

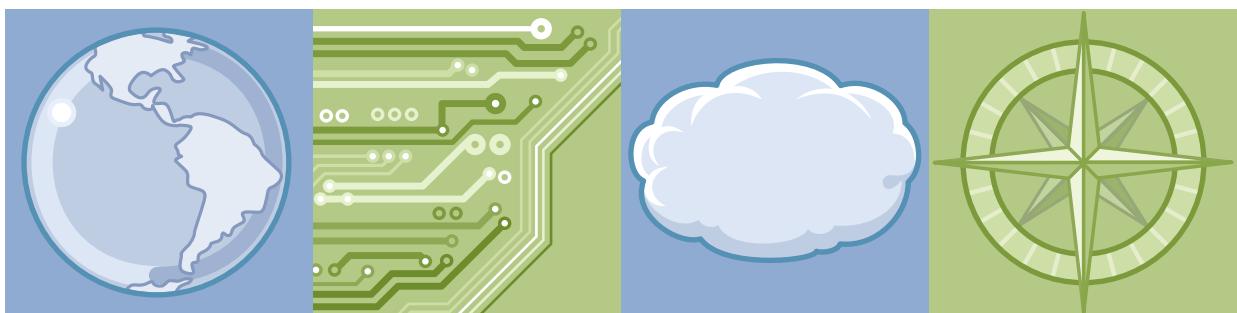


IBM Training

Student Exercises

Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I

Course code WB860 ERC 1.2



IBM Cloud
Middleware

Trademarks

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:

AIX®	CICS®	ClearCase®
DataPower®	DB™	DB2®
developerWorks®	Express®	IMS™
Initiate®	Lombardi®	Notes®
PartnerWorld®	Power®	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
Sametime®	WebSphere®	z/OS®

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Evolution® is a trademark or registered trademark of Kenexa, an IBM Company.

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

June 2016 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Contents

Trademarks	vii
Exercises description	ix
Exercise 1. Exploring IBM Process Designer and IBM Process Portal	1-1
Part 1: Start IBM Process Center	1-3
Part 2: Open a business process activity in IBM Process Designer	1-9
Part 3: Explore a business process in IBM Process Designer	1-13
Part 4: Use the Playback feature to examine a running business process in IBM Process Designer Inspector	1-19
Part 5: Use a coach to work with a running business process	1-21
Part 6: Wire activities together in a business process	1-24
Part 7: Examine the Hiring Sample Advanced BPD and the associated users	1-33
Part 8: Examine the BPD and start an instance	1-38
Exercise review and wrap-up	1-48
Exercise 2. Exploring IBM Integration Designer, part I	2-1
Exercise instructions	2-3
Part 1: Examine IBM Integration Designer capabilities and preferences	2-3
Part 2: Use the help menu to search for assistance	2-9
Part 3: Use the Installation Manager to find product updates	2-11
Part 4: Browse the Business Integration perspective and views	2-12
Part 5: Examine the modules and libraries of a business integration project	2-16
Part 6: Browse the IBM Integration Designer graphical editors	2-25
Part 7: Use the IBM Integration Designer documentation capabilities	2-27
Part 8: Use the Task Flows view	2-31
Exercise review and wrap-up	2-36
Exercise 3. Exploring IBM Integration Designer, part II	3-1
Exercise instructions	3-2
Part 1: Assemble an SCA application	3-2
Part 2: Examine project components	3-4
Part 3: Explore staging projects	3-5
Part 4: Use the IBM Integration Designer test environment	3-8
Part 5: Use cross-component testing to examine a running process application	3-26
Part 6: Deploy a library globally	3-29
Exercise review and wrap-up	3-38
Exercise 4. Working with web services	4-1
Exercise instructions	4-2
Part 1: Import an external Web Service Description Language file into IBM Integration Designer	4-2
Part 2: Create a Service Component Architecture component from a web service interface file	4-8
Part 3: Use the integrated test client to test a web service	4-16
Part 4: Use a web service export to expose an existing IBM Process Server application	4-20
Exercise review and wrap-up	4-32

Exercise 5. Creating business objects and shared interfaces	5-1
Exercise instructions	5-2
Part 1: Compare business objects between IBM Process Designer and IBM Integration Designer	5-2
Part 2: Implement business objects and define their elements	5-7
Part 3: Implement interfaces that use business objects as inputs and outputs	5-10
Exercise review and wrap-up	5-15
 Exercise 6. Creating a business process, part I	 6-1
Exercise instructions	6-2
Part 1: Create a business process	6-2
Part 2: Create process variables	6-7
Part 3: Implement WS-BPEL interface partners and reference partners	6-10
Part 4: Define a business process	6-17
Part 5: Compare business processes between IBM Integration Designer and IBM Process Designer	6-25
Exercise review and wrap-up	6-27
 Exercise 7. Creating a business process, part II	 7-1
Exercise instructions	7-2
Part 1: Implement basic and structured BPEL activities in a business process	7-2
Part 2: Compare the BPEL to the BPD in IBM Process Designer	7-42
Exercise review and wrap-up	7-44
 Exercise 8. Creating a business process, part III	 8-1
Exercise instructions	8-2
Part 1: Use data maps to transform process data	8-2
Part 2: Use context variables to create a runtime process description	8-16
Part 3: Assemble an SCA application that contains a business process	8-18
Part 4: Test a business process in the IBM Integration Designer test environment	8-30
Exercise review and wrap-up	8-37
 Exercise 9. Creating business rules	 9-1
Exercise instructions	9-3
Part 1: Create a rule group component that contains business rules in rule sets and decision tables	9-3
Part 2: Incorporate a rule group component into an assembly diagram	9-23
Part 3: Test a business rule group in the integrated test client	9-26
Part 4: Use the Business Rule Manager web client to interact with business rules at run time	9-29
Exercise review and wrap-up	9-35
 Exercise 10. Implementing WebSphere (JCA) adapters	 10-1
Exercise instructions	10-2
Part 1: Configure the WebSphere Adapter for Flat Files	10-2
Part 2: Use the external service tool to generate artifacts that are used in an application	10-3
Part 3: Incorporate adapter-related SCA artifacts into an assembly diagram	10-11
Part 4: Test an adapter in the IBM Integration Designer test environment	10-14
Part 5: Create the RecordIneligibleApplication Java component	10-16
Part 6: Test the RecordIneligibleApplication Java component	10-22
Exercise review and wrap-up	10-27

