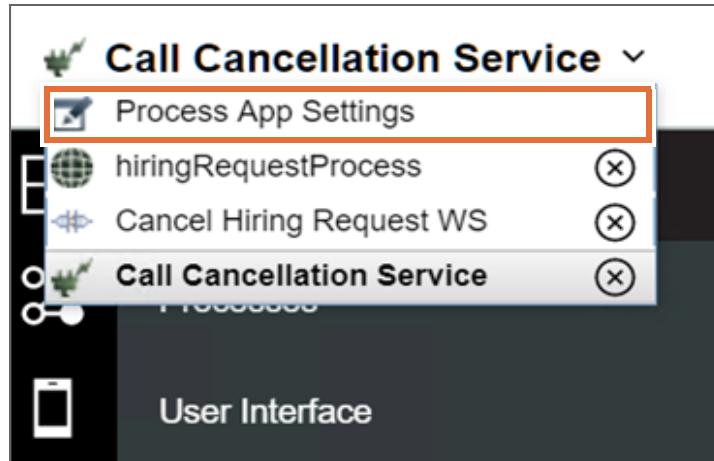
- \_\_\_ g. On the **Variables** tab, add an input variable named: `requestId` (`NameValuePair`)

|                                                                                                 |  |
|-------------------------------------------------------------------------------------------------|--|
| <b>Variables</b>                                                                                |  |
| <input type="text" value="Type to filter"/>                                                     |  |
| <b>Input</b> <ul style="list-style-type: none"> <li><b>requestId (NameValuePair)</b></li> </ul> |  |
| <b>Output</b>                                                                                   |  |
| <b>Private</b>                                                                                  |  |
| <b>Exposed Process Variables</b>                                                                |  |
| <b>Localization Resources</b>                                                                   |  |

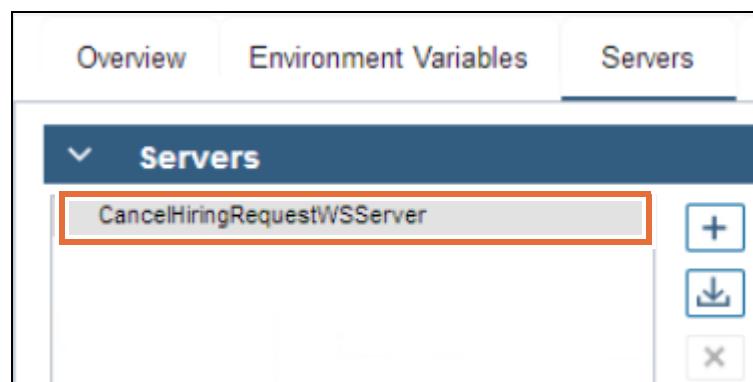
- \_\_ h. Click the **Diagram** tab. Click the **Cancel Hiring Request WS** component. Open the **Data Mapping** properties menu.
- \_\_ i. In the Input Mapping section, map the `tw.local.requestId` variable to the input of **Cancel Hiring Request WS**. Leave Input Header Mapping and Output Header Mapping blank.



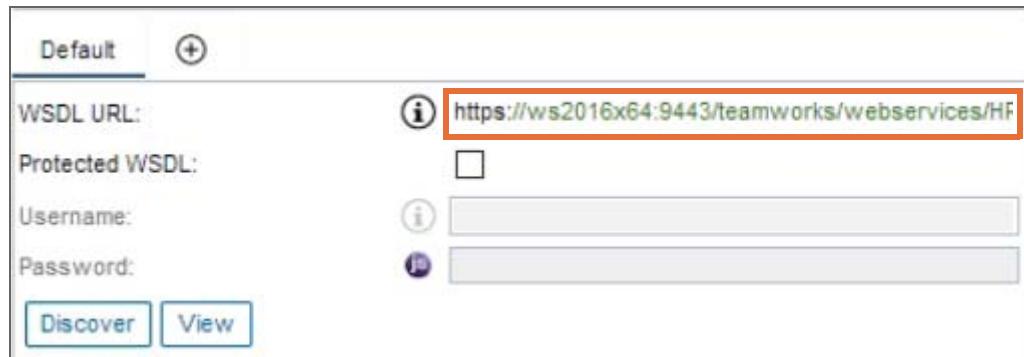
- \_\_ j. Save your work.
- \_\_ 3. Verify the new server that you just added.
  - \_\_ a. Click **Process App Settings**.



- \_\_ b. Click the **Servers** tab and verify that the new server **CancelHiringRequestWSServer** is listed.

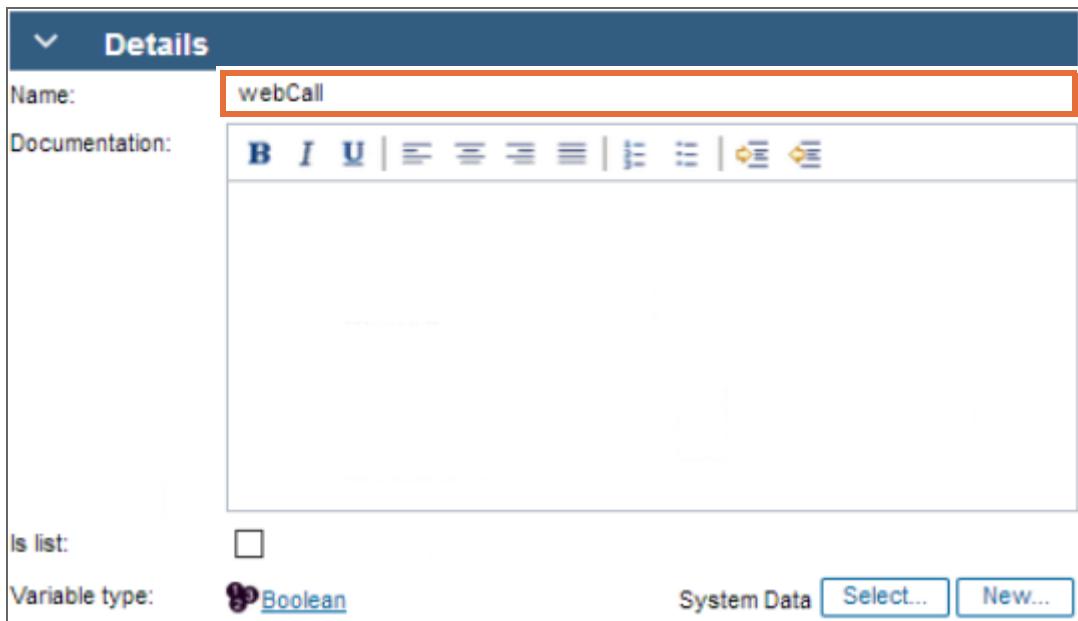


- \_\_\_ c. In the **Details** section, scroll to the bottom and examine the WSDL URL that you used earlier to create the web service.



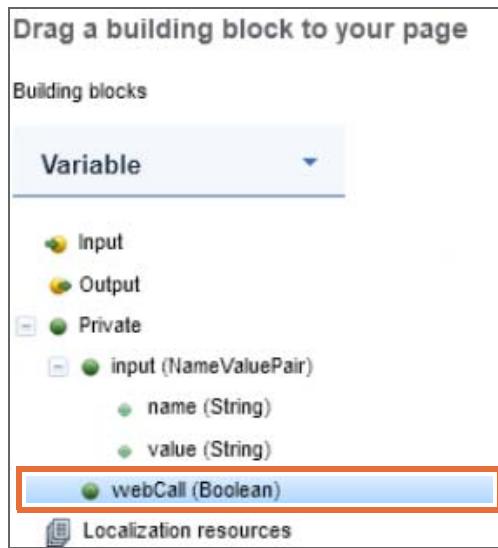
### Part 3: Test the integrations

- \_\_\_ 1. Modify the Cancel Hiring Request Test service to cancel an instance with a UCA or a web service call.
  - \_\_\_ a. Open the **User Interface > Cancel Hiring Request Test** client-side human service in the Process Designer.
  - \_\_\_ b. Add a private variable: `webCall` (Boolean)

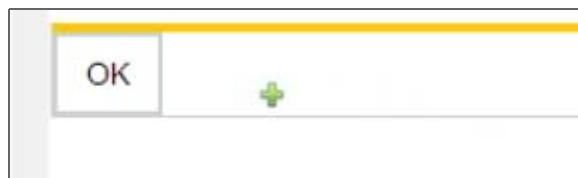


- \_\_\_ c. Click the **Coaches** tab and click the **Test Cancellation** coach.

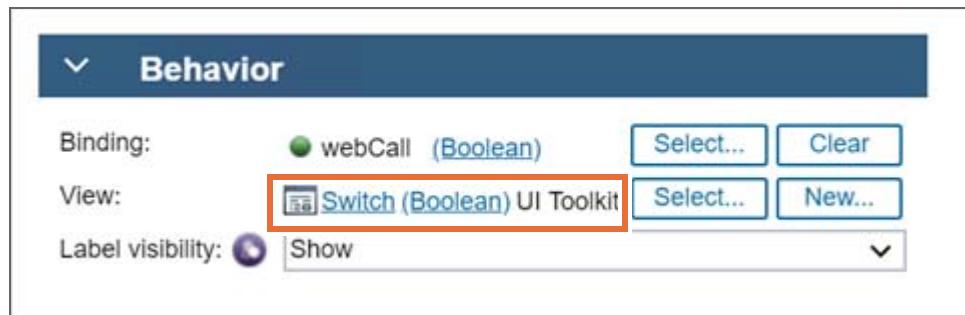
- \_\_ 2. Create a control on the page to allow users to select the type of test to perform.
- \_\_ a. Select **Variable** in the **Building blocks** panel and select **Private > webCall**



- \_\_ b. While holding down the left mouse button, drag the webCall variable to the top of the **OK** button. Let go of the mouse button when the top border displays a gold highlight.

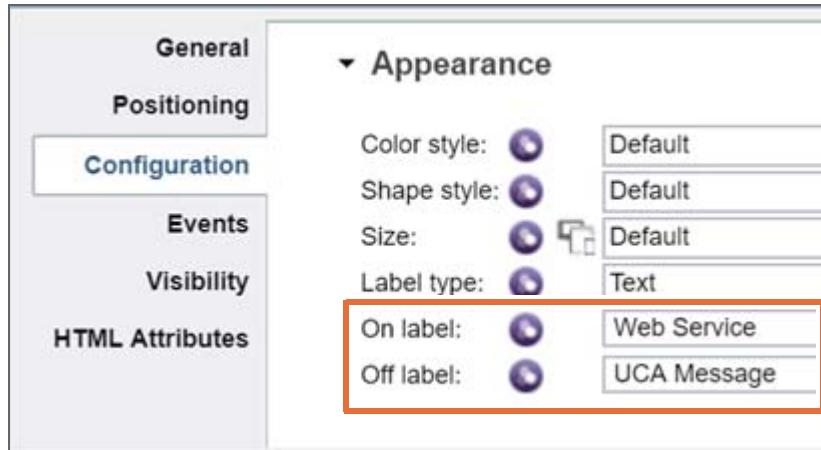


- \_\_ c. Change the **Web Call** control label to **Type of Test**.
- \_\_ d. In the **General > Behavior** section, click **Select** for View and change the type to **Switch**.



- \_\_ e. Click the **Configuration** menu and expand **Appearance**.
- \_\_ f. Set **On label:** Web Service

- \_\_ g. Set **Off label:** UCA Message



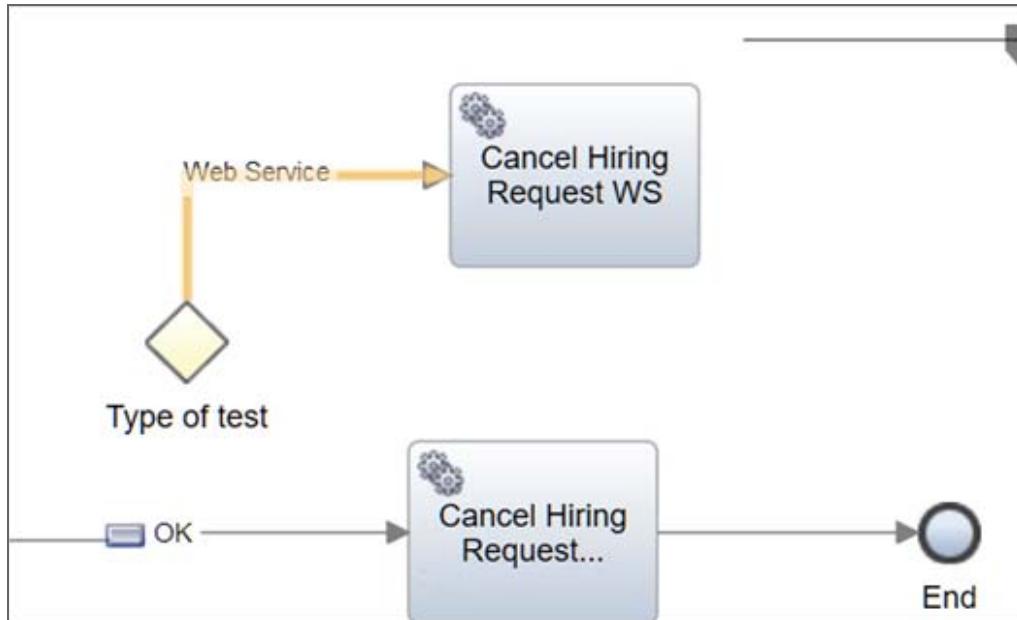
- \_\_ h. Save your work.
3. Add the **Cancel Hiring Request WS** service to the service flow.
- Click the **Diagram** tab.
  - Drag a **service** from the palette right above the Cancel Hiring Request service step on the canvas.
  - Name it: Cancel Hiring Request WS
  - Under the **Implementation > Behavior** section, ensure that **Call a service** is selected.
  - Click **Select** and under the Integration Service category, select **Call Cancellation Service**.



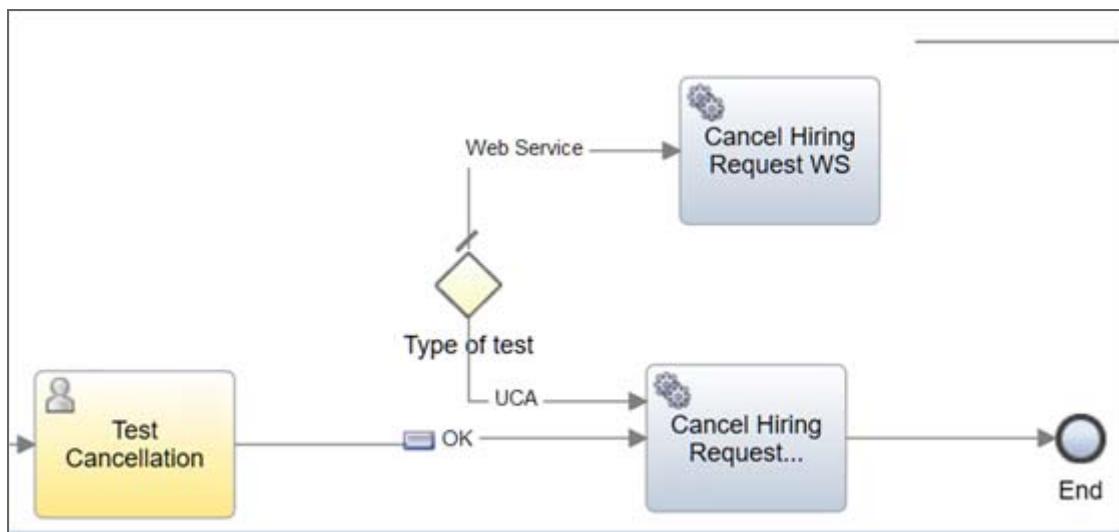
- \_\_ f. Open the **Data Mapping** properties menu, and map the `tw.local.input` variable to the `requestId` input variable.



- \_\_\_ 4. Route the flow dependent on which type of test the user selects.
  - \_\_\_ a. Add an **Exclusive Gateway** to the right of the Test Cancellation coach, and name it: Type of test
  - \_\_\_ b. Create a flow from the gateway to **Cancel Hiring Request WS**, and name this flow: Web Service. Select the **Name visible** checkbox.

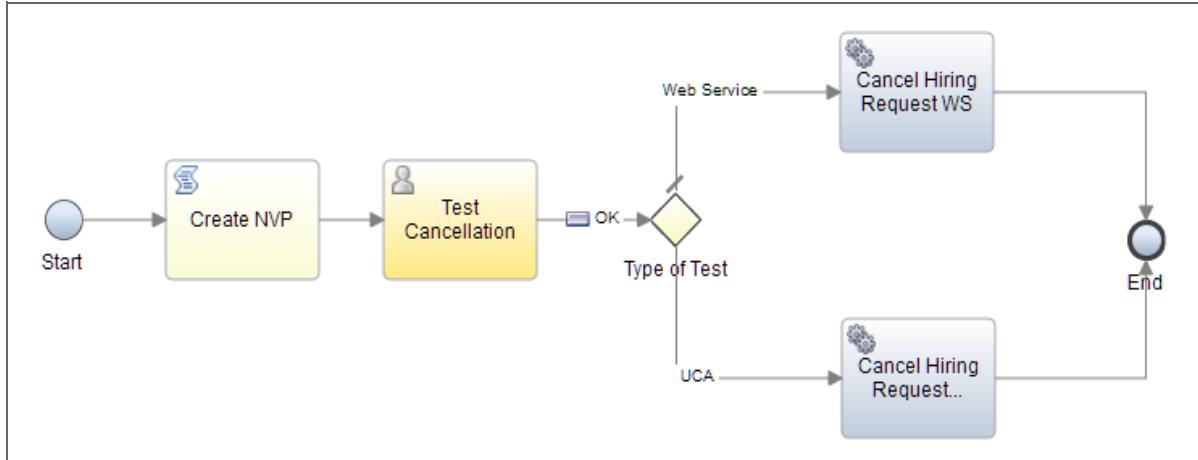


- \_\_\_ c. Create a second flow from the gateway to the **Cancel Hiring Request Trigger** service, and name this flow: UCA. Select the **Name visible** checkbox.

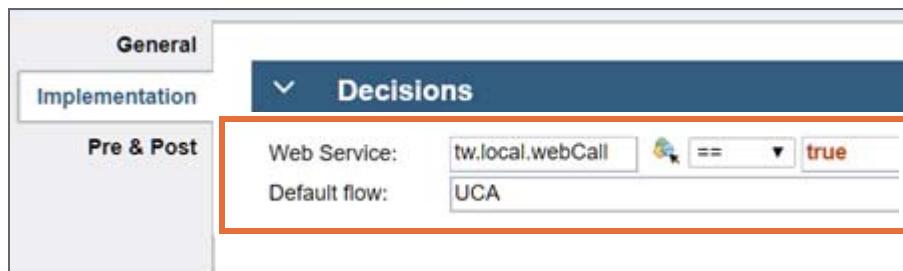


- \_\_\_ d. Delete the flow from the **Test Cancellation** coach to the **Cancel Hiring Request Trigger** Service.
- \_\_\_ e. Connect the flow from the **Test Cancellation** coach to the **gateway**.

- \_\_ f. Connect a flow from **Cancel Hiring Request WS** to the **End** event.

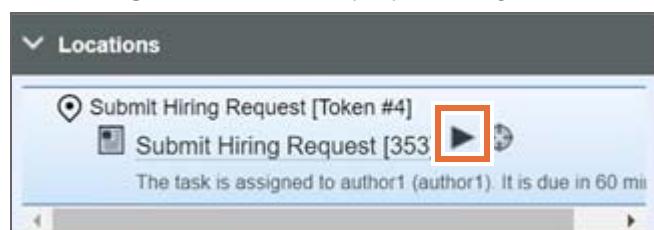


- \_\_ 5. Configure the gateway.
- \_\_ a. Click the **Type of test** gateway. Click the **Implementation** properties menu.
  - \_\_ b. Implement the decision gateway by making the **UCA** flow the Default Sequence Flow. This flow connects to the Cancel Hiring Request Trigger service.
  - \_\_ c. Enter `tw.local.webCall` in the Web Service path, select `==` from the operands menu, and enter `true` in the last field.