Exercise 11. Creating mediation services, part I	11-1
Exercise instructions	11-3
Part 1: Create a mediation module that contains a Mapping primitive	11-3
Part 2: Define an XML data map	11-6
Part 3: Test a mediation module that contains a Mapping primitive	11-17
Exercise review and wrap-up	11-27
Exercise 12. Creating mediation services, part II	12-1
Exercise instructions	12-2
Part 1: Create a mediation module that contains a message filter mediation primitive and a Mapping primitive	12-2
Part 2: Define an XML data map	12-13
Part 3: Test a mediation module that contains a message filter mediation primitive and a Mapping primitive	12-34
Exercise review and wrap-up	12-50
Exercise 13. Exploring Business Space	13-1
Exercise instructions	13-2
Part 1: Use the Business Space clients to work with human tasks	13-2
Part 2: Creating a business space (your own workspace) in the Business Space user interface	13-5
Exercise review and wrap-up	13-28
Exercise 14. Using component tests	14-1
Exercise instructions	14-2
Part 1: Create a component test project with an operation-level test case	14-2
Part 2: Create a component test project with a scenario-based test case	14-20
Part 3: Run component test project test suites in the IBM Integration Designer integrated test environment	14-37
Exercise review and wrap-up	14-40
Exercise 15. Bringing the UTE Process Server online	15-1
Exercise instructions	15-2
Part 1: Start IBM Process Center and the Process Server test environment	15-2
Part 2: Verify the status of the UTE Process Server	15-5
Part 3: Examine the connection properties	15-6
Part 4: Define the authentication alias in Process Server	15-12
Part 5: Verify the status of the online Process Server	15-17
Part 6: Configure SSL communication between Process Center and test environment Process Server	15-19
Part 7: Restart the server environment and verify the UTE server status and name change	15-28
Exercise review and wrap-up	15-30
Exercise 16. Exploring IBM Process Center	16-1
Exercise instructions	16-3
Part 1: Explore IBM Process Center repository	16-3
Part 2: Create a Toolkit in IBM Process Center	16-6
Part 3: Associate IBM Integration Designer artifacts with the toolkit	16-9
Part 4: Generate access to process applications and toolkits	16-22
Part 5: Removing the Account Services process application	16-34

Part 6: Deleting the Account Services process application snapshot	16-37
Exercise review and wrap-up	16-39

Exercise 17. Implementing Advanced Integration services 17-1

Exercise instructions	17-3
Part 1: Start the IBM Process Center environment	17-3
Part 2: Create an Advanced Integration service (AIS) by using IBM Process Designer	17-4
Part 3: Implement the AIS in IBM Integration Designer	17-15
Part 4: Unit Test the Update Systems of Record AIS Advanced Integration service in IBM Integration Designer	17-21
Part 5: Run the BPD in IBM Process Designer and execute the AIS	17-24
Part 6: Implement the revised AIS in IBM Integration Designer	17-28
Part 7: Unit Test the updated AIS and the new BPEL component in IBM Integration Designer	17-41
Part 8: Run the BPD in IBM Process Designer and execute the revised AIS and invoke the new BPEL component	17-44
Exercise review and wrap-up	17-49

Appendix A. Re-creating the exercise environment A-1

Part 1: Installing the IBM Integration Designer test environment	A-1
Part 2: Installing the IBM Business Process Manager Advanced V8.5.7	A-7
Part 3: Installing IBM Process Designer	A-10
Part 4: Define users with Process Administrative Console	A-12
Part 5: Stop the Process Center environment	A-15
Part 6: Download and install Adobe Flash player	A-16
Part 7: Download and install Adobe Reader	A-16
Part 8: Create desktop shortcuts for starting the Process Center environment	A-17
Part 9: Extending the default one-year expiration date for WebSphere certificate (Optional)	A-21

Appendix B. Creating the artifacts that are imported in the exercises B-1

Part 1: Creating the credit score web service	B-1
Part 2: Creating the artifacts in the FoundationLibrary	B-5

Appendix C. Account verification solution overview C-1

Part 1: The business process model	C-1
Part 2: Paths of the business process solution	C-2

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:

AIX®	CICS®	ClearCase®
DataPower®	DB™	DB2®
developerWorks®	Express®	IMS™
Initiate®	Lombardi®	Notes®
PartnerWorld®	Power®	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
Sametime®	WebSphere®	z/OS®

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Evolution® is a trademark or registered trademark of Kenexa, an IBM Company.

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

Exercises description

This course includes the following exercises:

- Exercise 1: Exploring IBM Process Designer and IBM Process Portal
- Exercise 2: Exploring IBM Integration Designer, part I
- Exercise 3: Exploring IBM Integration Designer, part II
- Exercise 4: Working with web services
- Exercise 5: Creating business objects and shared interfaces
- Exercise 6: Creating a business process, part I
- Exercise 7: Creating a business process, part II
- Exercise 8: Creating a business process, part III
- Exercise 9: Creating business rules
- Exercise 10: Implementing WebSphere (JCA) adapters
- Exercise 11: Creating mediation services, part I
- Exercise 12: Creating mediation services, part II
- Exercise 13: Exploring Business Space
- Exercise 14: Using component tests
- Exercise 15: Bringing the UTE Process Server online
- Exercise 16: Exploring IBM Process Center
- Exercise 17: Implementing Advanced Integration services

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 done. If required, the substeps provide guidance on completing the action.

 **Example**

— 1. Create a user account named **ADMIN**.

— a. Right-click **My Computer** and choose **Manage** from the menu.

— b. Expand **Local Users and Groups**.

... *continue*

In this example, the creation of a user account is the action to be done. The substeps underneath provide specific guidance on how to create a user account. (In this example, the instructions are for Windows operating system.) Words that are highlighted in bold represent menu items, field names, and so forth.

Each step and substep has an underscore that precedes it. 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 have interruptions during a lengthy exercise.

User IDs and passwords

The following table contains a list of user ID and password information for this course.

Entry point	User ID	Password
VMware image	Administrator	web1sphere
Windows 2012 R2	Administrator	web1sphere
Process Center Console	pcdeadmin	web1sphere
IBM Process Designer	pcdeadmin	web1sphere
Administration console for IBM Process Manager Advanced	bpmadmin	web1sphere
Administration console for Process Server test environment	admin	web1sphere

Exercise 1. Exploring IBM Process Designer and IBM Process Portal

What this exercise is about

This exercise introduces the IBM Process Designer development environment. It shows you some of the tools that are associated with creating, editing, and managing process applications. It also introduces the collaboration capabilities of IBM Process Portal.

What you should be able to do

After completing this exercise, you should be able to:

- Start IBM Process Designer
- Open a business process activity in IBM Process Designer
- Explore a business process in IBM Process Designer
- Use the Playback feature to examine a running business process in IBM Process Designer Inspector
- Use a coach to work with a running business process
- Wire activities together in a business process
- Start IBM Process Portal and explore its collaboration capabilities

Introduction

IBM Process Designer is a member of a family of products available with IBM Business Process Manager V8.5.7. It is a single authoring environment to design business process applications in editors that use Business Process Modeling Notation (BPMN).

Activities that are added to a process application might be implemented with a wide variety of activities. These activities include tasks that are started from the system, scripts, or user-initiated tasks (also called human tasks). The desktop version of IBM Process Designer provides a simple editor for creating client-facing user interfaces, called “coaches”, for allowing users to interact with the business process. The web version provides a more fully featured editor for creating web-based client-facing user interfaces. The web editor includes a wider range of web technologies while also using existing coach views.

Business processes might be tested in the integrated test environment that is called the “Inspector.” The Inspector allows users to play back a sample of the business process and monitor it for performance and usability. It also

starts the coaches or client-side human services in a standard browser, and follows the progress of the process application steps.

Completed business process applications can be saved to a centralized repository maintained by IBM Process Center. Different stages of the process can be saved to the repository, allowing dynamic version control. These “snapshots” can then be deployed to the server for execution.

IBM Process Portal provides an interface for you to work on your assigned tasks and provide a highly collaborative work experience with increased social capabilities.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

General exercise information

This section provides general information about the exercises in this course. Review this section before starting the exercises.



Important

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

C:\labfiles

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

Exercise instructions

Part 1: Start IBM Process Center

In this portion of the exercise, you start the web-based IBM Process Center. In the sections that follow, you open the web-based Process Designer and explore some basic business process definitions and some of the capabilities of this tool.



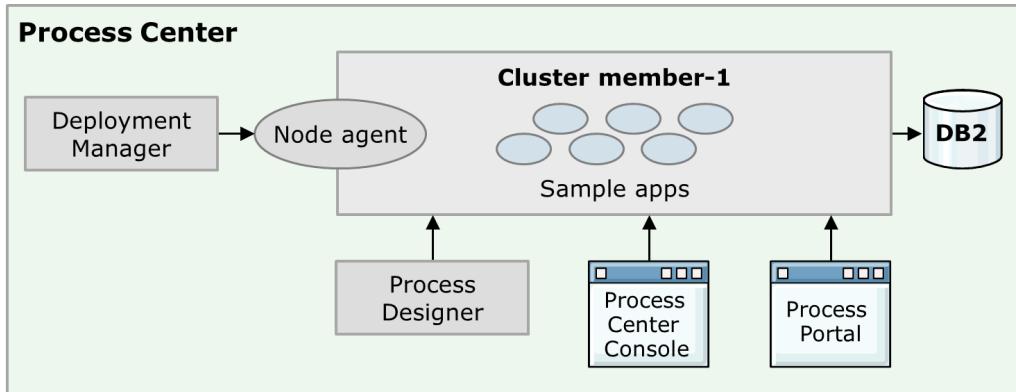
Note

For IBM BPM on Cloud users:

You do not need to start the Process Center. It is already running in your cloud environment.