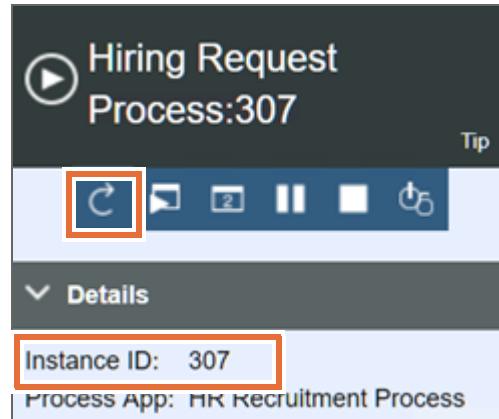
- \_\_ d. Save the service.
- \_\_ 6. Test the outbound web service call.

  - \_\_ a. In the Process Designer, open the **Hiring Request Process**.
  - \_\_ b. Create an instance of the process by clicking **Run**.
  - \_\_ c. Run the **Submit Hiring Request** activity by clicking **Start**.

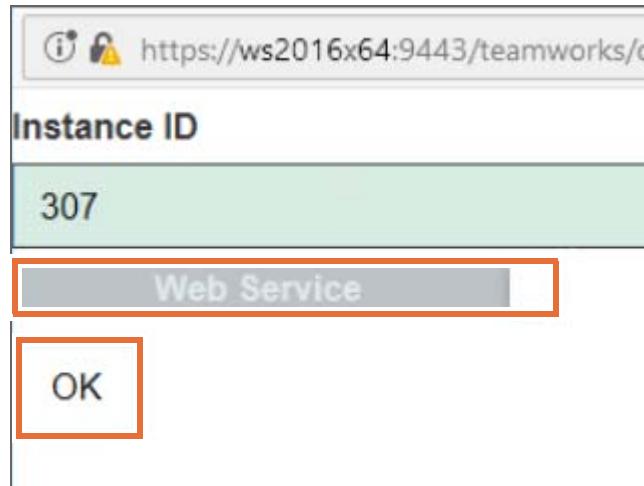


- \_\_ d. Use the following values for the variables on the form:
  - **Position Details > Job level** set to Manager
  - **Recruiting Details > New Position** set to **false** (not selected)

- Compensation Details > Salary to offer set to 100000
- \_\_\_ e. Click **Submit**.
- \_\_\_ f. Close the browser window.
- \_\_\_ g. Refresh the instance details until you see that the **Override Hire Request** task appears in the Locations section of the information window.
- \_\_\_ h. Note the **Instance ID** in the Details section.

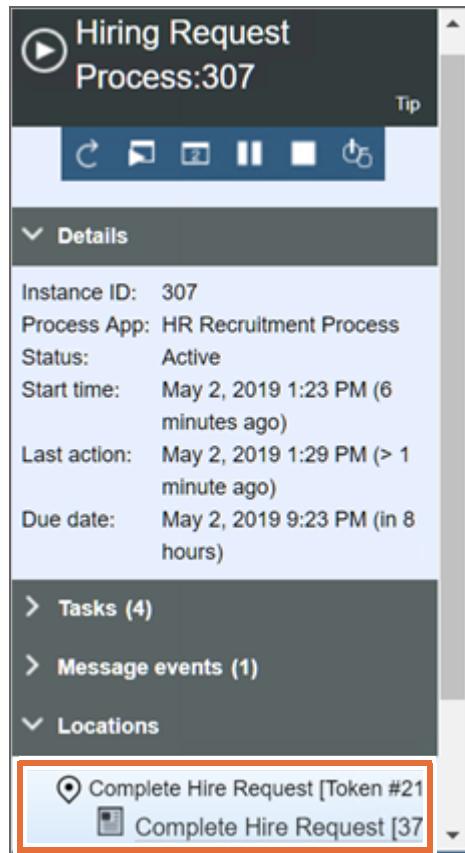


- \_\_\_ i. Open the **Cancel Hiring Request Test** client-side human service.
- \_\_\_ j. Click **Run**.
- \_\_\_ k. Enter the Instance ID and then click the **Web Service** option.
- \_\_\_ l. Click **OK**.



- \_\_\_ m. Close the service window and return to the **Hiring Request Process**.

- \_\_\_ n. Refresh the instance details window. Verify that the message event consumes all the tokens inside the linked process and moves the process down the message event flow. The instance moves from the **Override Hire Requests** task and is at the **Complete Hire Request** step. Feel free to complete the task.



You have completed this exercise.

In the first part of the exercise, you created an inbound integration by using an existing generic system service to implement the service. You viewed the WSDL in the browser. You then created an outbound web service to test the inbound web service integration. You configured the outbound web service, discovered the endpoint methods, and selected the method to call. Finally, you modified an existing test harness to call both the internal UCA and the outbound web service to test the ability to cancel an in-flight process.

## End of exercise

# Exercise 14. Handling content events in a process

## Estimated time

01:30

## Overview

In this exercise, you learn how to implement a content event in a process.

## Objectives

After completing this exercise, you should be able to:

- Use the CMIS capabilities of IBM Business Automation Workflow
- Implement a content event in a process
- Use the BPM document store to add a document to a process
- Use the Target Object Store to share documents between a BPM solution and a Case solution
- Build a simple HR Case solution and create an activity to start the Hiring Request Process in the HR BPM solution
- Demonstrate integration between the Case solution and the BPM solution

## Introduction

In this exercise, you create a content start event that starts the Hiring Request Process when a document is received. You also create a startable service that provides the capability to add a document to the BPM document store.

After building the solution to use the BPM document store, you then configure the solution to use the Target Object Store and demonstrate sharing documents between a Case solution and a BPM solution. You then build a simple Case solution and add the Hiring Request Process as an activity. You also demonstrate the integration between a Case solution and the BPM solution by adding a document to the Case and verifying that the Hiring Request Process was started.

## Requirements

Successful completion of the prior exercise is required. If you did not complete the prior exercise, you can import the solution for the prior exercise to perform this exercise.

## Index

[Part 1: Implement a document start event on a process by using the BPM document store](#)

[Part 2: Configure the Document Viewer to use the Target Object Store \(TOS\)](#)

[Part 3: Build Case solution](#)

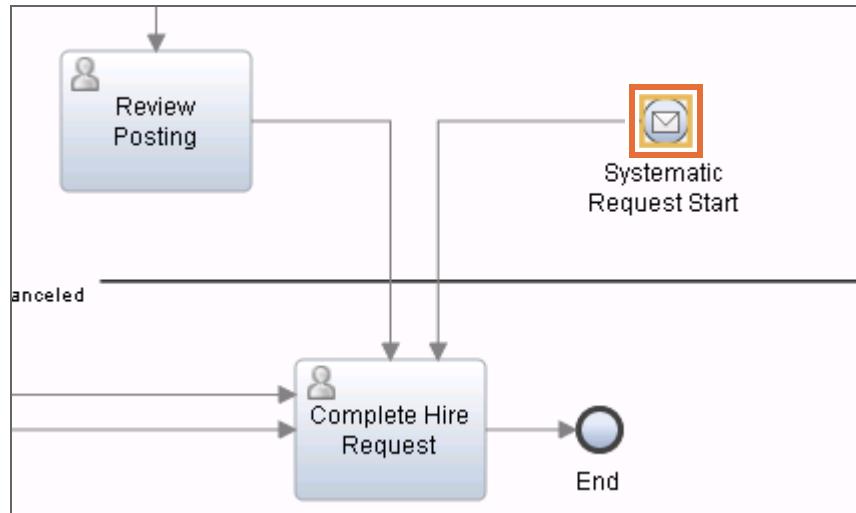
[Part 4: Test your solution](#)

[Part 5: Export the process application](#)

## Exercise instructions

### Part 1: Implement a document start event on a process by using the BPM document store

- \_\_\_ 1. Change the name of the start event on the process.
- \_\_\_ a. Open the **HR Recruitment Process** process application.
- \_\_\_ b. Open the **Hiring Request Process**. Click the **Systematic Request Start** event.



View the process on the **Definition** tab. The process has a Systematic Request Start that creates an instance of the process on a periodic basis. Management wants an employee application document that is uploaded to the repository to automatically trigger an instance of the process.

- \_\_\_ c. From the **General** properties menu, change the name of the event to **Hiring Request Document Start**.



- \_\_\_ d. Save your work.

- \_\_ 2. Create a dependency to the Content Management toolkit snapshot.
- \_\_ a. Open the library on the left and click the **plus sign (+)** next to the Toolkits category, then click the **8.6.0.0** snapshot under Content Management.

The screenshot shows the IBM Business Automation Workflow Designer. On the left, there's a sidebar with categories like HR Recruitment Process, Processes, User interface, Services, Events, Teams, Data, Performance, Files, and Toolkits. The 'Toolkits' category is currently selected, indicated by a blue bar at the bottom of the sidebar. To the right, there's a main workspace and a floating modal window titled 'Add dependency'. This modal lists several toolkits with their versions: Coaches (deprecated) 8.6.0.0, Content Management 8.6.0.0 (which is highlighted with a red box), Dashboards 8.6.0.0, Responsive Coaches (deprecated) 8.6.0.0, Responsive Portal Components 8.6.0.0, SAP Guided Workflow (deprecated) 8.6.0.0, System Governance 8.6.0.0, and another Content Management entry 8.6.0.0\_TC. Each toolkit entry has a small circular badge with the number '1' next to it. At the bottom of the sidebar, there's a blue button with a white plus sign (+) inside a square, which is also highlighted with a red box.

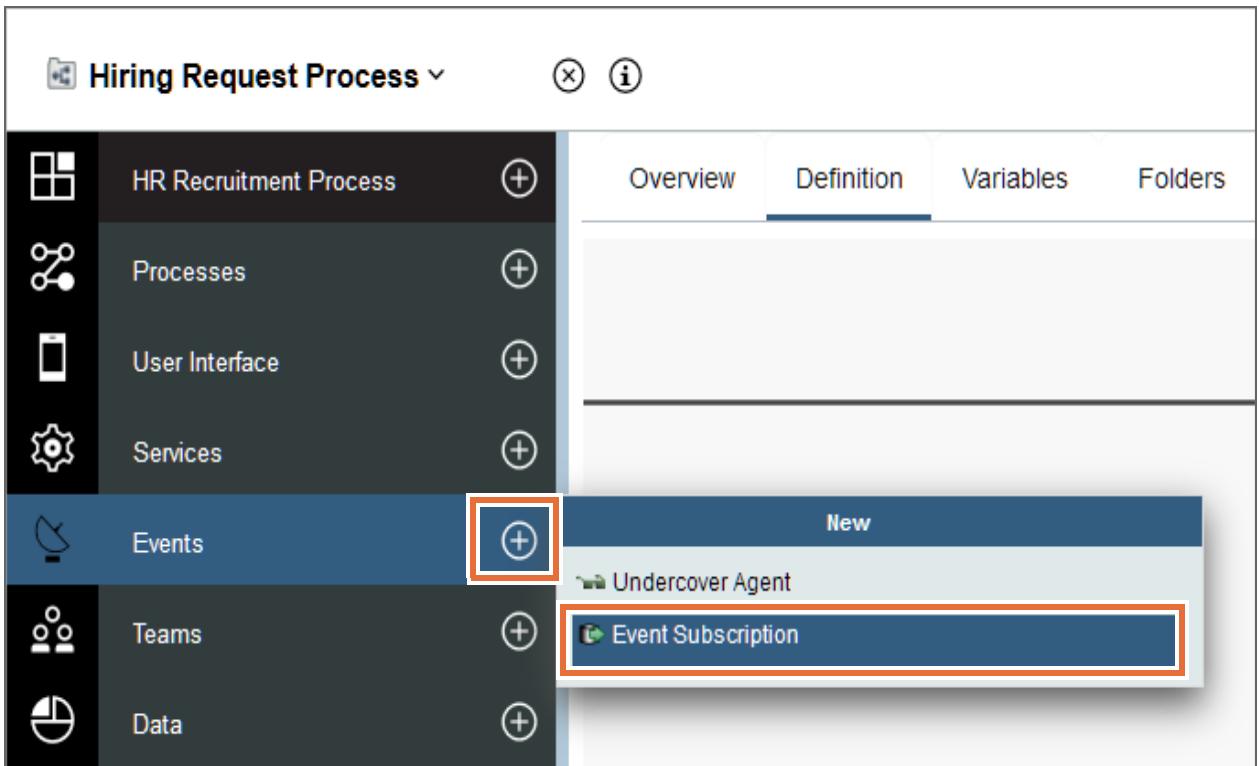


### Information

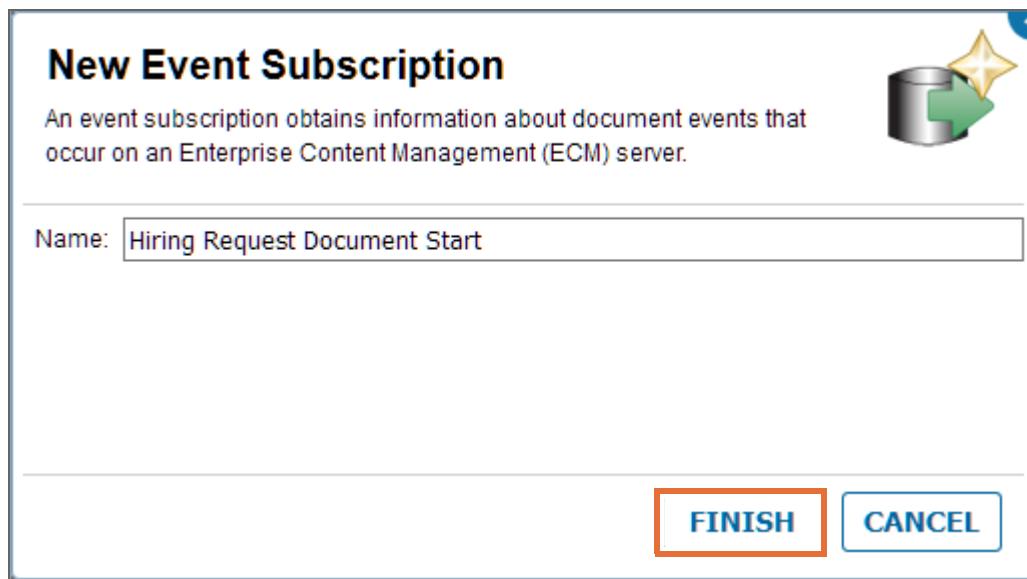
In IBM Business Automation Workflow, all artifacts are generally supported in the traditional WebSphere runtime environment. However, not all artifacts are supported in the container runtime environment, such as most deprecated artifacts (like BPDs, integration services and some other services). You can use either version 8.6.0.0 or version 8.6.0.0\_TC of the content management toolkit in a traditional environment. However, you must use version 8.6.0.0\_TC of the toolkit when implementing in a container runtime environment.

\_\_\_ 3. Create a new event subscription.

\_\_\_ a. In the library, click the plus sign (+) next to Events and then click **Event Subscription**.

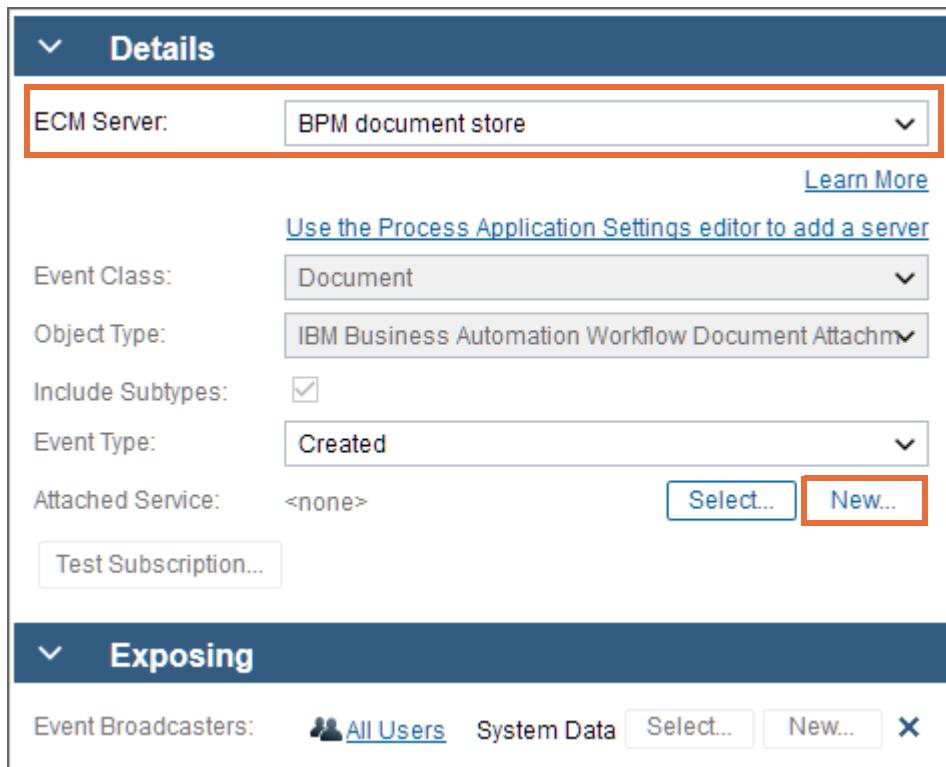


\_\_\_ b. Name the event subscription **Hiring Request Document Start** and click **FINISH**.



\_\_\_ c. The event subscription opens in Process Designer. On the right, in the **Details** section, select **BPM document store** for the **ECM Server** setting. Leave the rest of the settings as the default.

- \_\_\_ 4. Create and attach the Service Flow to invoke the Undercover agent.
- \_\_\_ a. At the bottom of the Details section, click the **New** button next to Attached Service.



**Details**

ECM Server: BPM document store [Learn More](#)  
[Use the Process Application Settings editor to add a server](#)

Event Class: Document

Object Type: IBM Business Automation Workflow Document Attachm

Include Subtypes:

Event Type: Created

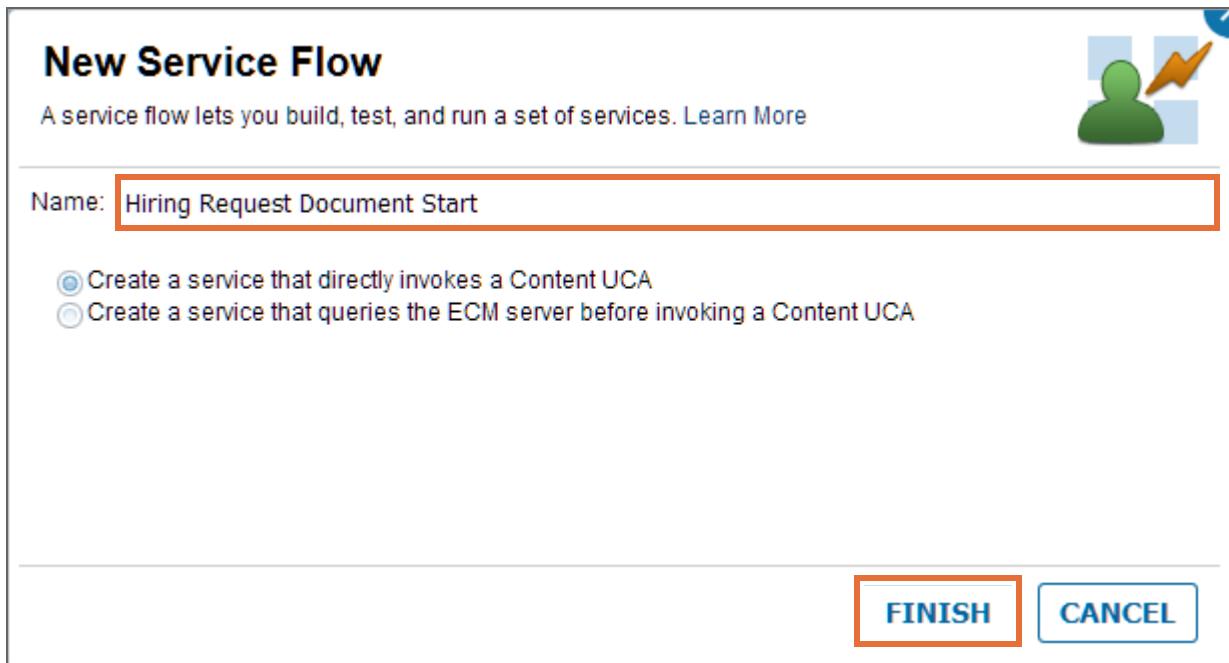
Attached Service: <none> [Select...](#) **New...**

[Test Subscription...](#)

**Exposing**

Event Broadcasters: All Users System Data [Select...](#) [New...](#) [X](#)

- \_\_\_ b. Name the New Service Flow as Hiring Request Document Start.
- \_\_\_ c. Leave the options as default and click **FINISH**.