IBM Process Designer maintains its business process definition (BPD) artifacts in the centralized IBM Process Center repository. Both the web and desktop versions of the IBM Process Designer use the Process Center repository. To create, view, edit, or inspect a BPD in IBM Process Designer, you must first start the IBM Process Center repository. Process Center is installed and configured as a single cluster topology. The following diagram depicts different processes that you are going to

start and their relationship within the Process Center cell.



- 1. Start the Process Center deployment manager. The deployment manager controls the behavior of the node agent and cluster members.
 - a. On your Windows desktop in the lab environment, select the **Start Process Center deployment manager** shortcut. It might be necessary to select the shortcut to completely view its description. Double-click the shortcut or press Enter to start the server.



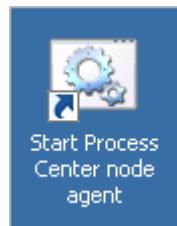
A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts.

- b. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.

```

Start Process Center deployment manager
CWUPO0001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
          C:\IBM\BPM\v8.5\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 2820
Press any key to continue . . . =
  
```

- 2. Start the Process Center node agent. The node agent connects the deployment manager to the cluster members.
- a. On your Windows desktop, select the **Start Process Center node agent** shortcut. Double-click the shortcut or press Enter to start the server.



- b. A DOS command window is displayed. Press any key to continue when prompted.

```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
  C:\IBM\BPM\v8.5\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 3220
Press any key to continue . . .
```

- 3. Start the single cluster. The single cluster contains the servers that run the Process Center application.
- a. On your Windows desktop, select the **Start Process Center Cluster** shortcut. Double-click the shortcut or press Enter to start the cluster member.



- b. A DOS command window is displayed. Press any key to continue when prompted.

```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
  C:\IBM\BPM\v8.5\profiles\Node1Profile\logs\SingleClusterMember1\startServer.log
ADMU0128I: Starting tool with the Node1Profile profile
ADMU3100I: Reading configuration for server: SingleClusterMember1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server SingleClusterMember1 open for e-business; process id is 3964
Press any key to continue . . .
```

- ___ 4. Verify that the Process Center Single cluster is started.
- ___ a. On your Windows desktop, select the **Deployment Manager Administrative Console** shortcut. Double-click the shortcut or press Enter to go to the console.

**Hint**

You can start a browser and go to <https://localhost:9044/ibm/console> rather than using the shortcut.

- ___ b. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.

I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...

- ___ c. On the next screen, accept the defaults and click **Confirm Security Exception**.

Permanently store this exception

Confirm Security Exception

- ___ d. Log in by entering `bpmadmin` in the **User ID** field and `web1sphere` in the **Password** field, and click **Log in**.

- __ e. Click **Servers > Deployment Environments**. Verify that the deployment environment is started and the status is green.

Select	Status	Deployment Environment Name
<input type="checkbox"/>		ProcessCenter

Total 1

- __ f. Click **Servers > Clusters > WebSphere application server clusters** to verify that the single cluster is running.

Select	Name	Status
<input type="checkbox"/>	SingleCluster	

Total 1

- __ g. Click **Logout** to exit.

- __ h. Close the browser.



Hint

You can also start IBM Process Center from a DOS command window.

- a. If you open a DOS command window, browse to the following directory:

C:\IBM\BPM\v8.5\profiles\dmgrProfile\bin

- b. Type the following command to start the IBM Process Center Deployment Manager:
`startManager.bat`
- c. To start the node agent and the cluster, browse to the following directory:
`C:\IBM\BPM\v8.5\profiles\Node1Profile\bin`
- d. Type the following command to start the IBM Process Center Node agent:
`startNode.bat`
- e. Type the following command to start the IBM Process Center Single Cluster:
`startServer.bat SingleClusterMember1`

When installed, IBM Process Center installs and configures several databases across a single instance to form the IBM Process Center repository. In the example of this course, the name of that instance is `BPMINST`. Several databases are installed in that instance.

5. To start the web-based IBM Process Designer, you must first log in to the Process Center and then open an application in the Process Designer. Double-click the **IBM BPM Process Center** icon on the desktop.



Hint

You can start a browser and go to `http://localhost:9081/ProcessCenter` rather than using the shortcut.

- a. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.

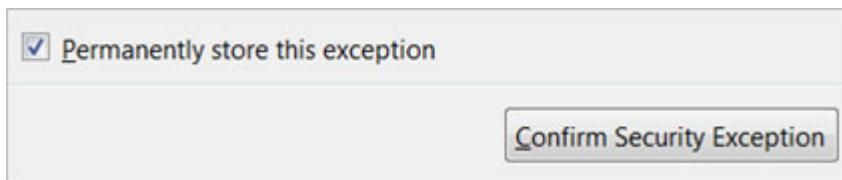
I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