## New Service Flow

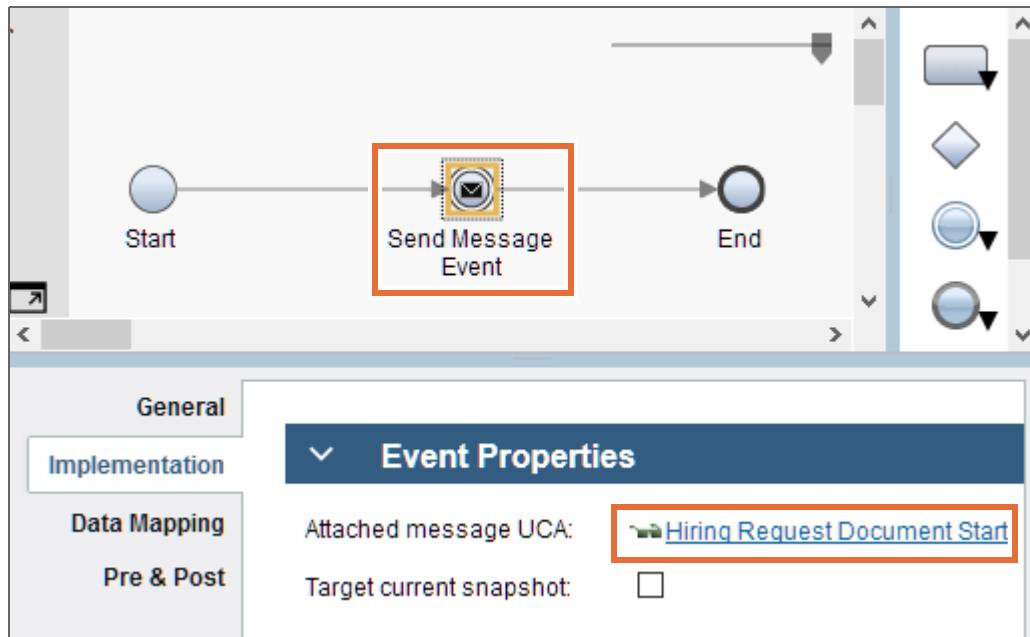
A service flow lets you build, test, and run a set of services. [Learn More](#)

Name: **Hiring Request Document Start**

Create a service that directly invokes a Content UCA  
 Create a service that queries the ECM server before invoking a Content UCA

**FINISH** **CANCEL**

When you chose the first option, the system creates a fully implemented service and opens it in IBM Process Designer. The service consists of a Start event, an End event, and an intermediate Send Message Event that invokes the new content undercover agent (UCA).



- \_\_\_ d. Click the **Variables** tab.

The system created an input variable `contentEvent (ECMContentEvent)`. The system requires this type of variable when the content start event is triggered. When you trigger the UCA, it creates an instance of a process.

The screenshot shows the 'Variables' tab in the IBM Process Designer interface. The 'Input' section is expanded, showing a list of variables under the heading 'contentEvent (ECMContentEvent)'. The variables listed are: serverName (String), repositoryId (ECMID), eventClass (ECMEventClass), objectTypeId (ECMID), eventType (ECMEventType), objectId (ECMID), relatedObjectId (ECMID), relatedObjectTypeId (ECMID), and relatedObjectClass (ECMEventClass). The 'Output', 'Private', 'Exposed Process Variables', and 'Localization Resources' sections are also visible but collapsed.

- \_\_ e. Return to the **Diagram** tab.
- \_\_ f. Click the **Send Message Event** on the canvas.

Open the **Data Mapping** properties menu. The `tw.local.contentEvent` variable is already mapped for you.

The screenshot shows the 'Input Mapping' section of the Data Mapping properties menu. A variable named `tw.local.contentEvent` is selected and highlighted with a red box. A tooltip labeled 'Input (ECMContentEvent)' is displayed next to the variable name.

- \_\_ g. Open the **Implementation** properties menu and click the **Hiring Request Document Start** link for the undercover agent to open the UCA.

The screenshot shows the 'Event Properties' section of the Implementation properties menu. The 'Attached message UCA:' field contains a link labeled 'Hiring Request Document Start', which is highlighted with a red box.

The Undercover Agent is already configured for you. Under the Scheduler section, you see that the Event Marker is **Content**, which makes it available to implement a Content Start Event. Under the Details section, you see the same business object that is used in the service flow. The Undercover Agent requires an input variable of type `ECMContentEvent`, and this variable is sent to the process that triggers the Undercover Agent.

The screenshot shows the Scheduler properties menu. The 'Scheduler' section has 'Schedule Type: On Event' and 'Event Marker: Content' (highlighted with a red box). The 'Details' section has 'Queue Name: Async Queue' and 'Implementation: Variable' (highlighted with a red box). The 'Variable Type' field is set to `ECMContentEvent Content Management` (highlighted with a red box). The 'Parameter Mapping' section shows a warning symbol and a tooltip 'Input (ECMContentEvent)'.

The warning symbol is present to indicate that no mapping is implemented. With no mapping implemented, the default mapping takes place.

- \_\_\_ h. Always change the event message to better identify UCA events that occur in the logs. You should always change the event message identifiers from the default hash to a relevant event label when creating undercover agents. At the bottom, in the Event section, change the Event Message for the UCA to **HiringRequestDocumentStart**.



- \_\_\_ i. Save your work.
- \_\_\_ 5. Configure the start event
  - \_\_\_ a. Return to the **Hiring Request Process** and open the **Definition** tab.
  - \_\_\_ b. Select the **Hiring Request Document Start** event.

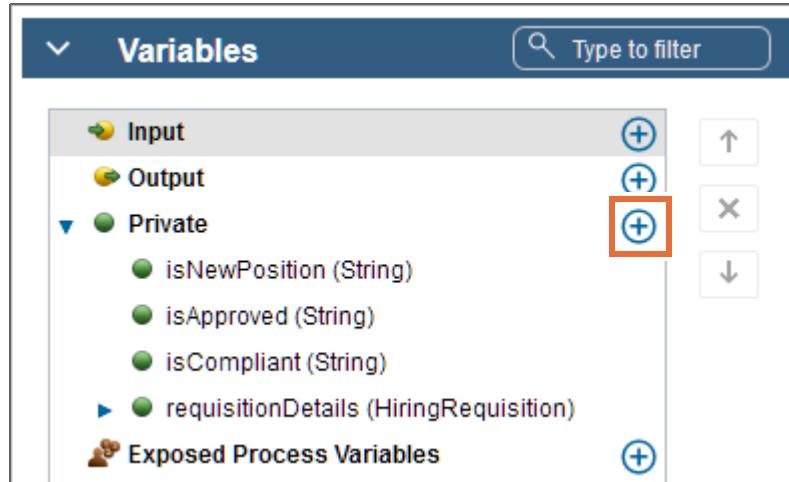


- \_\_\_ c. Click the **Implementation** properties menu. Change the Start Event Type to **ECM Content**.
- \_\_\_ d. Select the **Hiring Request Document Start** for the attached undercover agent.

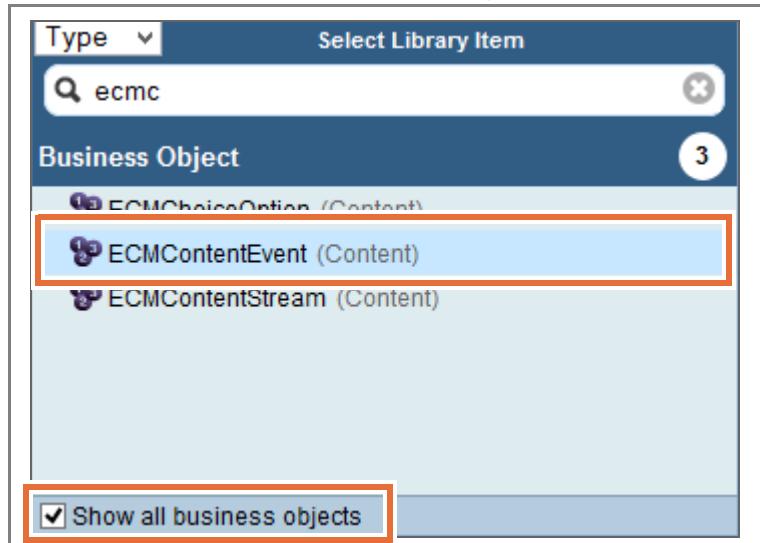
A screenshot of the 'Implementation' properties dialog for a process step. On the left, there is a vertical navigation bar with tabs: General, Implementation (which is selected), Data Mapping, Pre & Post, and Tracking. The main area contains several sections:
 

- Start Event Type:** A dropdown menu is set to 'ECM Content', which is highlighted with a red border.
- Event Properties:** This section includes fields for 'Attached content UCA' (containing 'Hiring Request Document Start', also highlighted with a red border), 'Condition' (set to '1'), and 'Consume message' (an unchecked checkbox).

- \_\_ e. Click the **Variables** tab. Click the **plus sign (+)** to create a private variable.



- \_\_ f. Name the variable `content`. Next to the Variable type option, click **Select**.
- \_\_ g. Select the **Show all business objects** checkbox, then select **ECMContentEvent**. The content (`ECMContentEvent`) variable is complete.



- \_\_ h. Return to the **Definition** tab.

- \_\_ i. Open the **Hiring Request Document Start** event and go to the **Data Mapping** properties menu. Map the output of the start event to the `tw.local.content` private variable. There is no need to map the parameters.

**Output (ECMContentEvent)** ➔ `tw.local.content`

- `serverName (String)` ➔ [ ]
- `repositoryId (ECMID)` ➔ [ ]
- `eventClass (ECMEvent...)` ➔ [ ]
- `objectTypeId (ECMID)` ➔ [ ]
- `eventType (ECMEventTy...)` ➔ [ ]
- `objectId (ECMID)` ➔ [ ]

- \_\_ j. Save your work.
- \_\_ 6. Test your solution by running the Undercover Agent.
- \_\_ a. Return to the Undercover Agent by clicking **Events > Hiring Request Document Start UCA** in the Process Library.

**Events**

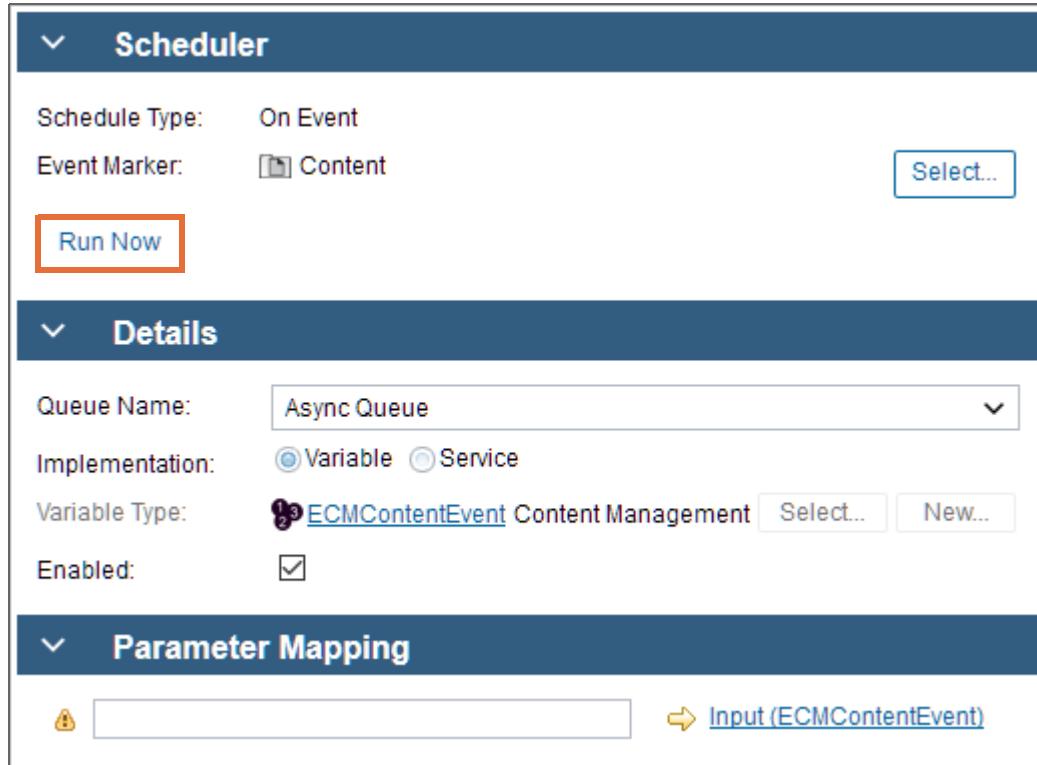
- Teams**
- Data**
- Performance**
- Files**

Type: Events

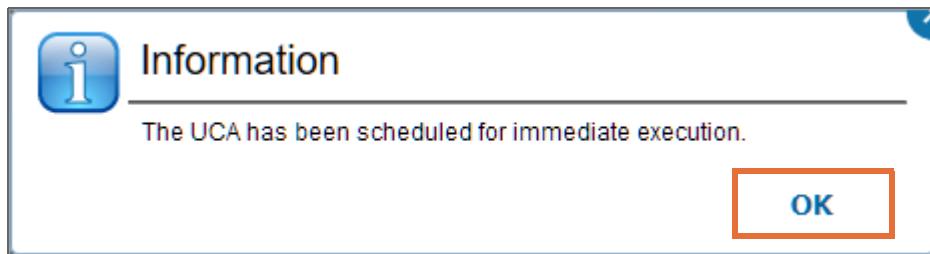
**Event Subscription**

- Hiring Request Document Start
- Undercover Agent**
- Cancel Hiring Request
- Hiring Request Document Start**
- Systematic Request (UCA)

- \_\_\_ b. Click **Run Now**.



- \_\_\_ c. Click the **OK** button on the Information window when it confirms that the UCA is scheduled for immediate execution.

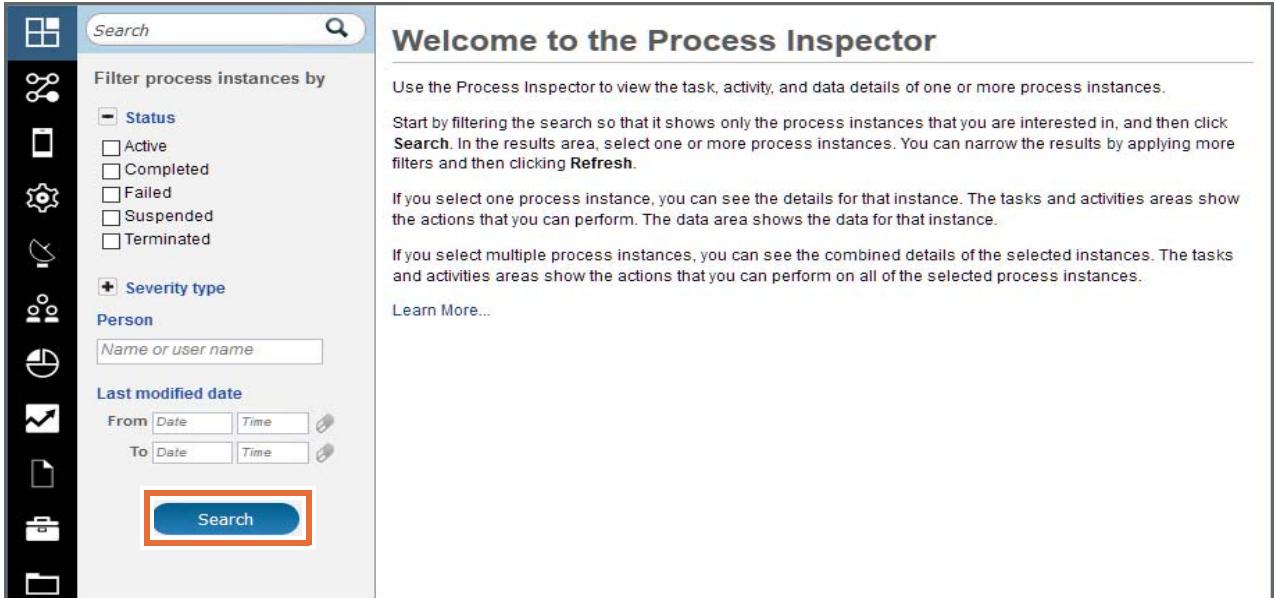


- \_\_\_ d. Verify the creation of a Hiring Request Process instance by clicking the **Inspector** tab, then click the **Search** icon.



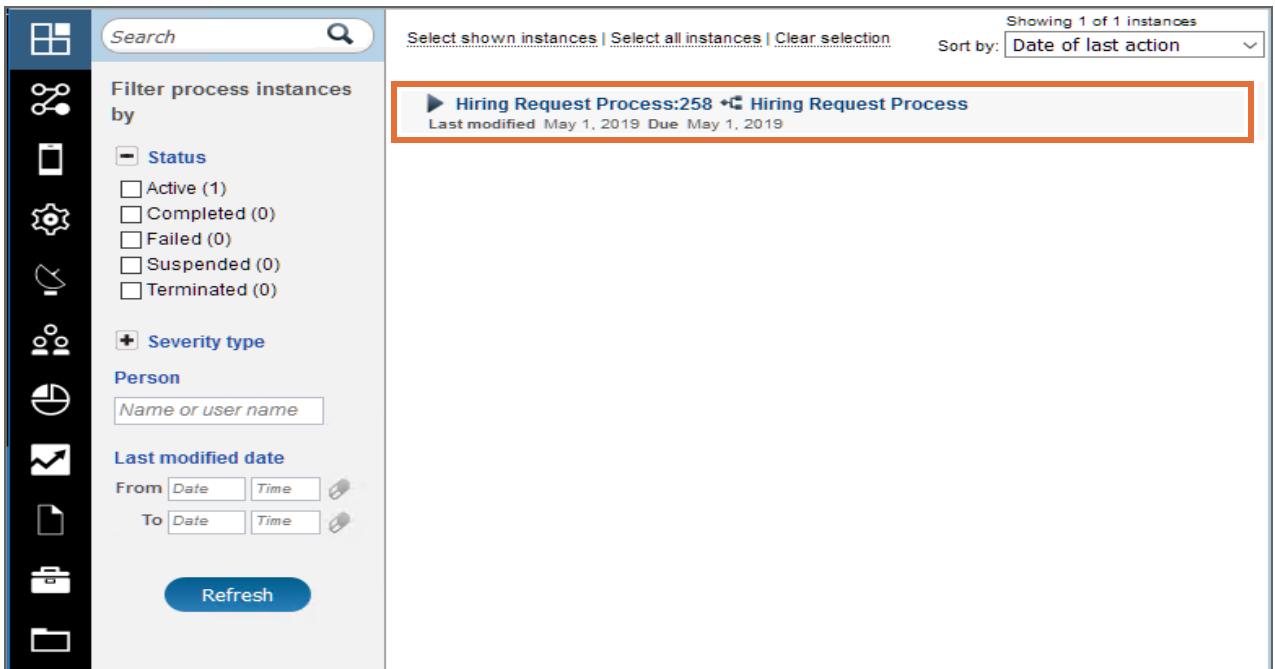
Clicking the Search icon starts the Process Inspector.