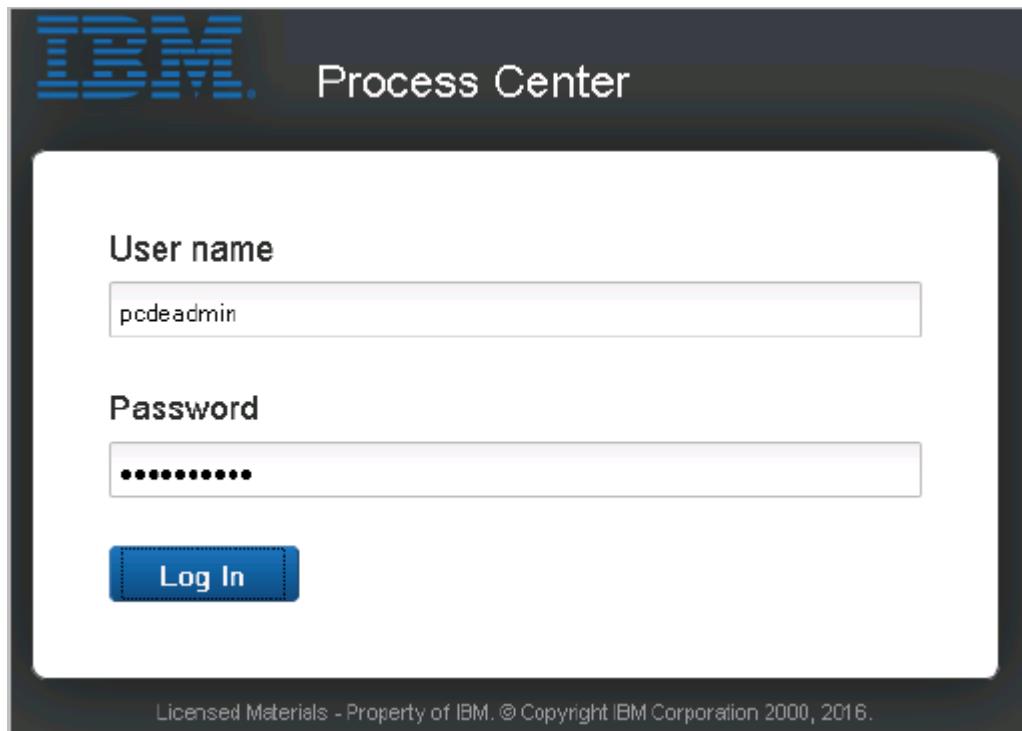
Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...

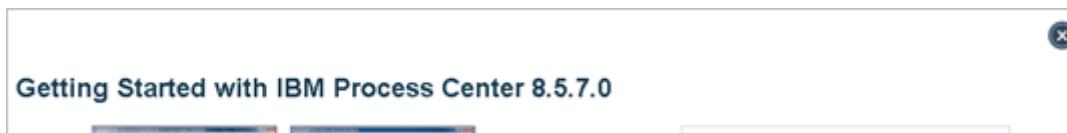
- ___ b. On the next screen, accept the defaults and click **Confirm Security Exception**.



- ___ 6. Enter pcdeadmin in the **User name** field and web1sphere in the **Password** field, and click **Log In**.



- ___ 7. Close the **Getting Started with IBM Process Center 8.5.7.0** display.



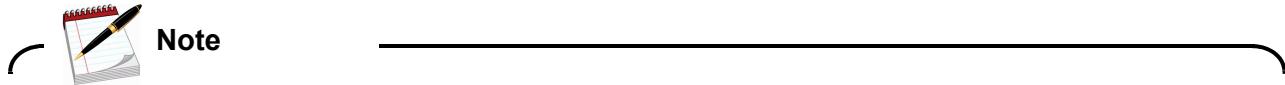
Part 2: Open a business process activity in IBM Process Designer

When IBM Process Designer completes authenticating your user name and password against the IBM Process Center repository, the Process Center perspective of IBM Process Designer is displayed. This perspective contains a list of all the process applications that are currently in your IBM Process Center repository. Process applications are containers in the IBM Process Center repository for the process models and supporting implementations that are created in IBM Process Designer. A process application includes process models, the services to handle implementation of activities and integration with other systems, and any other assets that are required to run the processes.

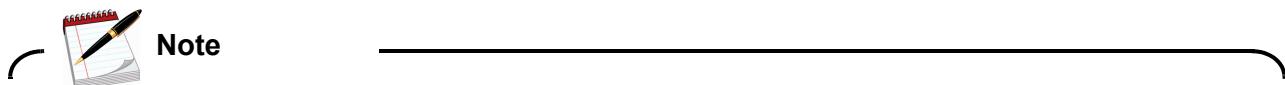
You can use this perspective to work with these business process applications by opening the process application in design mode, by using IBM Process Designer, or by creating snapshots. A

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.

- 1. Examine the Process Center perspective.



Your screen might not exactly match the one depicted.



For IBM BPM on Cloud users:

The Process Center perspective is the same when the Process Center is on the cloud.

The Process Center repository is the default view. This workspace is an important tool for designing process applications and storing them on the IBM Process Center repository. You are encouraged to open each of these components and explore as you read.

- 1) **Process Apps view:** This view contains a list of all the process applications that currently exist in the IBM Process Center repository. From this view, you can work with the process application in IBM Process Designer, or with its snapshots in the repository.
- 2) **Process applications:** Each entry in this list represents a process application that is stored in the repository, which can be designed or installed on an available IBM Process Server. The name of the process application is a hyperlink, which allows the user to manage the snapshots, examine the history, and work with the users who have access to the process library. You

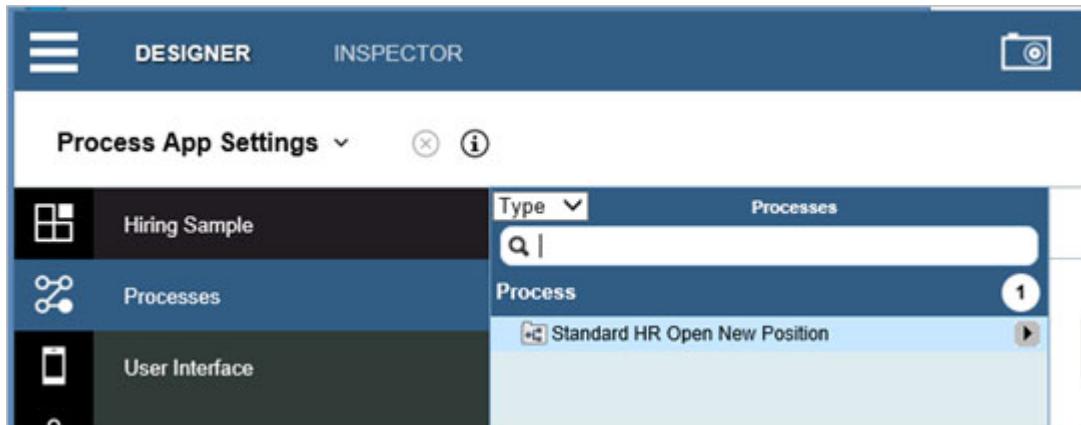
can use an icon that is attached to the process application to mark the process application as a favorite.

- 3) **Open in Designer shortcut:** Each process application in the list has a corresponding shortcut that opens that process application in design mode. Opening a process application in design mode grants you access to all the business processes, data objects, and implementations that are associated with that process application.
 - 4) **Create New Process App:** You can use this shortcut to create a business process application. You must provide a unique application name, a unique ID, and an arbitrary description. When the placeholder for the process application is saved, your new process application opens in design mode.
 - 5) **Toolkits view:** You can create toolkits to enable IBM Process Designer users to share library items across process applications. One toolkit can be shared across many process applications, and one process application can use many toolkits.
- Users who have access to a toolkit can create a dependency on the toolkit and use the library items within it for their process development efforts. To create and import toolkits, you must have access to the Process Center repository.
- 6) **Servers view:** This view contains all the process servers that are connected to the IBM Process Center repository. Authorized users can install snapshots of process applications on connected IBM Process Servers. For each server, you can see the snapshots that are currently installed.
 - 7) **Admin view:** From this view, you can do administrative tasks such as granting access to the IBM Process Center repository by administering users.

- 2. Open the **Hiring Sample** process application in IBM Process Designer.
 - a. Verify that you are in the **Process Apps** tab by clicking **Process Apps** at the top.
 - b. Hover your mouse over the process application that is named **Hiring Sample (HSS)**. On the right side of the **Hiring Sample (HSS)** process application, click the **Open in Designer** link.



- c. Click **Processes** and click **Standard HR Open New Position** to open the process application in the editor.



 **Note**

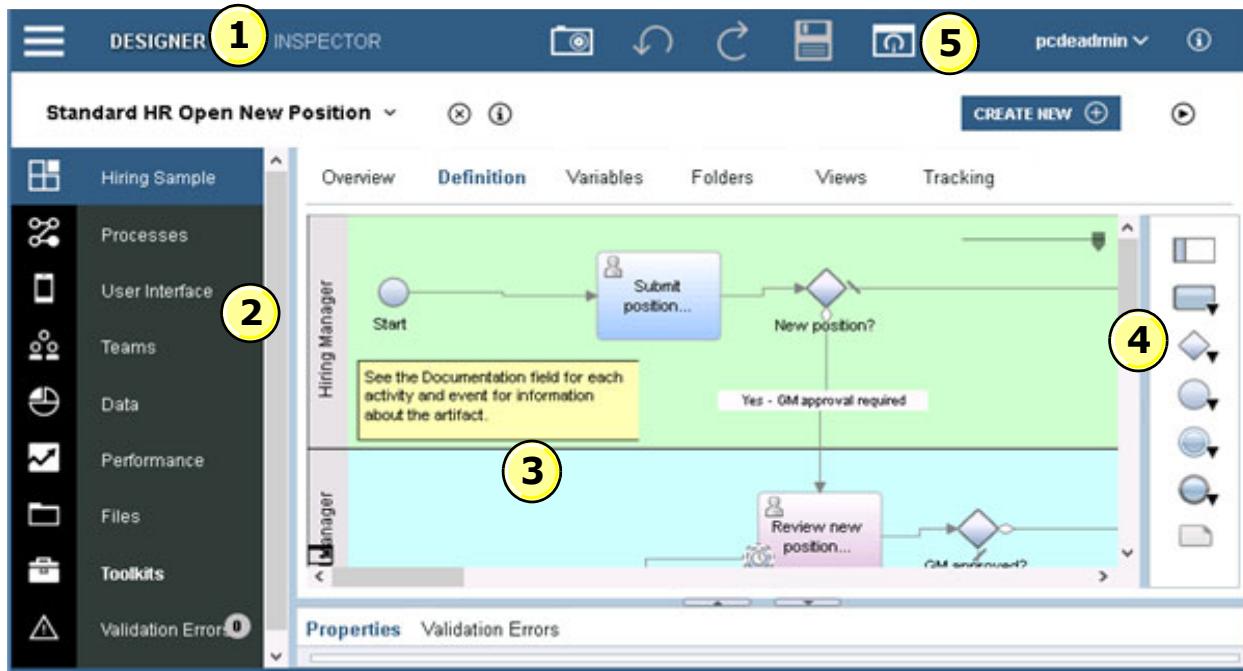
Depending on the artifact that you are working on, you use either the Eclipse-based Process Designer on the desktop, or the web Process Designer.