- \_\_ e. If the instance you created is not shown, click the **Search** button.



The screenshot shows the 'Welcome to the Process Inspector' page. On the left, there is a sidebar with various icons for filtering process instances by status (Active, Completed, Failed, Suspended, Terminated), severity type, person, and last modified date. Below these filters is a large blue 'Search' button, which is highlighted with a red rectangular border. The main content area is titled 'Welcome to the Process Inspector' and contains instructions on how to use the search feature to view task, activity, and data details of one or more process instances. It also provides information on selecting multiple instances and viewing combined details.

You created an instance of the process, but you didn't upload a document to create an instance of the process. You create the client-side human service to upload your document next. Depending on the number of process instances, the screen capture might not match.



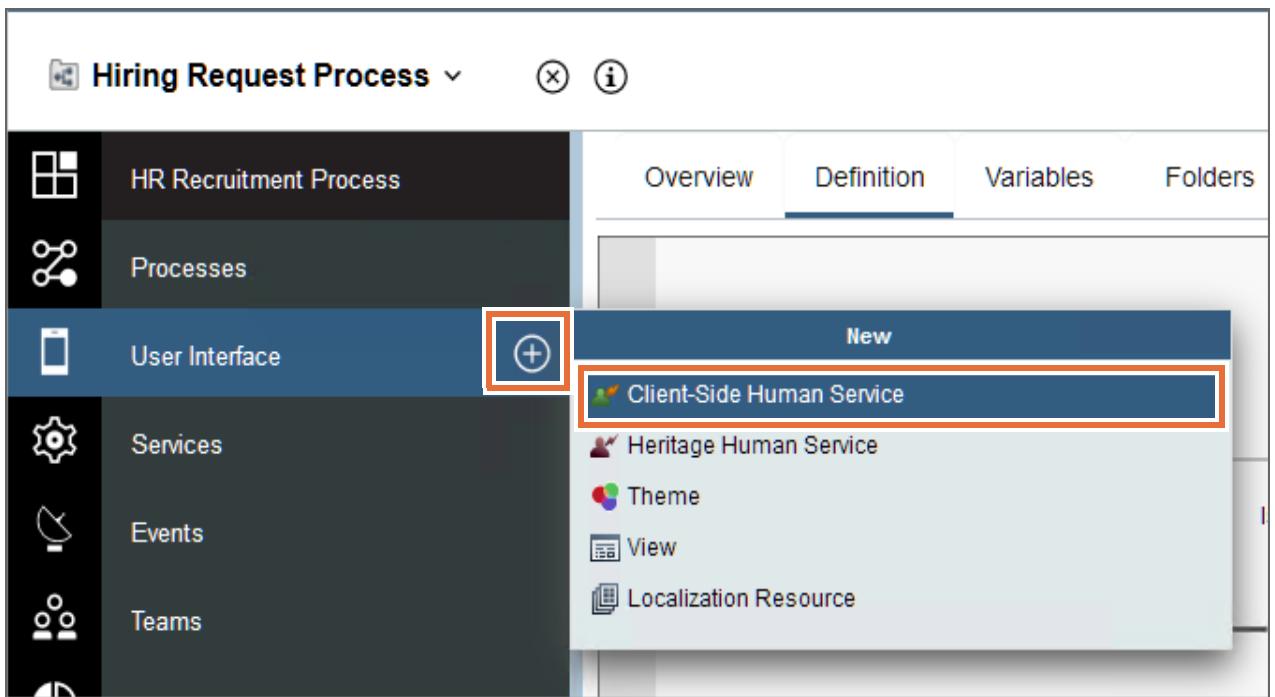
The screenshot shows the 'Process Inspector' interface after a search has been performed. The sidebar remains the same, but the main content area now displays a list of process instances. A single instance is selected and highlighted with a red rectangular border. The selected instance is labeled 'Hiring Request Process:258' and includes a small icon of a person. Above the list, there are buttons for 'Select shown instances', 'Select all instances', and 'Clear selection', along with a 'Sort by:' dropdown menu set to 'Date of last action'. The status of the selected instance is listed as 'Active (1)'.

- \_\_ 7. Create a client-side human service to trigger the UCA  
\_\_ a. Switch back to the **Designer** tab.



The screenshot shows the application's navigation bar at the top. It features tabs for 'DESIGNER' and 'INSPECTOR', with the 'DESIGNER' tab being the active one and highlighted with a red rectangular border. To the right of the tabs are several icons for search, refresh, and other functions, followed by a link 'Return to Workflow Center'.

- \_\_ b. In the IBM Process Designer library, click the **plus sign (+)** next to User Interface, and then click **Client-Side Human Service**.



- \_\_ c. Name the service **Hiring Request From Document**. Select the **Intended for use on multiple devices** checkbox and click **FINISH**.

This is a configuration dialog for a new client-side human service. At the top, it says 'New Client-Side Human Service'. Below that is a description: 'The client-side human service enables you to build, test, and run interactive tasks, dashboards, or user interfaces for process instances, and is optimized to run in a web browser.' To the right is a small icon of a person with a lightning bolt. The main form has fields for 'Name' (set to 'Hiring Request From Document') and checkboxes for 'Intended for use on multiple devices' (which is checked) and 'Use as a nested service' (which is unchecked). At the bottom are 'FINISH' and 'CANCEL' buttons, with 'FINISH' also highlighted with a red box.



## Information

If you forget to select the checkbox during the initial creation, or if you decide later to change to a responsive coach, you can change the service. You can mark any Client-side Human Service as a responsive coach service in the **Overview** tab of the service. Select the **Intended for use on multiple devices** checkbox in the Common section. By selecting **Intended for use on multiple devices** checkbox, you are enabling the responsive capabilities on the coach.

The screenshot shows the 'Overview' tab selected in the top navigation bar. Below it, the 'Common' section is expanded. The 'Name:' field contains 'Hiring Request From Document'. The 'Label:' field has two buttons: 'Select...' and 'Clear'. The 'Documentation:' field contains a rich text editor toolbar with icons for bold (B), italic (I), underline (U), and various alignment and list options. At the bottom of the 'Common' section, there is a checkbox labeled 'Intended for use on multiple devices:' which is checked. The entire 'Common' section is highlighted with a red border.

- \_\_\_ d. Click the **Variables** tab.
- \_\_\_ e. Create a private variable and name it document (ECMDocumentInfo).

The screenshot shows the 'Details' dialog box with the following fields:

- Name:** document (highlighted with a red box)
- Documentation:** Rich text editor toolbar
- Is list:**
- Variable type:** [13 ECMDocumentInfo Content Management](#) (highlighted with a red box)
- Select...** and **New...** buttons

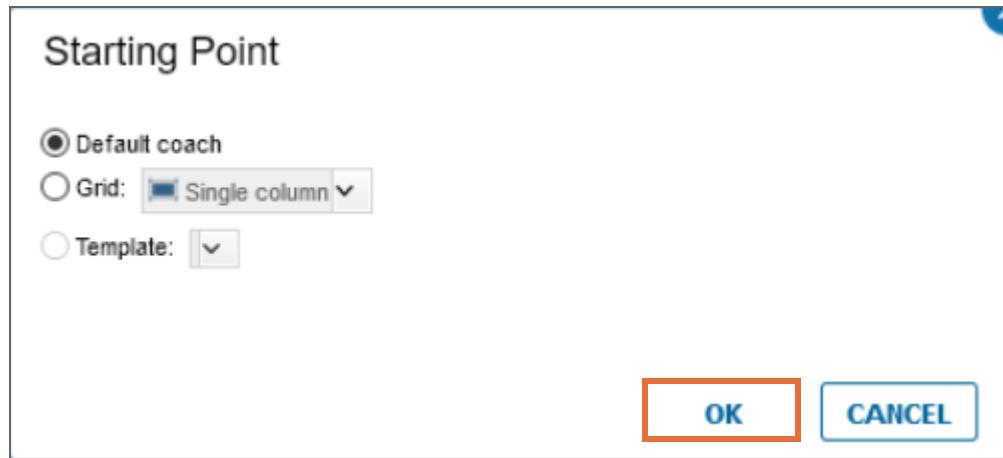
Be sure to choose the ECMDocumentInfo variable type and not the ECMDocument variable type.

- \_\_\_ f. Save your work.
- \_\_\_ 8. Create the coach
  - \_\_\_ a. Click the down arrow next to Coach and select **Coach**.

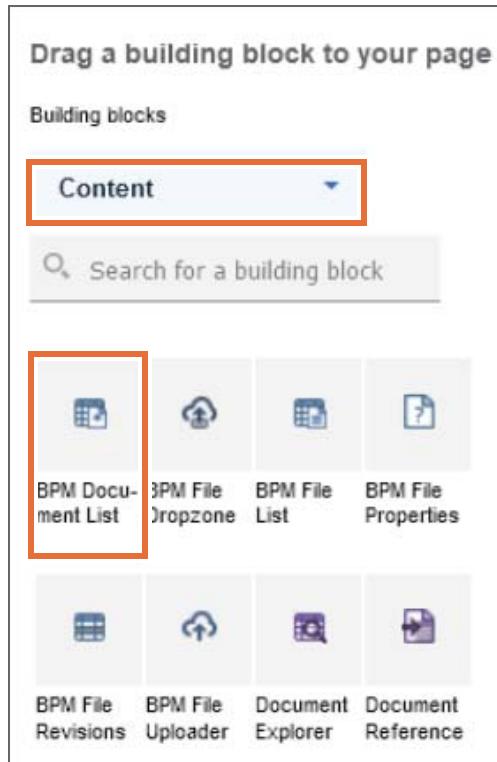
The screenshot shows the 'Variables' tab with the following interface:

- Coach:** Coach
- Variables** tab selected
- Variables** list:
  - Input**
  - Output**
  - Private** (selected, expanded)
    - document (ECMDocumentInfo) (highlighted with a red box)
  - Exposed Process Variables**
  - Environment Variables**
  - Localization Resources**
- Type to filter** input field
- Buttons for **+**, **↑**, **X**, and **↓** (for reordering variables)

- \_\_ b. In the **Starting Point** window, ensure that the **Default coach** is selected, and click **OK**.



- \_\_ c. Select **Content** in the **Building blocks** panel and select **BPM Document List**.



When you added the Content Management Toolkit as a dependency, the palette was updated to include controls from the Content Management Toolkit.

- \_\_ d. While holding down the left mouse button, drag the control above the **OK** button. Let go of the mouse button when the top border displays a gold highlight.

- \_\_ e. In the **General > Behavior** properties section, verify that the binding is set to **document (ECMDocumentInfo)**.

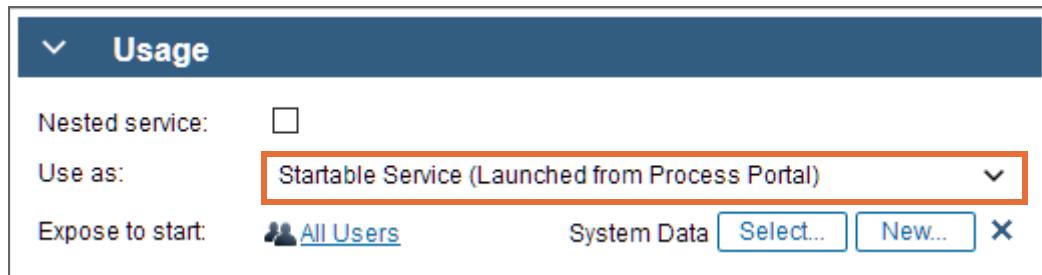
The screenshot shows the 'Behavior' properties section. The 'Binding' field is highlighted with a red box and contains the value 'document (ECMDocumentInfo)'. To the right of the binding field are 'Select...' and 'Clear' buttons. Below the binding field is a 'View:' field containing 'BPM Document List (ECMDocumentInfo) Content Management' with 'Select...' and 'New...' buttons. At the bottom is a 'Label visibility:' dropdown set to 'Show'.

- \_\_ f. Click the **Configuration** properties menu.  
 \_\_ g. Expand the Content management section and Select the **Allow create**, **Allow updates**, **Allow document deletions**, and **Confirm on deletion** check boxes.

The screenshot shows the 'Configuration' properties menu. On the left is a sidebar with tabs: General, Positioning, Configuration (selected), Events, Visibility, and HTML Attributes. The main area shows the 'Content management' section expanded, with four checkboxes checked: 'Allow create', 'Allow updates', 'Allow document deletions', and 'Confirm on deletion'. Other options like 'Allow revisions display' and 'Max results' have their checkboxes unchecked.

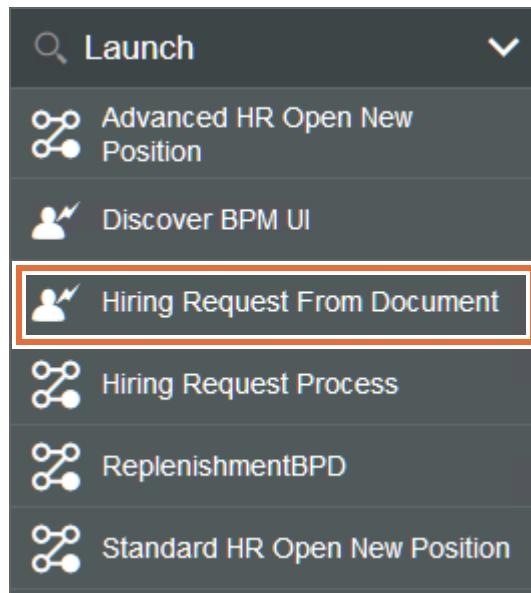
- \_\_ h. Save your work.  
 \_\_ 9. Configure the coach as a Startable Service.  
 Since the **Hiring Request From Document** client-side human service is not part of the process, there is no need to link it to an activity in the process. In order to allow Process Portal users to start the service, you configure it as a Startable Service.  
 \_\_ a. Select the **Overview** tab for the **Hiring Request From Document** client-side human service.

- \_\_\_ b. Under the Usage section, select **Startable Service (Launched from Process Portal)** for how it is to be used.



**All Users** is automatically added as the **Expose to start** selection. This allows all users of the Process Portal to run the Startable Service.

- \_\_\_ c. Save your work.
- \_\_\_ 10. Upload a document by using the coach.
  - \_\_\_ a. Open Process Portal in another tab. The Hiring Request From Document is displayed under the Launch menu.  
If asked for credentials, enter the following: **Username:** author1 **Password:** passw0rd
  - \_\_\_ b. Click the **Hiring Request From Document** startable service in the Launch menu.



- \_\_ c. In the Hiring Request From Document coach, click **Add**.

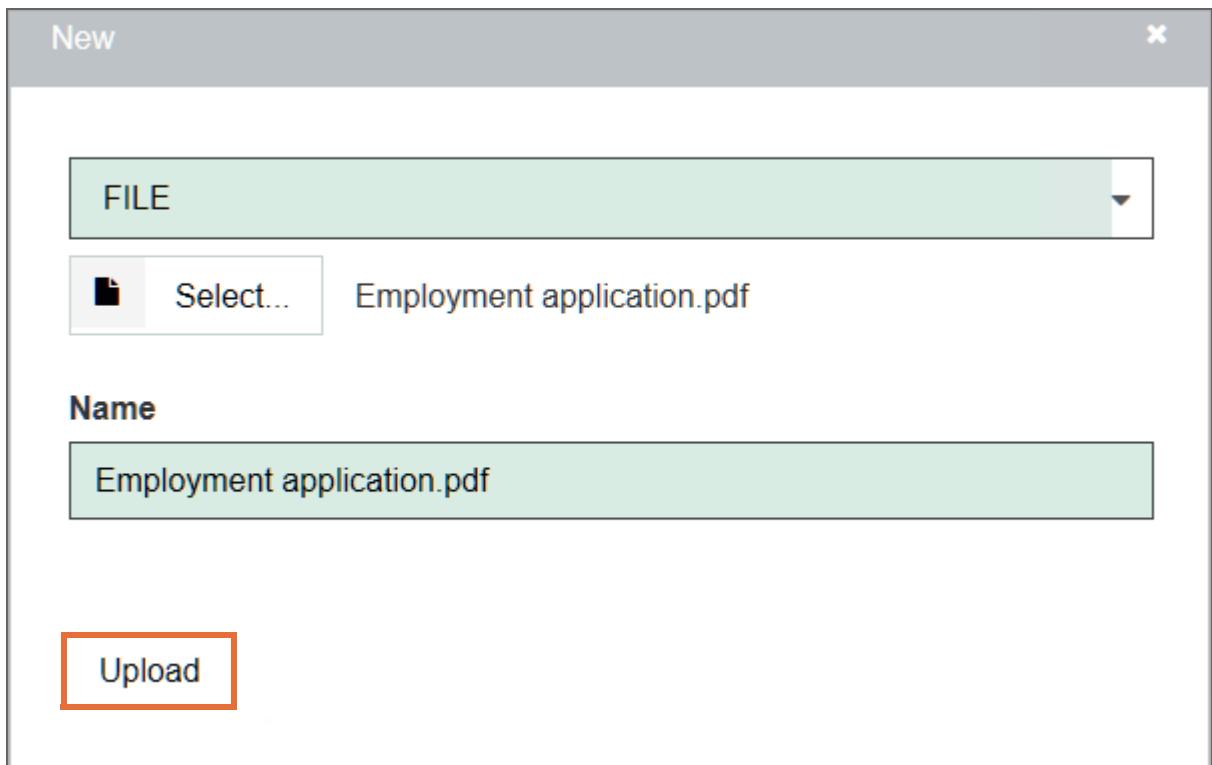


- \_\_ d. In the dialog box, click **Select** and navigate to the following file:

C:\labfiles\Support files\Ex14\Employment application.pdf

- \_\_ e. Click **Open**.

- \_\_ f. Click **Upload**.





## Troubleshooting

If the initial upload fails, verify there is a file selected. If no file displays as selected, close the window and try the operation again.

New

FILE

Select... No file selected

Name

Employment application.pdf

Upload

After the upload completes, the document list is refreshed.

Hiring Request From Document

BPM Document List

| Name                       | File name or URL           | Major Version Number | Last Modified | Modified By |
|----------------------------|----------------------------|----------------------|---------------|-------------|
| Employment application.pdf | Employment application.pdf | 1.0                  | 5/1/2019      | author1     |

OK

- \_\_\_ g. You see a notice in the lower right of the screen that you received a new task.



\_\_ 11. Verify the document start event works

- \_\_ a. Click **OK** to close the service. The Work page is displayed with a new work item.

The screenshot shows the IBM Work Center interface. At the top, there's a header with a menu icon, the word 'Work', and a summary: '1 Total Open' with counts for 'On Track', 'At Risk', and 'Overdue'. Below the header is a search bar with placeholder text 'Enter search text...' and three icons: a magnifying glass, a folder, and a refresh arrow. The main content area displays a single work item with a blue border. Inside, there's a thumbnail icon of a document, the title 'Step: Complete Hire Request', the subtitle 'Hiring Request Process:270', and the due date 'Due: May 1, 2019, 2:59:00 PM'. There are also two small buttons at the bottom left of the card.

## Troubleshooting

You might have other work items left from other exercises. If the work item is not listed, press F5 to refresh the page. If it is still not listed, restart the server.

- \_\_ b. Return to IBM Process Designer and click the **Inspector** view.

The screenshot shows the top navigation bar of the IBM Process Designer. It includes tabs for 'DESIGNER' and 'INSPECTOR', which is highlighted with a red box. To the right of the tabs are several icons: a magnifying glass, a refresh arrow, a circular arrow, and a circular arrow with a dot. On the far right, there's a link 'Return to Workflow Center'.

- \_\_ c. If the instance you created is not shown, click **Search**. If there are process instances in the list, click **Refresh**.

The screenshot shows the 'Welcome to the Process Inspector' screen. On the left, there's a sidebar with various icons and a search bar. The main area has a heading 'Welcome to the Process Inspector' and instructions for using the tool. It includes sections for filtering process instances by status (Active, Completed, Failed, Suspended, Terminated) and severity type. There are also fields for searching by person name or user name, and for specifying a last modified date range. A large red box highlights the 'Search' button at the bottom of the sidebar.

- \_\_\_ d. Click the newly created **Hiring Request Process** instance that got added to the instance page. The latest instance is the one with the highest instance ID.

The screenshot shows a list of instances. At the top, there are three buttons: 'Select shown instances', 'Select all instances', and 'Clear selection'. Below this, a single instance is listed with a red border around it. The instance name is 'Hiring Request Process:406'. To its right is a small icon of a document with a checkmark. Below the name, it says 'Last modified May 9, 2019 Due May 9, 2019'.

- \_\_\_ e. Scroll down in the right pane and expand the **Data** section. Expand the content variable. The content variable is populated, and the `objectId` is shown.

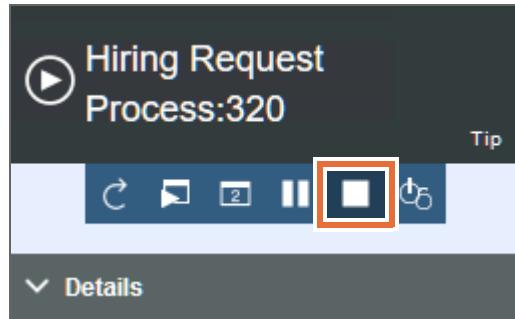
The screenshot shows the 'Data' section of an instance details view. The 'content(ECMContentEvent)' variable is expanded, showing its properties. The 'objectId(ECMID)' property is highlighted with a red border.

| Property                           | Value                                           |
|------------------------------------|-------------------------------------------------|
| eventClass(ECMEEventClass)         | Document                                        |
| objectTypeId(ECMID)                | IBM_BPM_Document                                |
| relatedObjectId(ECMID)             | <Empty>                                         |
| relatedObjectTypeid(ECMID)         | <Empty>                                         |
| relatedObjectClass(ECMEEventClass) | <Empty>                                         |
| repositoryId(ECMID)                | {10C21AF6-7D1D-4829-9F2D-835A9310916C}          |
| serverName(String)                 | EMBEDDED_ECM_SERVER                             |
| eventType(ECMEEventType)           | Created                                         |
| <b>objectId(ECMID)</b>             | <b>idd_C085756A-0000-C31F-BD55-9C4565685B52</b> |

The creation of the document in the internal document repository triggers the creation of the instance. If you created this functionality as part of a production development effort, the next step inside the process is to associate the document with the instance. You would then create the step when the instance is complete and persist the document to an ECM server.

\_\_ 12. Clean up your environment

\_\_ a. Stop the process by clicking the **Terminate** icon.

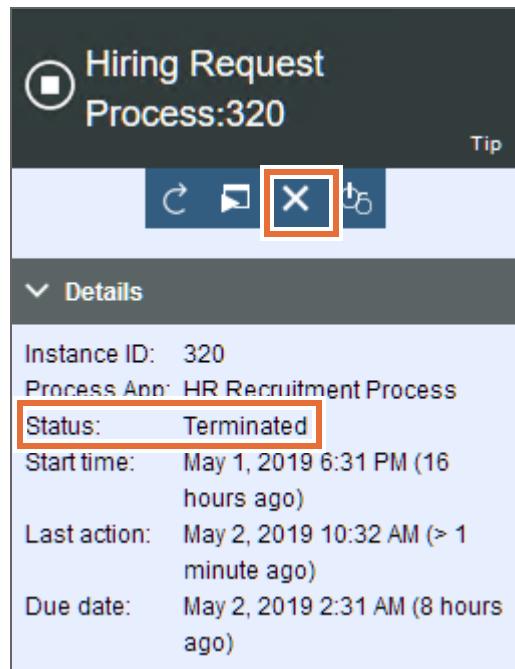


\_\_ b. Click **Yes** when presented with the confirmation dialog.



\_\_ c. The process is now in a **Terminated** status.

\_\_ d. Click the **Delete** icon to delete the process instance.



\_\_ e. Click **Yes** when presented with the confirmation dialog.

It is important to clean up the environment to maintain performance service levels. Leaving running processes in the environment that are unnecessary can degrade performance and cause unnecessary clutter.

- \_\_ f. Click **Refresh** to verify that the process instance is no longer listed.

The screenshot shows a search interface for process instances. On the left, there is a sidebar with a 'Search' bar at the top. Below it, there are sections for filtering by 'Status' (Active, Completed, Failed, Suspended, Terminated) and 'Severity type'. Under 'Person', there is a text input field for 'Name or user name'. Below these, there are date range filters for 'Last modified date' with 'From' and 'To' fields. At the bottom of the sidebar is a blue 'Refresh' button, which is highlighted with a red rectangular border. The main area of the interface displays a message: 'Showing 0 of 0 instances' and 'Select shown instances | Select all instances | Clear selection'. A dropdown menu 'Sort by:' is set to 'Date of last action'. The central message area says 'No process instances fulfill your search criteria.' and 'Modify your search filters and then click Refresh.'



## Information

If you have extra process instances as a result of testing, you can use the Process Inspector to terminate and delete these instances. You can highlight instances that are in the same state and perform batch operations that appear to the right.

| Showing 8 of 8 instances            |                                                                                                  |
|-------------------------------------|--------------------------------------------------------------------------------------------------|
| Sort by: Date of last action        |                                                                                                  |
| <input checked="" type="checkbox"/> | Hiring Request Process:328 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:327 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:326 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:325 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:324 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:322 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:323 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |
| <input checked="" type="checkbox"/> | Hiring Request Process:321 * Hiring Request Process<br>Last modified May 2, 2019 Due May 2, 2019 |

7 processes selected

Status:  
• Terminated (7)

Actions

Delete

## Part 2: Configure the Document Viewer to use the Target Object Store (TOS)

Up to this point, you configured an out-of-the box use of the local BPM document store. However, Business Automation Workflow also has a Target Object Store. The Target Object Store can be used to share documents between Case solutions and Process applications.

Management would like to see how content can be shared between a Case solution and a BPM solution. Specifically, management would like to see how a Case solution can trigger the new HR structured process solution and share documents between the two solutions by using one content repository.

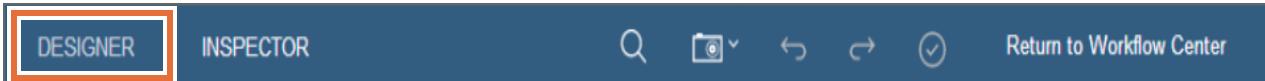
As part of this exercise, you create a simple, bare-bones Case solution that starts the Hiring Request Process from the HR Recruitment Process BPM solution. To do this, you add the Hiring Request Process as an activity to the Case solution.

In this section, you also define an ECM server that points to the Target Object Store (TOS). Then, you reuse what you built in the Hiring Request From Document coach and configure the view to display the TOS document store instead of the BPM document store. Then, you test the integration between the Case solution and the BPM solution by adding a new case and document to the Case.

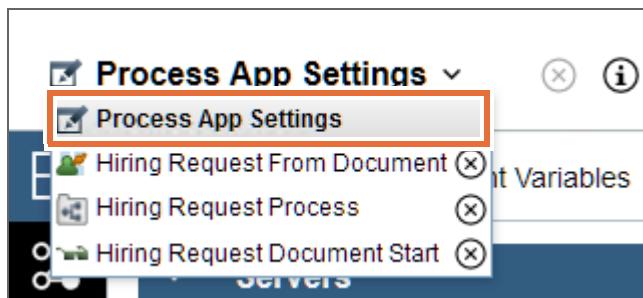
After the document is added, you verify that the Hiring Request Process was started and that the newly configured view displays the new document in the document list.

- \_\_\_ 1. Define the new ECM server to use the Target Object Store.

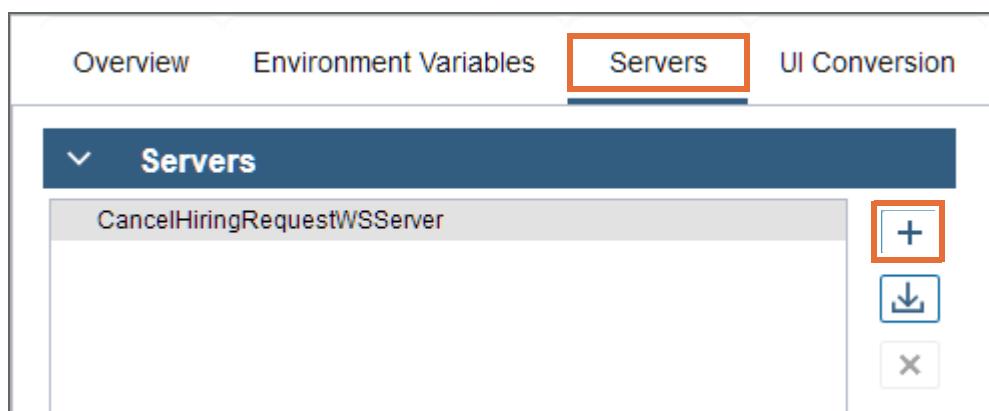
- \_\_\_ a. Return to the **Designer** tab.



- \_\_\_ b. Open the **Process App Settings** by selecting it from the drop-down list.

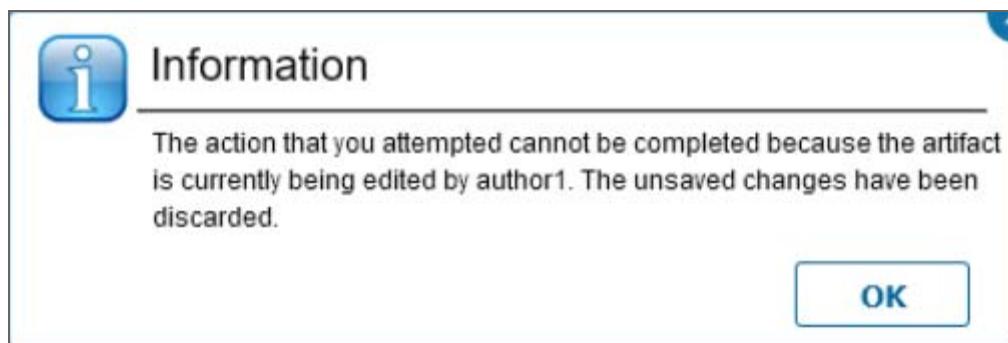


- \_\_\_ c. Select the **Servers** tab.
  - \_\_\_ d. Click **Add a new server**.



### Troubleshooting

If you see the following Information window, click **OK** and close the browser and log in again to Workflow Center.



\_\_\_ e. Configure the new server with the following information:

| Field                                  | Value                                |
|----------------------------------------|--------------------------------------|
| Name                                   | ECM_TOS                              |
| Type                                   | Enterprise Content Management Server |
| Description                            | ECM Target Object Store              |
| Hostname                               | ws2016x64                            |
| Port                                   | 9443                                 |
| Context path                           | /fncmis                              |
| Secure server                          | <b>checked</b>                       |
| Repository                             | tos                                  |
| User ID                                | author1                              |
| Password                               | passw0rd                             |
| Always use this connection information | <b>checked</b>                       |

\_\_\_ f. Save your work.

- \_\_ 2. Create the ECM document authorization service.
  - \_\_ a. Click **New** next to ECM document authorization service.

**Details**

|             |                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Name        | ECM_TOS                                                                                                                                   |
| Type        | Enterprise Content Management Server                                                                                                      |
| Description | <p><b>B I U</b>   ≡ ≡ ≡ ≡   <math>\frac{1}{2}</math> ≡   <math>\diamond</math> ≡ <math>\diamond</math></p> <p>ECM Target Object Store</p> |

**Default** (+)

|                                         |                                                                  |
|-----------------------------------------|------------------------------------------------------------------|
| Host name:                              | ws2016x64                                                        |
| Port:                                   | 9443                                                             |
| Context path:                           | /fnncmis                                                         |
| Secure server:                          | <input checked="" type="checkbox"/>                              |
| Repository:                             | tos                                                              |
| User ID:                                | author1                                                          |
| Password:                               | *****                                                            |
| Always use this connection information: | <input checked="" type="checkbox"/>                              |
| ECM document authorization service:     | <none> <span>Select...</span> <span>New...</span> <span>X</span> |
| Event broadcasters:                     | <none> <span>Select...</span> <span>New...</span> <span>X</span> |

Test connection

- \_\_\_ b. Name the new Service Flow ECM DocAuth Service and click **FINISH**.

New Service Flow

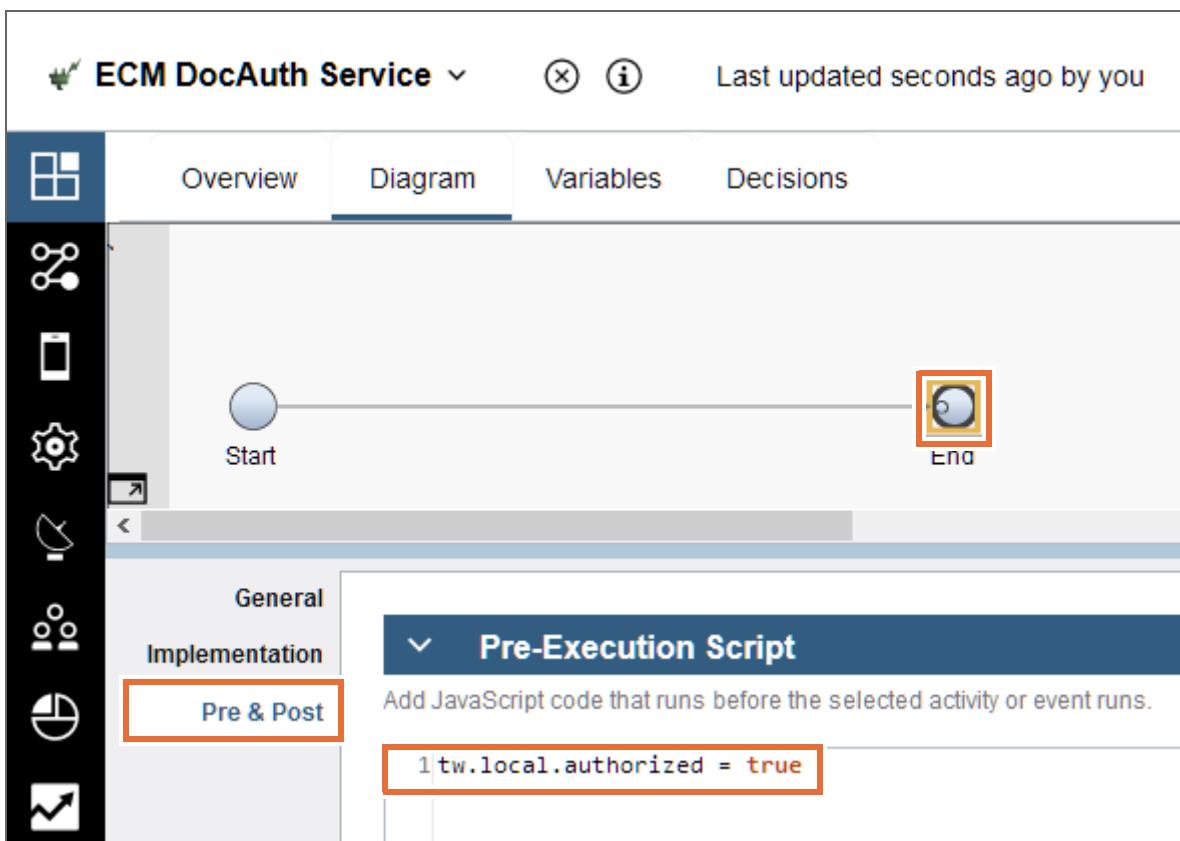
A service flow lets you build, test, and run a set of services. [Learn More](#)

Name:

Use as a team service:

**FINISH** **CANCEL**

- \_\_\_ c. Select the **End** event.  
 \_\_\_ d. Select the **Pre & Post** properties menu below.  
 \_\_\_ e. Enter the following for the Pre-Execution Script: `tw.local.authorized = true`



- \_\_\_ f. Save your work.

- \_\_\_ g. Close the **ECM DocAuth Service** window.

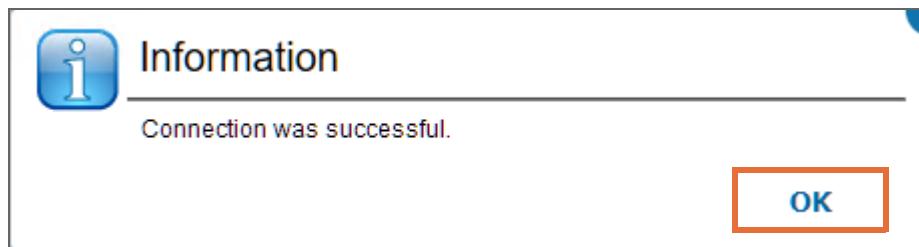


- \_\_\_ h. Return to the **ECM\_TOS** configuration.
- \_\_\_ 3. Configure the new ECM server to broadcast events to all users and test the connection.
- \_\_\_ a. Click **Select** next to Event broadcasters and select **All Users**.
- \_\_\_ b. Verify your configuration and save your work.

| Details                                 |                                                                                                                          |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Name                                    | ECM_TOS                                                                                                                  |
| Type                                    | Enterprise Content Management Server                                                                                     |
| Description                             | <b>B I U</b>   <b>≡ ≡ ≡ ≡</b>   <b>≡ ≡</b>   <b>≡ ≡</b><br>ECM Design Object Store                                       |
| <b>Default</b> <a href="#">+</a>        |                                                                                                                          |
| Host name:                              | ws2016x64                                                                                                                |
| Port:                                   | 9443                                                                                                                     |
| Context path:                           | /fnecmis                                                                                                                 |
| Secure server:                          | <input checked="" type="checkbox"/>                                                                                      |
| Repository:                             | tos                                                                                                                      |
| User ID:                                | author1                                                                                                                  |
| Password:                               | *****                                                                                                                    |
| Always use this connection information: | <input checked="" type="checkbox"/>                                                                                      |
| ECM document authorization service:     | <a href="#">Select...</a> <a href="#">New...</a> <a href="#">X</a>                                                       |
| Event broadcasters:                     | <a href="#">All Users</a> <a href="#">System Data</a> <a href="#">Select...</a> <a href="#">New...</a> <a href="#">X</a> |
| <a href="#">Test connection</a>         |                                                                                                                          |

- \_\_\_ c. Click **Test connection** to verify connection to the object store.

- \_\_\_ d. Click **OK** on the Information dialog.



Now that the server is configured, you can return to the Startable Service coach to configure it to use the new ECM\_TOS server.