You create and edit business process definitions (BPD), services, and heritage user interfaces in the desktop Process Designer. You create and edit processes and responsive user interfaces in the web Process Designer. You can edit artifacts such as business objects in both the web and desktop designer. For a complete discussion of the web-based Process Designer, see the IBM Business Process Manager v8.5.7 IBM Knowledge Center at:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbpm.main.doc/topics/cbpm_whatsnew.html?lang=en

Part 3: Explore a business process in IBM Process Designer

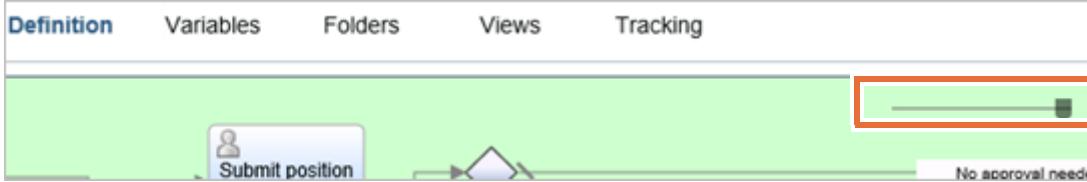
The business process application opens in IBM Process Designer. (You are encouraged to open each of these components and explore as you read.)



- 1) **Designer, Inspector:** These represent two different views available in IBM Process Designer. The default, Designer view, is used for building business process definitions by using tools that are commonly found in Business Process Modeling Notation (BPMN). You can test the execution of your business process by starting it in the Inspector. The Inspector is a sample environment that can be used for debugging, tracing variables, and following the flow of your business process. The Optimizer is not available in the web-based version.
- 2) **Design palette:** The design palette offers a number of tools that are used in designing a business process application. Developers discover business processes, user interface designs, implementation tools, data object editors, and many other useful tools.
- 3) **Editor pane:** When you open a resource from the diagram or from the design palette, it is displayed in the editor pane.
- 4) **Tools:** The tools included here represent the common tools that can be used for each editor. In the preceding example, the tools are common for the business process that is open in the editor. Different tools would be provided for different implementations, such as human tasks, business rules, and coaches.

- 5) **Control icons:** The control icons at the top of the page provide quick access to snapshots, undo or redo actions, save work, and the Process Center view. These icons are unique to the web-based version of the Process Designer.

- 1. Examine the flow of the business process that is open in the design view.
- a. Maximize the window for readability. When the window is maximized, you can control the zoom level of the editor pane by moving the zoom slider in the upper right corner.



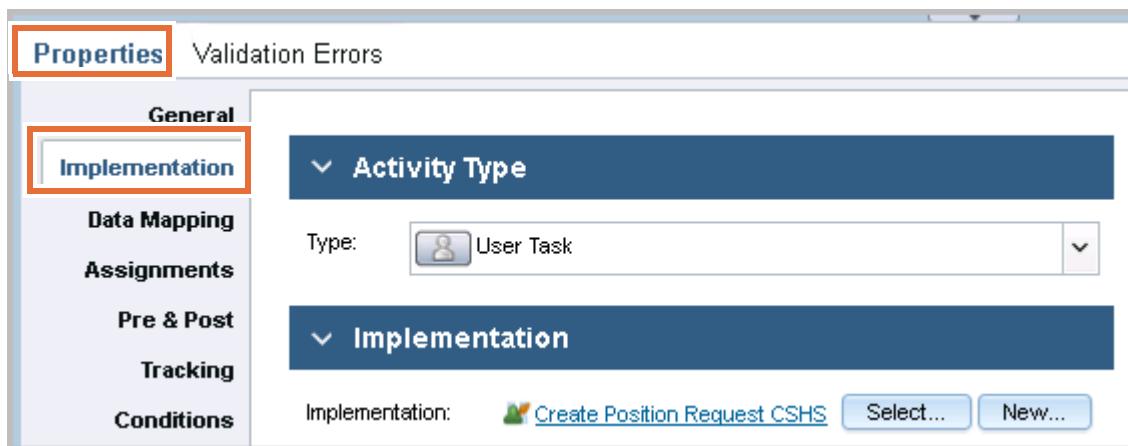
- b. Examine the process from the upper left to the lower right of the editor. Read the notes that are added along the paths. Notes are useful to follow the logic of a business process, even though they do not add any function to the design. Notes are to process designs as comments are to Java code.
- c. The business process is divided into horizontal swimlanes. Each swimlane represents a different actor upon the business process. Examine the labels for each swimlane along the left column. Name the actors for this process.

 **Note**

Swimlanes and activities are depicted with colors. These colors are arbitrary and are meaningful only to the project and developer. Colored swimlanes and activities have no special function in IBM Process Designer. The color of a swimlane or activity can be changed depending upon your project. By default, new swimlanes are colorless and new activities are gray.

- d. Along the paths of the business process are activities, such as **Submit position request**. Each activity represents a task that is assigned to the actor who owns the swimlane. In this example, the **Hiring Manager** must complete the **Submit position request** activity.
- e. Decision points divide the business process into different possible paths. For example, when the **Submit position request** activity is complete, a decision must be made: does the requisition require general manager (GM) approval? If approval is required, one path is selected and if not, another path. Each path is labeled.
- 2. Examine the implementations of the business process activities.
- a. Click the **Submit position request** activity in the flow.