- \_\_\_ 4. Create the ECM list View

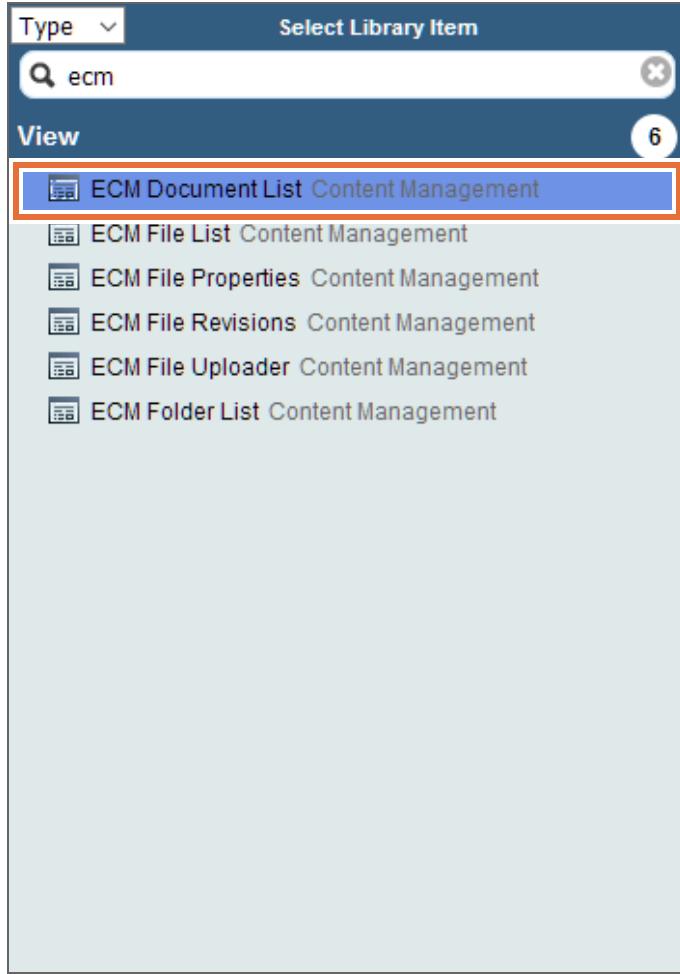
- \_\_\_ a. Return to the **Hiring Request From Document** client-side human service.  
\_\_\_ b. Click the Coach downward arrow and select Coach.



- \_\_\_ c. Select the **BPM Document List**.  
\_\_\_ d. Under the **Behavior** section in the **General** properties tab, click **Select** next to View.

|                 |                                                                                                                                                                                                                                                                                              |  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| General         | <b>Behavior</b><br>Binding: <input checked="" type="radio"/> document ( <a href="#">ECMDocumentInfo</a> )<br>View: <a href="#">BPM Document List (ECMDocumentInfo) Content Management</a> <b>Select...</b> <a href="#">New...</a><br>Label visibility: <input checked="" type="radio"/> Show |  |
| Positioning     |                                                                                                                                                                                                                                                                                              |  |
| Configuration   |                                                                                                                                                                                                                                                                                              |  |
| Events          |                                                                                                                                                                                                                                                                                              |  |
| Visibility      |                                                                                                                                                                                                                                                                                              |  |
| HTML attributes |                                                                                                                                                                                                                                                                                              |  |

- \_\_ e. Enter `ecm` to filter the selection. Select the **ECM Document List** view.



Because the ECM Document List uses the same content object (ECMDocumentInfo), you can leave the binding as is.

- \_\_ f. Rename the view to **ECM Document List**.
- \_\_ g. Click the Configuration properties tab and expand **Content management advanced**.
- \_\_ h. Enter `ECM_TOS` for the ECM server configuration name.

**Content management advanced**

|                                |                                      |
|--------------------------------|--------------------------------------|
| ECM server configuration name: | <input type="text" value="ECM_TOS"/> |
| Document object type ID:       | <input type="text"/>                 |
| Folder path:                   | <input type="text"/>                 |

- \_\_ i. Save your work.

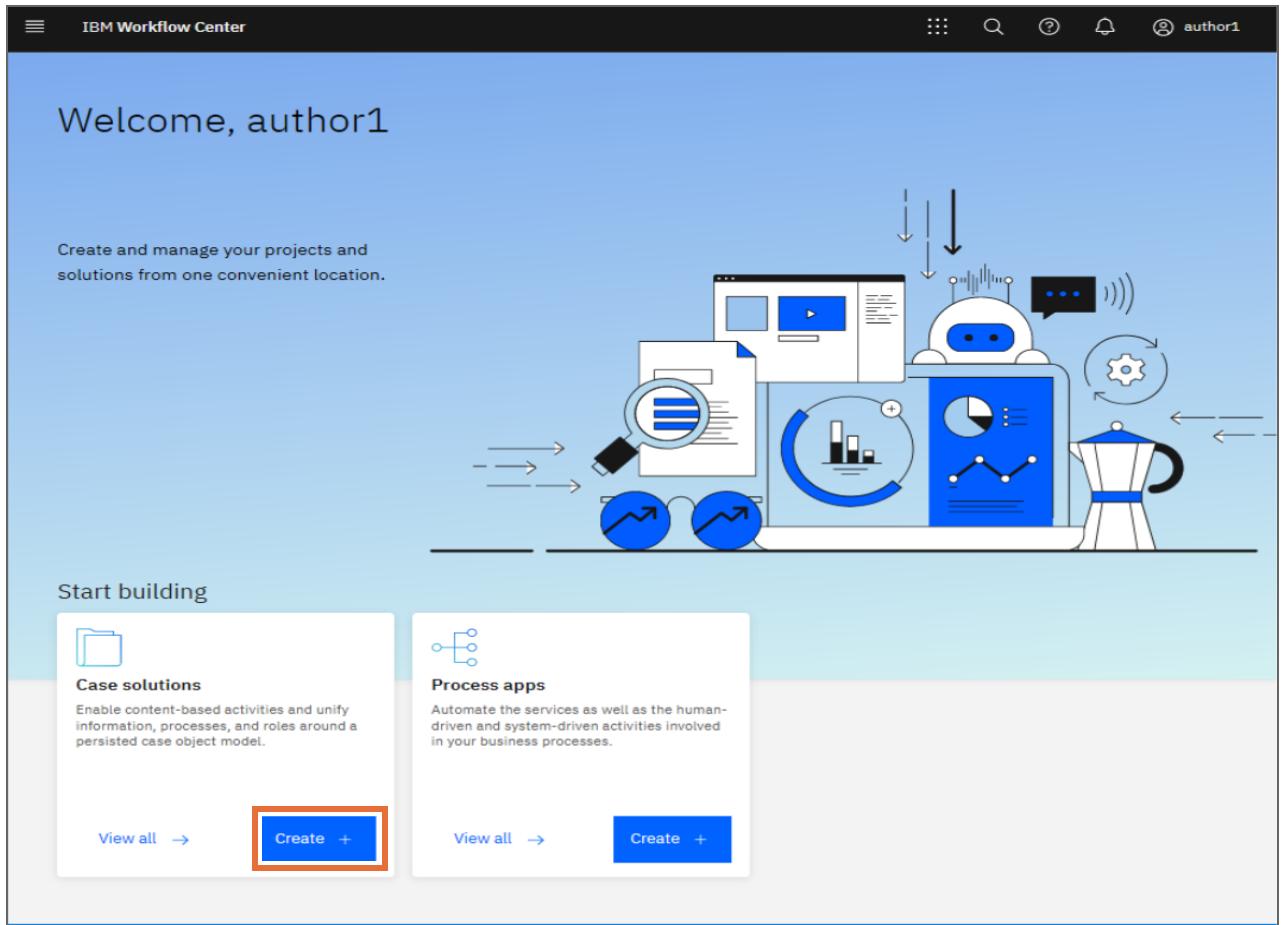
### Part 3: Build Case solution

Although this course does not cover Case Management development, it is important to understand how the integration works between a Case solution and a BPM solution.

In this section, you build a simple, bare-bones Case solution by using the Workflow Center and Case Builder. You only create the necessary components to run the solution and test integration between the Case and the BPM solution.

— 1. Create a new case solution

- a. Open a new tab in the browser and access the Workflow Center.  
If prompted, log in to the Workflow Center as author1 / passw0rd
- b. Click **Create** on the Case solutions tile to create a new Case solution.



- c. Name the solution: HR Case Solution
- d. Leave the acronym as the default.
- e. Make sure that the checkbox is selected for **Open in Designer**
- f. Choose **Traditional** for Target environment

\_\_ g. Click **Create**.

Create a case solution

Name ⓘ

HR Case Solution

Acronym ⓘ

HRCS

Open in Designer

Select a base template and snapshot (optional)  
No case templates available

Select a base template

Target environment

Traditional

Description

Highlight your text to see formatting options.

Description

Cancel Create

The solution is opened in Workflow Center.



### Hint

If you do not see all the options, make sure you are using the **Advanced** View Mode in the user preferences of the Workflow Center.

\_\_ 2. Add Role

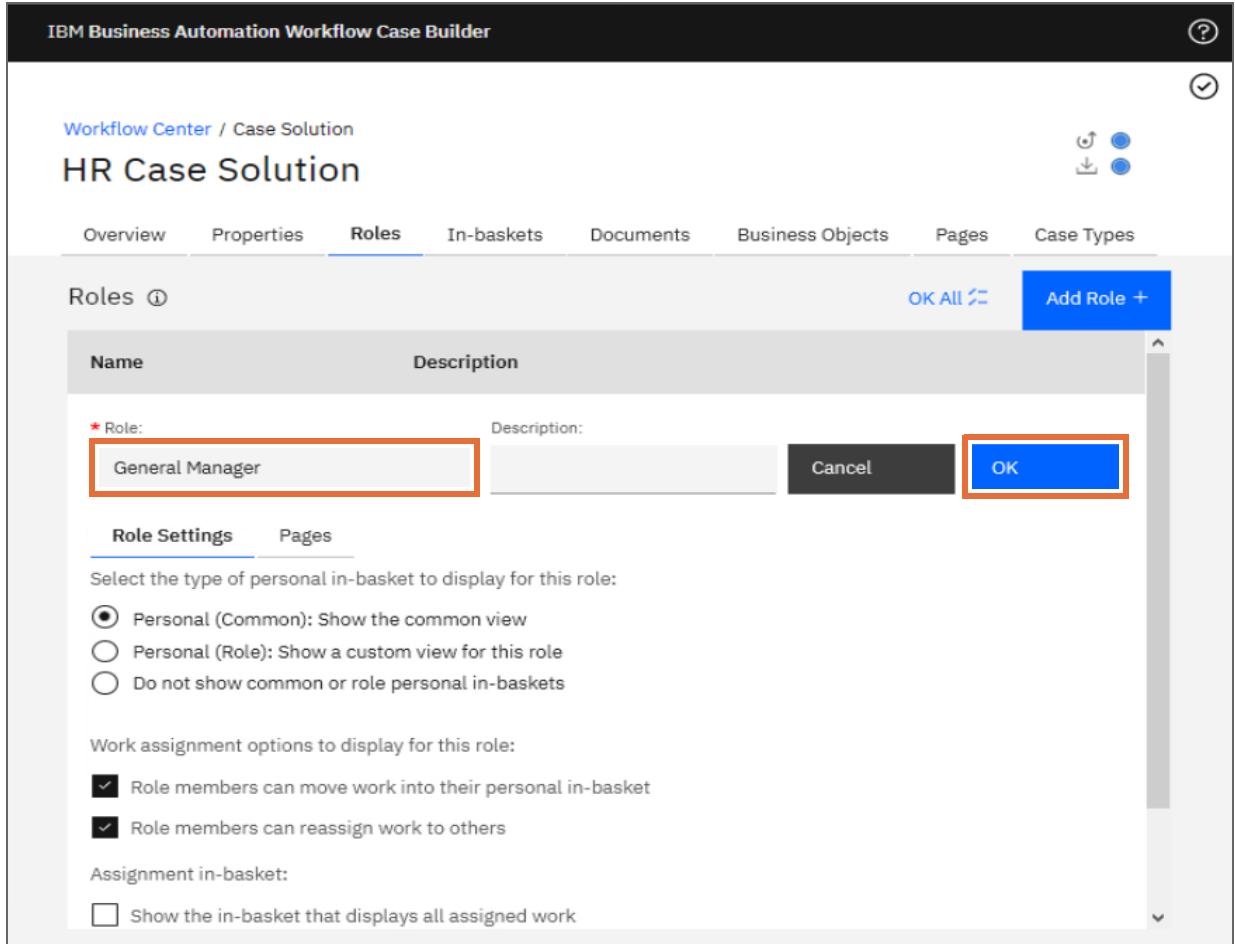
- \_\_ a. Click the **Roles** tab.

The screenshot shows the 'IBM Business Automation Workflow Case Builder' interface. At the top, there's a navigation bar with 'Workflow Center / Case Solution' and a 'Case Solution' title 'HR Case Solution'. Below the title, a toolbar has tabs: 'Overview' (underlined), 'Properties', 'Roles' (which is highlighted with a red box), 'In-baskets', 'Documents', 'Business Objects', 'Pages', and 'Case Types'. Under the 'Roles' tab, there's a large white box containing a folder icon, solution prefix 'HRCS', source control information, and creation details ('Created by author1' and 'Created on September 16, 2020'). Below this box is a 'Description' section with a text input field. In the top right corner of the interface, there are download and refresh icons.

- \_\_ b. Click **Add Role**.

- \_\_ c. Enter General Manager for the Role.

- \_\_ d. Click **OK**.



- \_\_ 3. Add Document Class
- \_\_ a. Click the **Documents** tab.
  - \_\_ b. Click **Add Document Class** and select **New**.
  - \_\_ c. Enter **HR** for Name.
  - \_\_ d. Add **HRCLASS** to the Unique Identifier

- \_\_ e. Click **OK**.

The screenshot shows a configuration dialog box titled "General". It is used for adding a new document class. The "Name" field is filled with "HR". The "Unique Identifier" field contains "HRCS\_HRCLASS". There is a large empty text area for "Description:". At the bottom, there are two buttons: "Cancel" and "OK". The "OK" button is highlighted with a blue border.

The document class is displayed to the left.

- \_\_ 4. Add Case Type
- \_\_ a. Click the **Case Types** tab
  - \_\_ b. Click **Add Case Type**.  
The Case is opened in a new editor
  - \_\_ c. Enter **HR** for the Case type name.

\_\_ d. Click **Activities**.

The screenshot shows the 'Case Type' configuration page for the 'HR' case type. The top navigation bar includes links to 'Workflow Center', 'HR Case Solution', and 'Case Type'. The main header 'HR' is displayed above the configuration tabs. The tabs are: 'Case Type' (selected), 'Properties', 'Views', 'Case Folders', 'Stages', 'Rules', and 'Activities' (highlighted with a red box). Below the tabs, the 'Case Type' section is titled 'Case Type' with a help icon. It contains the following fields:

- Case type name: HR
- Case type unique identifier: HRCS\_ HR
- Case type description: (empty text area)
- Starting document class: <None>
- External repository:
  - Allow documents and attachments from repositories other than the case management object stores
  - Display system-generated titles initially instead of the original document titles

\_\_\_ 5. Add Activity

- \_\_\_ a. Click **Add Activity** and select **Activity with Existing Process**

The screenshot shows the 'Workflow Center / HR Case Solution / Case Type' interface. The 'Activities' tab is selected. A red box highlights the 'Add Activity' button. A secondary red box highlights the 'Activity with Existing Process' option in the dropdown menu.

- \_\_\_ b. Enter Hiring Request Process as the **Name**.

- \_\_\_ c. Enter the following for description:

This activity starts the Hiring Request Process of the HR Recruitment Process BPM solution.

\_\_ d. Click **Next**.

Add Activity

General

\* Name:

\* Unique Identifier:

Description:  
This activity starts the Hiring Request Process of the HR Recruitment Process BPM solution.

This activity starts:  
 Automatically    Manually    Discretionally

This activity is:  
 Hidden

Back Cancel Next

- \_\_ e. Click the Preconditions drop down menu and select **A document is filed in the case**.
- \_\_ f. Click the **Any document class** checkbox to clear it.
- \_\_ g. Verify that the **HR Document Class** is selected.

\_\_ h. Click **Next**.

Add Activity

Preconditions

What preconditions must be met for this activity to start?

A document is filed in the case  Activity is repeatable

Any document class

Document Classes:

HR

The above precondition and the following conditions:

Match: All

| Property | Operator | Value |
|----------|----------|-------|
|----------|----------|-------|

Back Cancel

- \_\_ i. Select **HR Recruitment Process** for the Workflow Project name  
\_\_ j. Leave **Default Version** for the Snapshot name

- \_\_ k. Select **Hiring Request Process** under the list of processes populated below.

Add Activity

HR Recruitment Process

Snapshot name:

Default Version

Select a process:

Filter processes      Refresh      Open Web Process Designer

| Process Name           | Description |
|------------------------|-------------|
| Approve Hire Request   |             |
| Hiring Request Process |             |

Back      Cancel      Next

- \_\_ l. Click **Next**.

- \_\_ m. Because there are no properties in the Case, there is no need to map any properties in the next screen. Click **Finish**.

Add Activity

Selected process: Hiring Request Process

Map Properties

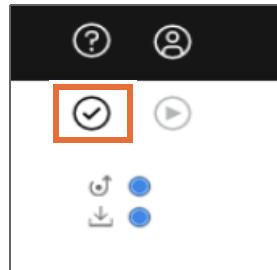
Map process data fields to solution properties.  
Click the "+" button to save the data field to property mapping.

Process data field name: <None>      Case type property name: <None>

Property map:

Back      Finish      Next

- \_\_\_ n. The activity is added to the Case as an optional activity.
- \_\_\_ o. Click **Save** in the upper right corner.

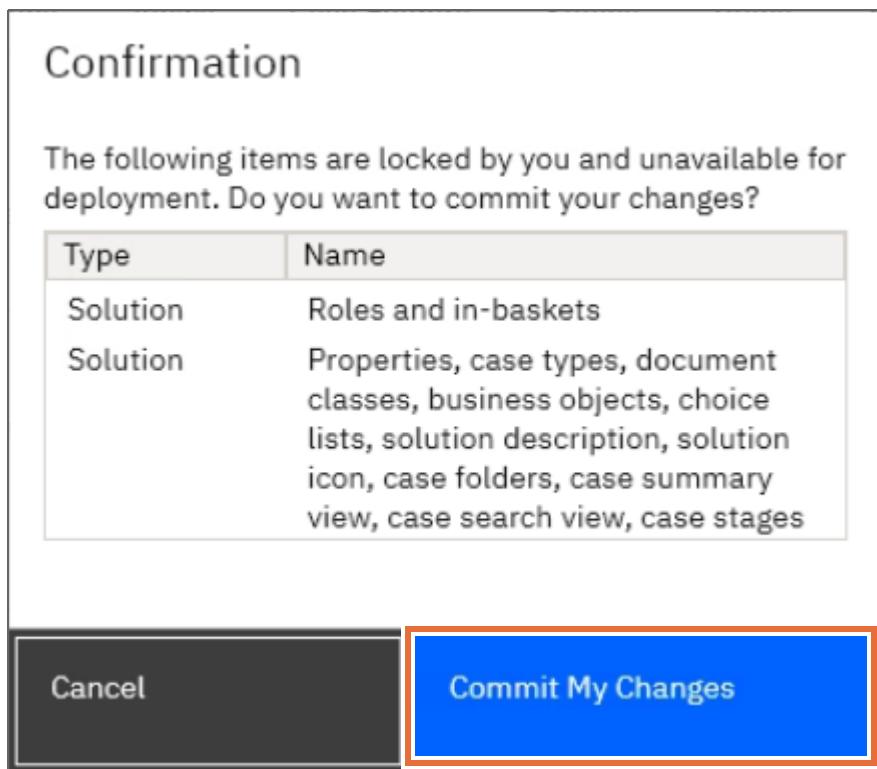


- \_\_\_ 6. Deploy the Case solution.

- \_\_\_ a. Click **Commit** in the upper right corner.



- \_\_\_ b. Click **Commit My Changes** in the Confirmation dialog box. The commit icon turns to a green check mark.



- \_\_ c. Click **Deploy** in the upper right corner.



After successful deployment, the deployment icon changes to a green check mark.



## Troubleshooting

If the deployment fails, you can select the Overview tab, then click the three dots to the right and select **Deployment errors** to view the errors in the deployment.

The deployment errors are listed below.

The deployment errors were retrieved.

```

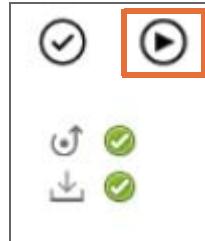
11/30/20 2:35:39 PM EST FNRPA0580E The document class HRCS_HRCLASS was found in Content Platform Engine with ID {C1770D5B-31B4-4498-AA3C-DA12B290D675} but is defined in solution with ID {B47A4BF3-2069-438D-B2AE-1E23944A8D6E}.
11/30/20 2:35:39 PM EST FNRPA0138E The solution cannot be deployed because the reused HRCS_HR case type ID does not match the ID that was specified in the solution.
11/30/20 2:35:41 PM EST FNRPA0525E The required process application and snapshot information is missing from the solution.

```

## Part 4: Test your solution

In this section, you test the solution, demonstrating the document list view now displays content from the Target Object Store (TOS) and that the Hiring Request process starts when a document is added to the HR Case solution.

- 1. Create a new case.
  - a. Click **Start** to run the Case solution. The solution opens in a new window.

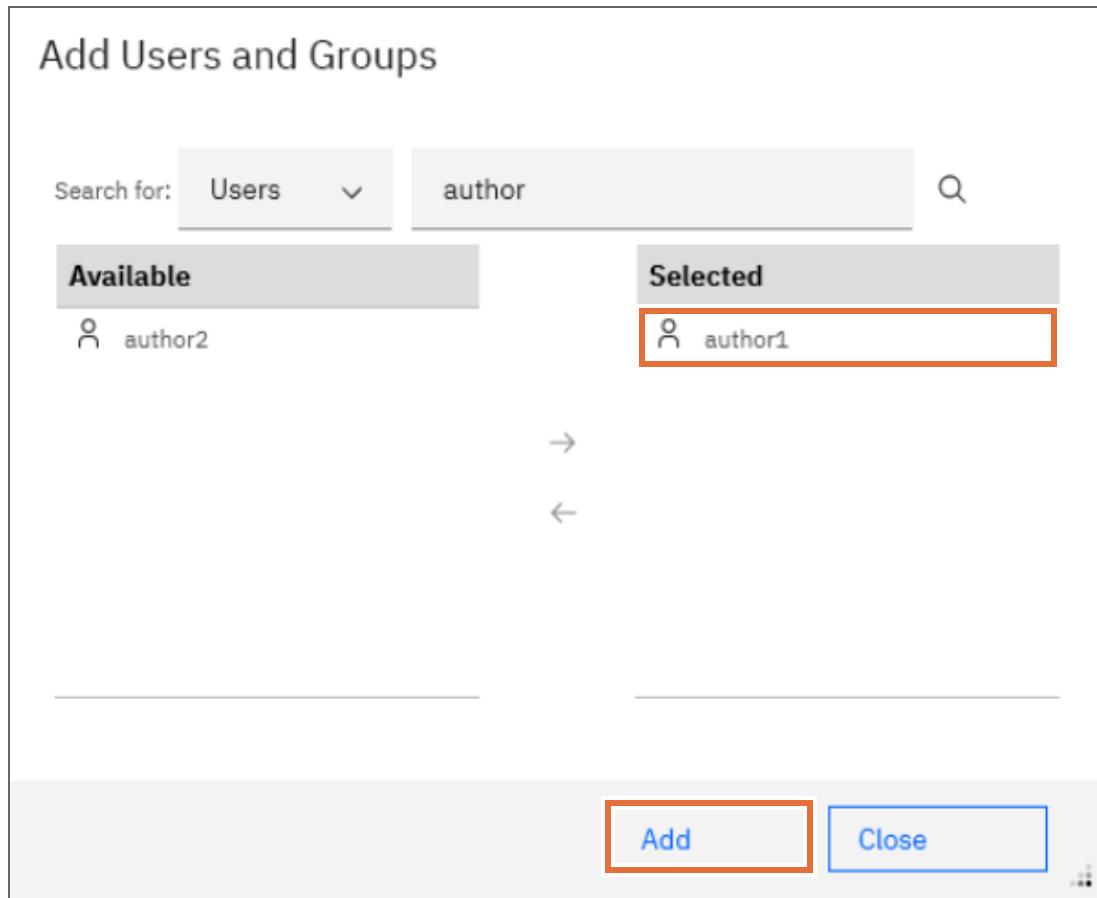


- b. Before adding a case to the solution, you need to add yourself as a user to the General Manager role. Click **HR Case Solution** in the upper right, expand HR Case Solution, and click **Manage Roles**.

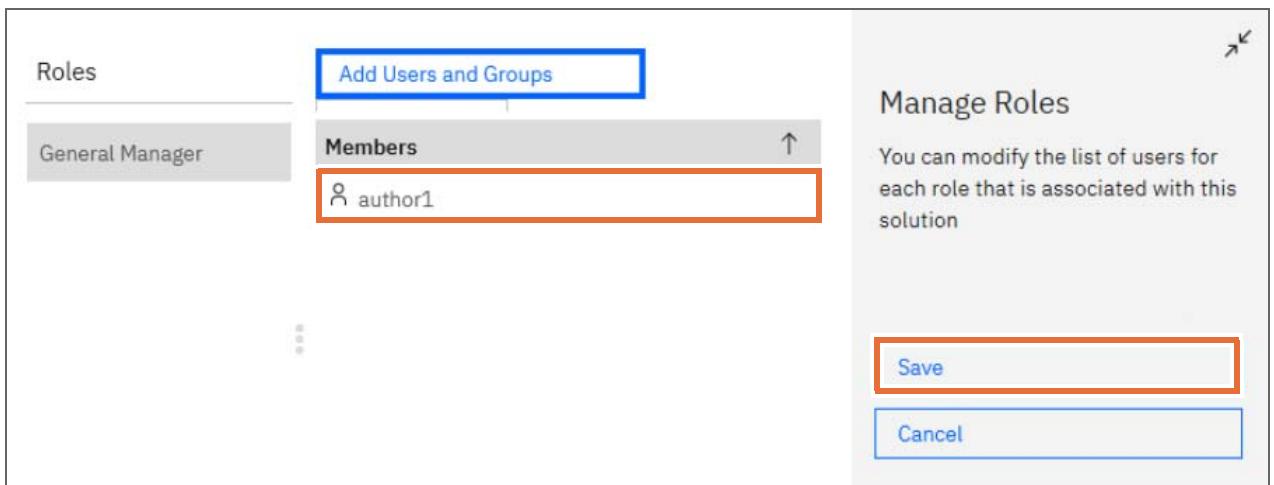


- c. Click **Add Users and Groups**
- d. Enter `author` for the search criteria and click the **Search** icon.
- e. Select **author1** and click **Add** (right arrow).

\_\_ f. Once author1 is in the Selected column, click **Add**.



\_\_ g. Verify author1 is added as a user to the General Manager role and click **Save**.



\_\_ h. Click **Add Case**, then select **HR**.

The screenshot shows the 'IBM Business Automation Workflow Case Client' interface. At the top, there are navigation tabs for 'Cases' and 'Work'. On the right, it displays 'HR Case Solution | General Manager'. Below the tabs, there is a search bar with a dropdown menu set to 'Added On' and a date selector showing '9/17/2020'. A large red box highlights the 'Add Case' button, which is currently active. To the right of the search area, a message says 'No items to display'. At the bottom, a status bar indicates '9/17/2020, 4:14 PM - 1 process roles were found.'

\_\_ i. Click **Add**.

The screenshot shows the 'Add Case' dialog box for the 'HR' role. The title bar says 'IBM Business Automation Workflow Case Client' and the tab bar shows 'Cases', 'Work', and 'Add Case' (which is highlighted). The main area is labeled 'HR' and contains the message 'No properties to display'. At the bottom right, there are two buttons: 'Add' (highlighted with a red box) and 'Cancel'. A status bar at the bottom indicates '9/17/2020, 4:14 PM - 1 process roles were found.'

- \_\_ j. A message that states the new case was successfully created is displayed in the bottom status bar. Click **Search** to list the new case.

The screenshot shows the 'IBM Business Automation Workflow Case Client' interface. At the top, there are tabs for 'Cases' (selected) and 'Work'. On the right, it says 'HR Case Solution | General Manager'. Below the tabs, there's a button 'Add Case' and some icons. A search bar is present with the placeholder 'Search:' and a dropdown menu set to 'Added On'. Underneath, a date '9/17/2020' is selected. A red box highlights the 'Search' button. Below the search area, a message says 'No items to display'. At the bottom of the screen, a status bar shows the message '9/17/2020, 4:42 PM - The new case HRCS\_HR\_000000100001 of type HR was successfully created.'

- \_\_ 2. Add a document to the case.

- \_\_ a. Click the title link for the newly added case. The title link for your case might be different.

| Title                                | Added On           | ↑ Case State | Modified By | Modified On        |
|--------------------------------------|--------------------|--------------|-------------|--------------------|
| <a href="#">HRCS_HR_000000100001</a> | 9/17/2020, 4:42 PM | Working      | author1     | 9/17/2020, 4:42 PM |

- \_\_ b. Click **Add** and select **Add document from Local System**.

The screenshot shows the details of a case titled 'Case HRCS\_HR\_000000100001'. The top navigation includes 'Cases', 'Work', and a link to the case. The case title is highlighted with a red box. Below the title, the status is 'Modified: 9/17/2020, 4:42 PM | HR'. There are tabs for 'Comments', 'Split Case', 'Save', and 'Close'. Under the 'Documents' tab, which is selected, there's a message 'No properties to display'. Below this, there are buttons for 'Add' and 'Actions'. A red box highlights the 'Add' button. A dropdown menu is open, showing options: 'Add Folder', 'Add Document from Local System' (which is also highlighted with a red box), and 'Add Document from Repository'. At the bottom, a status bar shows '9/17/2020, 4:46 PM - The 000000100001 folder returned 0 items.'

- \_\_\_ c. In the Add Document window, click the **Browse** button, and navigate to:

C:\labfiles\Support files\Ex14\AAA Employment application.pdf

**General**

What do you want to save? Local document

\* File name: **Browse...** No files selected.

Major version ⓘ

**Properties**

\* Class: Document

Document Title: ⓘ

- \_\_\_ d. Select **HR** for the document class by clicking the **Class** drop down list, selecting **HR**, then clicking **OK**.
- \_\_\_ e. Verify that the correct class and document are selected. Then, click **Add** to add the document to the case.

**General**

What do you want to save? Local document

\* File name: **Browse...** AAA Employment application.pdf

Major version ⓘ

**Properties**

\* Class: HR

Document Title: ⓘ AAA Employment application.pdf

**Add Document**

The values that you enter for the document properties can be used to find the document later.

**Add**

**Cancel**

- \_\_\_ f. The document is listed as part of the case.
- \_\_\_ 3. Validate the status of the started activity.
- \_\_\_ a. Click the **Activities** tab.

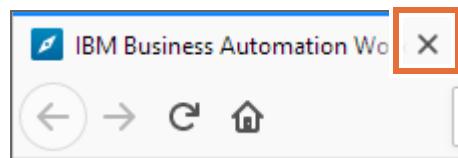
The screenshot shows the IBM Business Automation Worklist interface. At the top, there are tabs for 'Cases' and 'Work', with 'Case HRCS\_HR\_000000100001' selected. On the right, it says 'HR Case Solution | General Manager'. Below the tabs, the case ID 'HRCS\_HR\_000000100001' is displayed along with the modification date 'Modified: 9/17/2020, 4:42 PM | HR'. There are 'Comments' and 'Split Case' buttons. Under the 'Activities' tab, it says 'No properties to display'. There are 'Add' and 'Actions' buttons, and icons for sorting and filtering. The 'Documents' section lists a file named 'AAA Employment application.pdf' with the modification details 'Modified By: author1 | Modified On: 9/17/2020, 4:56 PM'. A message at the bottom states '9/17/2020, 4:56 PM - The 000000100001 folder returned 1 items.'

- \_\_\_ b. The case reports that the Hiring Request process is started.

The screenshot shows a process history window with a title 'Required (1)'. It contains one item: 'Hiring Request Process' with the note 'Started on 5/15/2019, 12:57 PM'.

You have added a case to the existing HR Case solution and added a document to the new case. This event started the Hiring Request Process.

- \_\_\_ c. Close the case window by clicking the X in the tab.



- \_\_\_ 4. Verify that the Hiring Request Process is started in Process Portal and that the document is listed in the ECM Document List view.
- \_\_\_ a. Open Process Portal.  
If asked for credentials, enter the following: **Username:** author1 **Password:** passw0rd
- \_\_\_ b. Verify the Hiring Request Process started and that the **Step: Submit Hiring Request** is in your work queue.

- \_\_\_ c. Click the **Hiring Request From Document** startable service.

The page might take a minute to load. The AAA Employment application.pdf document appears in the list. Your document list might look different.

| Name                                   | Last Modified | Version |  |
|----------------------------------------|---------------|---------|--|
| AAA Employment application.pdf         | 5/2/2019      | 1.0     |  |
| CmAcmCaseActivitySweep                 | 4/27/2019     | 1.0     |  |
| CmAcmCaseHealthAnalysis                | 4/27/2019     | 1.0     |  |
| CmAcmCaseOperations                    | 4/27/2019     | 1.0     |  |
| CmAcmCaseOperationsComponentDefinition | 4/27/2019     | 3.0     |  |
| CmAcmEventHandler                      | 4/27/2019     | 1.0     |  |
| CmAcmRuleDeploymentAndOperations       | 4/27/2019     | 1.0     |  |
| CmAcmRuleOperationsComponentDefinition | 4/27/2019     | 3.0     |  |
| Detail Deployment Log                  | 4/30/2019     | 5.0     |  |
| Detail Deployment Log                  | 4/30/2019     | 4.0     |  |

**OK**



## Troubleshooting

If the AAA Employment application.pdf document does not display in the list, you can click **Load More...** to display more documents. You can then sort the documents by name.

- \_\_\_ d. Click **OK** to close the service.

You have completed this exercise.

## Part 5: Export the process application

Now that this course is complete, you are ready to export the process application. This export saves your application as a `.twx` file in a folder for future use. If you wish to keep your solution, you can export it following this procedure and upload it to a web-based file sharing program such as Box.

- 1. Create a snapshot
  - a. Return to the Workflow Center and click the **Process apps** tile.
  - b. Click **Details** on the HR Recruitment Process tile.

The screenshot shows the 'Process apps' section of the IBM Workflow Center. There are three project cards displayed:

- HR Recruitment Process**: Contains HR processes used for recruitment. The 'Details →' button is highlighted with a red box.
- Performance**: Contains performance-related processes.
- Process Portal**: Contains process portal-related processes.

At the top, there are search and filter buttons ('Recently updated', 'All'), and buttons for 'Import' and 'Create'. A message at the top says 'Displaying 8/8 projects'.

- c. Select the **Snapshots** tab.
- d. Open the list of Options by clicking the three dots on right side of the **Current** tip.  
The tip is the current working version of a project.

\_\_ e. Select **Create a snapshot**

Back to process apps /

## HR Recruitment Process

Open in Designer

Overview    **Snapshots**    Permissions    History    Comments

Snapshots (0) ①

Name    Status    Created    Last updated    Target environment

| Name      | Status  | Created                | Last updated | Target environment |
|-----------|---------|------------------------|--------------|--------------------|
| → Current | author1 | 9/22/2020, 12:51:50 PM | author1      | Traditional        |

⋮

Create a snapshot

Open in Designer

\_\_ f. Enter WB835 Solution for the name.

### Create a snapshot

Snapshot name

WB835 Solution

Description

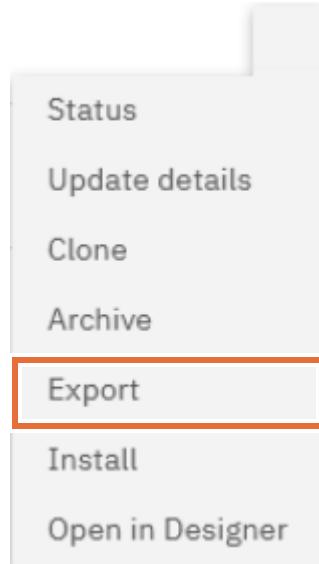
Highlight your text to see formatting options.

Description

Cancel    Create

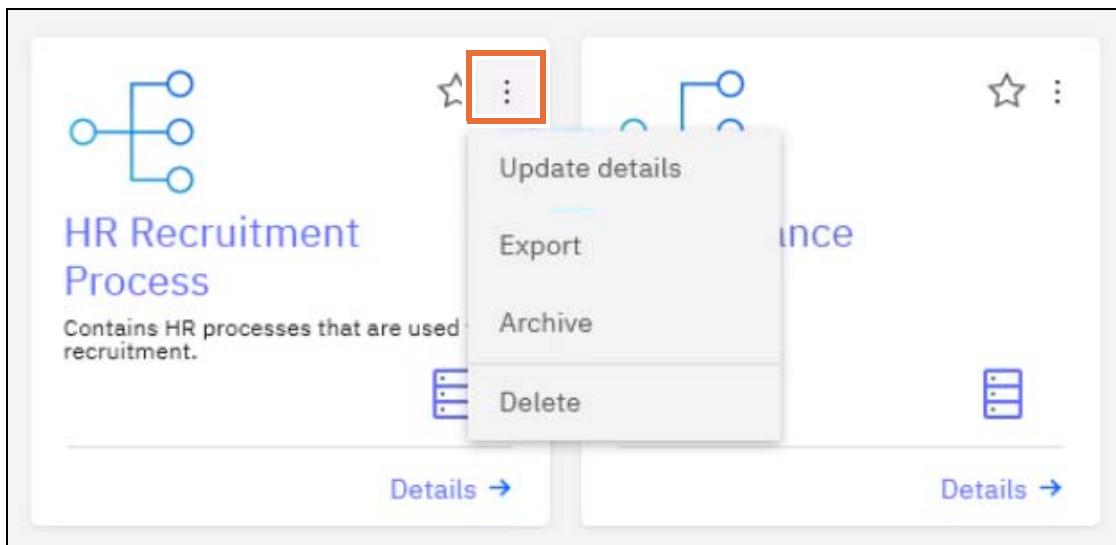
\_\_ g. Click **Create**.

- 2. Export the application. Exporting the application also exports the toolkit that it depends on.  
You can only export snapshots.
- a. Open the list of Options for the snapshot and select **Export**

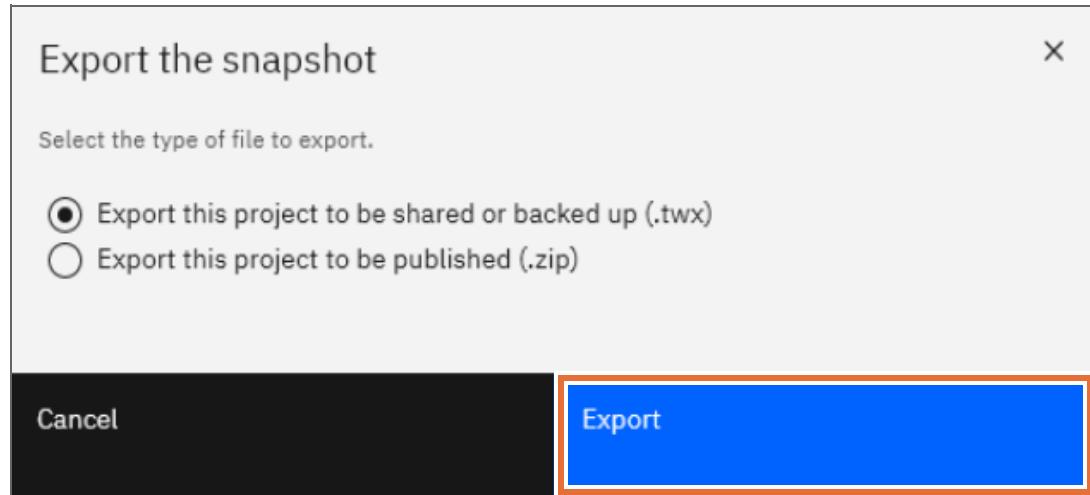


### Hint

You can also export, archive, and delete process applications directly from the **Process apps** page by accessing the list of options on the tile.



- \_\_\_ b. In the **Export the snapshot** window, click **Export**. It may take a minute or two to generate the **.twx** file.



- \_\_\_ c. Click **OK** to in the dialog box to save the file. The application is saved to the default folder, which is the Downloads directory.  
\_\_\_ d. Verify that the application is listed in the downloads directory.

This concludes the solution export.

In the first part of the exercise, you modified an existing process to set up a document start event. You created the event subscription and verified the user-generated settings. You then created a coach and tested the event subscription. You proved that the document was sent as part of the start event output variable.

You then created an ECM server to point to the Target Object Store and reconfigured the BPM Document List view to show the new object store. As part of building a simple Case solution, you attached the Hiring Request Process from the BPM solution as an activity in the Case. Then, you tested the document viewer and integration by running the Case and adding a document to the Case. To verify the integration, you viewed the Submit Hiring Request task in the Work page. To verify the document viewer, you displayed the new document in the Document List view.

You have completed all exercises in this course.

## End of exercise

# Appendix A. Solutions

In this section, instructions for importing solution files is provided. The following solution files are located at c:\solutions.

| Solution file                                     |
|---------------------------------------------------|
| HR_Recruitment_Process - Exercise_1_solution.twx  |
| HR_Recruitment_Process - Exercise_2_solution.twx  |
| HR_Recruitment_Process - Exercise_3_solution.twx  |
| HR_Recruitment_Process - Exercise_4_solution.twx  |
| HR_Recruitment_Process - Exercise_5_solution.twx  |
| HR_Recruitment_Process - Exercise_6_solution.twx  |
| HR_Recruitment_Process - Exercise_7_solution.twx  |
| HR_Recruitment_Process - Exercise_8_solution.twx  |
| HR_Recruitment_Process - Exercise_9_solution.twx  |
| HR_Recruitment_Process - Exercise_10_solution.twx |
| HR_Recruitment_Process - Exercise_11_solution.twx |
| HR_Recruitment_Process - Exercise_12_solution.twx |
| HR_Recruitment_Process - Exercise_13_solution.twx |
| HR_Recruitment_Process - WB835_Solution.twx       |
| HR_Case_Solution - WB835_Solution                 |

There is a working solution for each exercise. The solution files follow the naming convention used in Exercise 1 when exporting your process application. You can import solution files to:

- **Compare your work against a known working solution** to troubleshoot an issue in your current exercise.

To compare your current work against a known solution, import the solution file labeled with the exercise you are working on.

- **Continue working the exercises in the course** after having an issue in a previous exercise.

If you are unable to troubleshoot an issue, you can continue the next exercise by importing the solution for the current exercise.

- **Skip to an exercise of interest.**

If you want to skip ahead to an exercise, you can import the solution file of the exercise previous to that exercise. For instance, if you want to skip to Exercise 10, import the solution file: HR\_Recruitment\_Process - Exercise\_9\_solution.twx. After importing the file, you can follow the instructions for exercise 10.

Be sure to archive, export, and delete your current working solution beforehand to prevent file conflicts. You can reimport your solution after archiving and deleting the imported solution file. The remaining sections provide instructions for these three scenarios.

## Part 1: Export current process application

Before performing any of the subsequent sections, you need to export and delete your current process application to prevent file conflicts. You performed this action in exercise 1. The same steps are here for your reference. Be sure to perform these before importing another solution file.

- 1. Create a snapshot

Only process applications with snapshots can be exported. If you have an existing snapshot you can export, you can skip creating the snapshot.

- a. Access the Workflow Center and click the **Process apps** tile.
- b. Click **Details** on the HR Recruitment Process tile.

The screenshot shows the IBM Workflow Center interface. At the top, there's a navigation bar with icons for search, help, notifications, and user author1. Below it is a header bar with the title "IBM Workflow Center". The main content area is titled "Process apps (8)". It includes a search bar, a filter dropdown set to "Recently updated", and two buttons: "Import" and "Create". A message below the search bar says "Displaying 8/8 projects". There are three project cards visible:

- HR Recruitment Process**: Contains HR processes used for recruitment. Its "Details" button is highlighted with a red box.
- Performance**
- Process Portal**: Contains Process Portal.

- c. Select the **Snapshots** tab.
  - d. Open the list of Options by clicking the three dots on right side of the **Current** tip.
- The tip is the current working version of a project.

e. Select **Create a snapshot**

Back to process apps /

## HR Recruitment Process

[Open in Designer](#)

Overview    **Schemas**    Permissions    History    Comments

Schemas (0) ⓘ

All

| Name      | Status  | Created                | Last updated                      | Target environment |
|-----------|---------|------------------------|-----------------------------------|--------------------|
| → Current | author1 | 9/22/2020, 12:51:50 PM | author1<br>9/22/2020, 12:51:50 PM | Traditional        |

[Create a snapshot](#)

[Open in Designer](#)

f. Enter the name of the exercise you are working on such as `Exercise 1 solution` for the name.

**Create a snapshot**

Snapshot name

Description

Highlight your text to see formatting options.

*Description*

[Cancel](#) [Create](#)

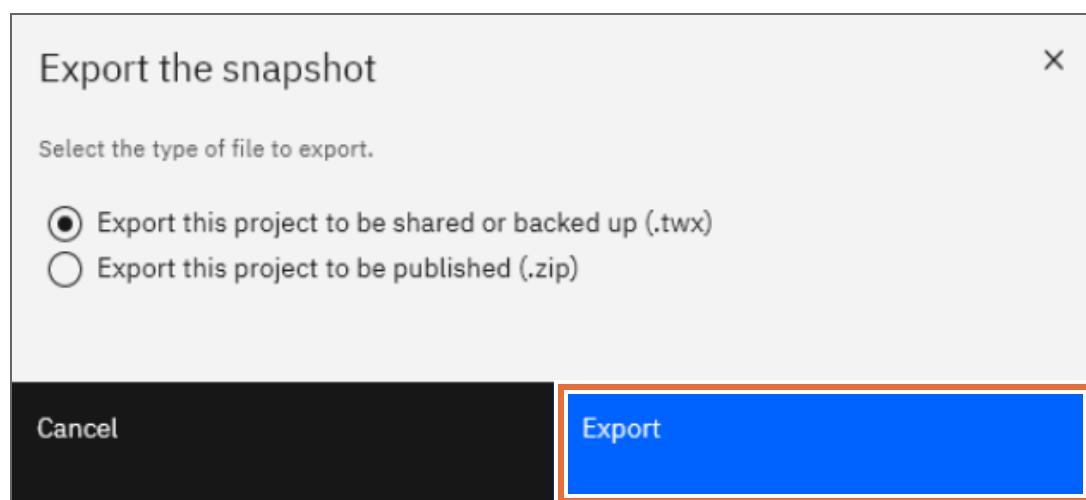
g. Click **Create**.

- 2. Export the application. Exporting the application also exports the toolkit that it depends on.  
You can only export snapshots.

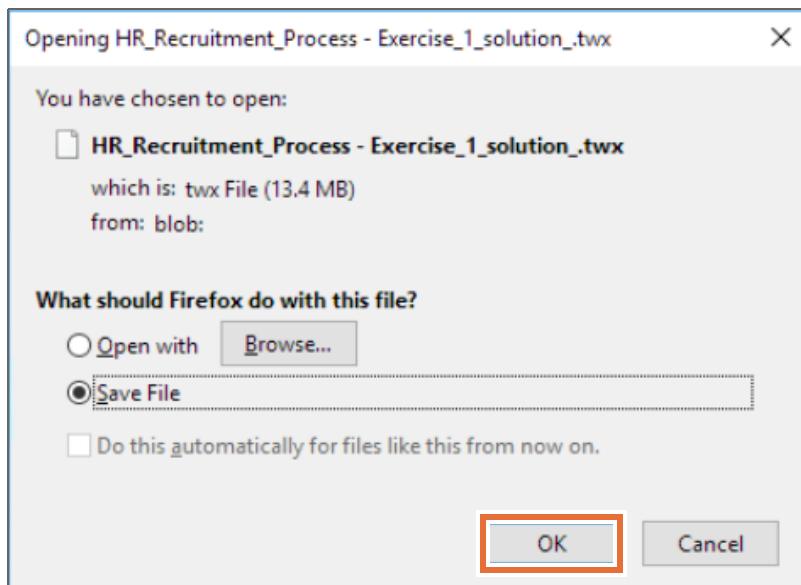
- a. Open the list of Options for the snapshot and select **Export**

| Name                        | Status | Created                           | Last updated                      | Target environment |
|-----------------------------|--------|-----------------------------------|-----------------------------------|--------------------|
| → Current                   | New    | author1<br>9/22/2020, 12:51:50 PM | author1<br>9/22/2020, 12:51:50 PM | Traditional        |
| → Exercise 1 solution (E1S) | New    | author1<br>9/22/2020, 3:35:42 PM  | author1<br>9/22/2020, 3:35:42 PM  | Traditional        |

- b. In the **Export the snapshot** window, click **Export**. It may take a minute or two to generate the .twx file.



- \_\_\_ c. Click **OK** to save the file. The application is saved to the default folder, which is the Downloads directory.



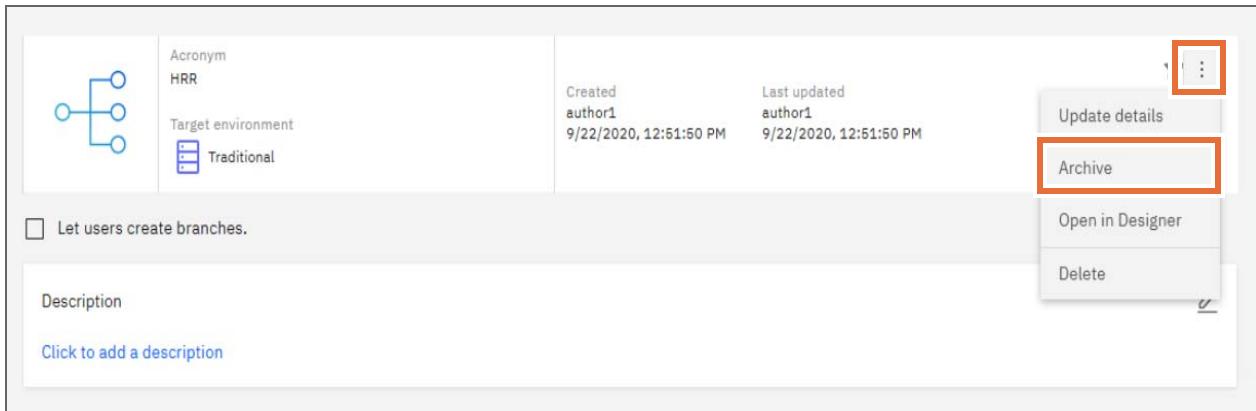
- \_\_\_ d. Verify that the application is listed in the downloads directory.
- \_\_\_ 3. Archive the process application and delete it. Although you don't have to archive an application before deleting it, it is a good practice. Archiving an application also archives the toolkit that it depends on.



### Important

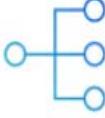
Don't archive a toolkit that other projects depend on nor use archived toolkits when creating toolkit dependencies in Process Designer. Archived items are not part of the active library.

- \_\_\_ a. Return to the **Overview** tab.
- \_\_\_ b. Open the list of Options for the application and select **Archive**.

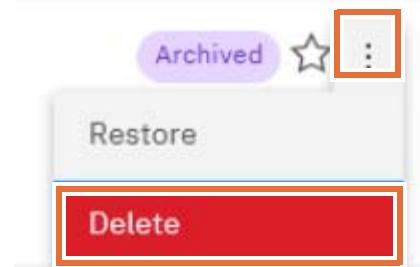


- \_\_\_ c. In the window that opens, click **Archive** to confirm the action. Clicking **Archive** does not delete the application; it removes it from the available **Process Apps** and makes it ready for deletion.

An **Archived** tag appears in the application Overview tile.

|                                                                                   |                                                                                                                                           |                                              |                                                   |                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Acronym<br>HRR<br><br>Target environment<br> Traditional | Created<br>author1<br>9/22/2020, 12:51:50 PM | Last updated<br>author1<br>9/23/2020, 11:14:26 AM | <b>Archived</b>   |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- \_\_\_ d. Open the list of Options for the application and select **Delete**.



- \_\_\_ e. In the window that opens, click **Delete** to confirm the deletion. The process application is now deleted.

You are returned to the list of Process apps.

- \_\_\_ f. Verify that **HR Recruitment Process** is no longer listed.

You can now import an existing solution file.

## Part 2: Perform troubleshooting

- **Compare your work against a known working solution**

After exporting your current working solution, archiving, and then deleting it from the repository, you can import an existing working solution.

If you are having issues in your current exercise, import the solution file corresponding to that exercise as listed in the table. Review the working solution to determine the cause of your issue.

After you determine the cause of the issue, you can export and delete the solution and re-import your working solution to apply the fix.

- **Continue working the exercises in the course**

After exporting your current working solution, archiving, and then deleting it from the repository, you can import an existing working solution.

After importing the working solution for the exercise you cannot troubleshoot, you can continue with the next exercise using the imported solution file. You can continue to develop using this file for the remainder of the course.

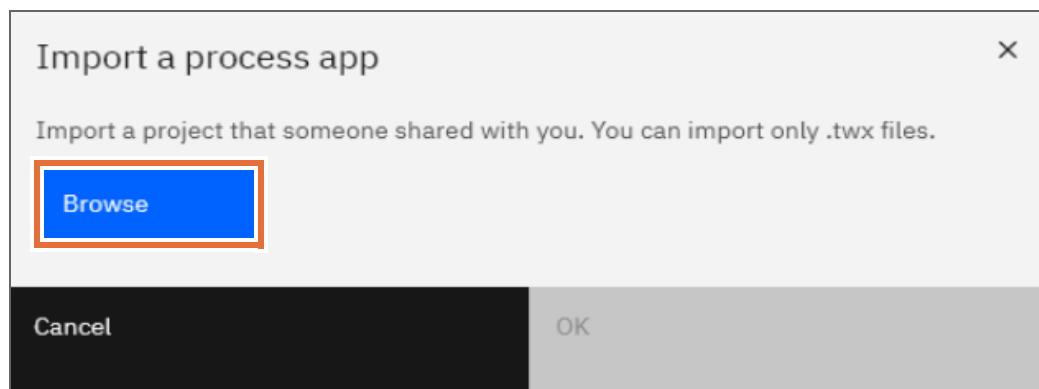
- **Skip to an exercise of interest**

If you are just starting the course and would like to skip to an exercise. You can import the solution file for the exercise prior to the one you would like to perform. You can continue to develop using this file for the remainder of the course. For instance, if you want to skip to Exercise 10, import the solution file: `HR_Recruitment_Process - Exercise_9_solution.twx`. After importing the file, you can follow the instructions for exercise 10.

### **Part 3: Import a solution file**

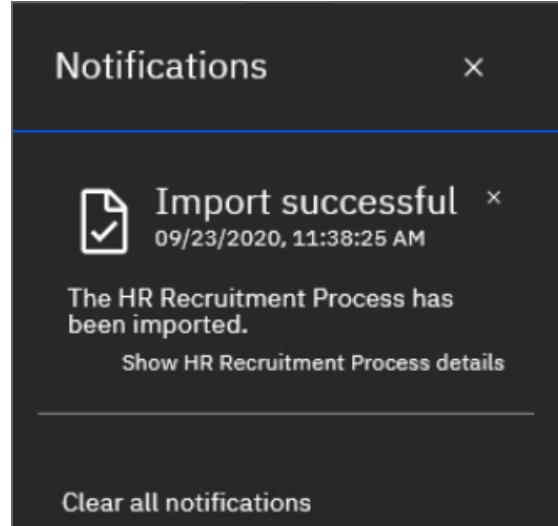
In this section, you import a process application solution for a specific exercise for troubleshooting purposes. This import also helps in making this course modular as you can then work on any exercise in any order. Although, for a good learning experience, it is highly recommended to do the exercises in the order they are provided.

- \_\_\_ 1. Import a process application.
  - \_\_\_ a. Access the Workflow Center and click the **Process apps** tile.
  - \_\_\_ b. On the list of Process apps, click **Import**.
  - \_\_\_ c. Click **Browse**.

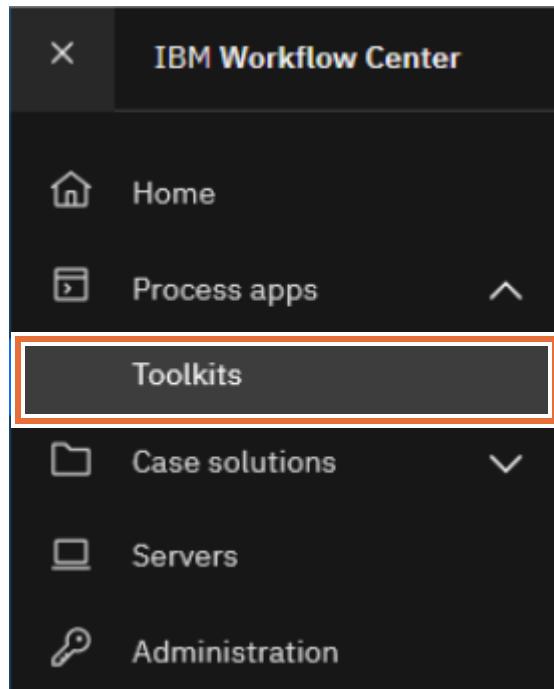


- \_\_\_ d. Navigate to the `c:\solutions` folder and select the file for the exercise you would like to import:  
`HR_Recruitment_Process - Exercise_1_solution.twx`
- \_\_\_ e. Click **Open**.
- \_\_\_ f. Click **OK**.

- \_\_\_ g. The Notifications window displays the message



- \_\_\_ h. Close the **Notifications** window by clicking the **X** in the upper right corner.  
\_\_\_ i. Verify that the **HR Recruitment Process** application is listed.  
\_\_\_ j. Open the list of Toolkits.



- \_\_ k. If the solution file contains a Toolkit, verify the Toolkit is listed.

Toolkits (10)

Recently updated All

Import Create +

Displaying 10/10 projects

|                                                                                                                                                                     |                                                |                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Hiring Requisition Toolkit</b><br>This toolkit contains all common artifacts that are associated with the HR Recruitment Processes.<br><a href="#">Details →</a> | <b>UI Toolkit</b><br><a href="#">Details →</a> | <b>SAP Guided Workflow (deprecated)</b><br>Toolkit of SAP Guided Workflow<br><a href="#">Details →</a> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------|

- \_\_ l. Click **Details** on the **Hiring Requisition Toolkit** tile.
- \_\_ m. Click **Let users update the toolkit** to allow changes to be made.

|                        |                |                                             |                                                  |                                                        |
|------------------------|----------------|---------------------------------------------|--------------------------------------------------|--------------------------------------------------------|
|                        | Acronym<br>HRT | Created<br>author1<br>9/17/2020, 2:59:04 PM | Last updated<br>author1<br>9/17/2020, 2:59:09 PM | <input type="checkbox"/> Let users update the toolkit. |
| Target environment<br> | Traditional    |                                             |                                                  |                                                        |

Toolkit updates can be managed using this option.

## End of Appendix



IBM Training



© Copyright International Business Machines Corporation 2020.