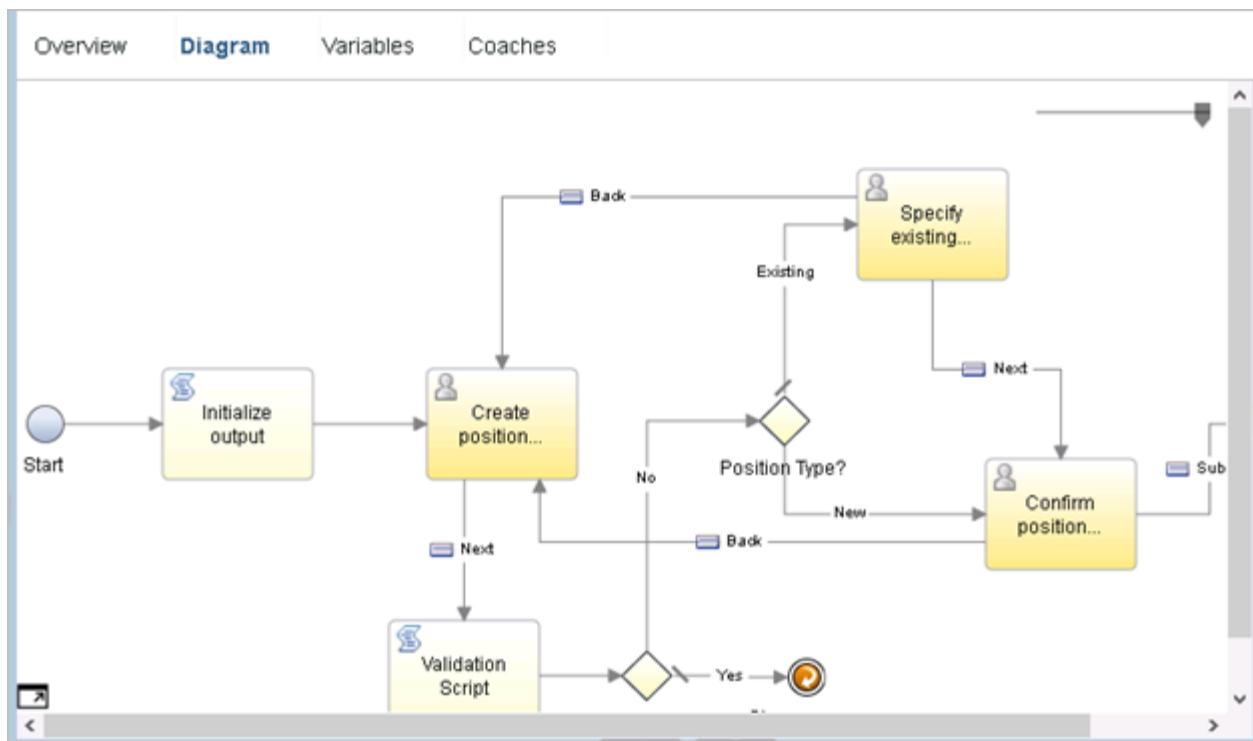
- ___ b. At the bottom of the window, examine the **Properties** tab. Click the **Implementation** option in the **Properties** tab.



The implementation of this activity is a service that is called **Create Position Request CSHS**. As indicated, this service is a hyperlink.

- ___ c. Click the link to see the implementation of the **Create Position Request CSHS** service.

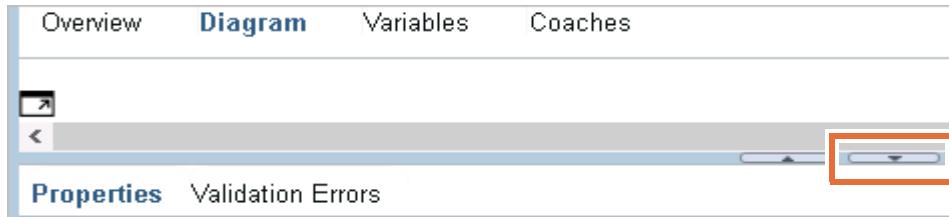
The **Create Position Request CSHS** service is a subflow within the **Hiring Sample** flow itself. This type of service implementation is a *human task*: a task that depends upon the input of an actor (in this case, the hiring manager) for its successful completion. The human tasks are colored and display an icon that suggests where human input is required.





Information

The implementation diagram does not appear when the Properties display is maximized. Click the down arrow icon to see the diagram.

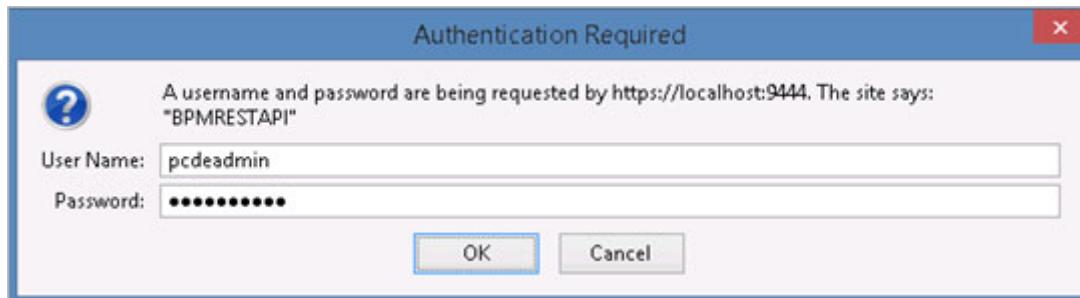


- ___ d. The subflow indicates that when the output is initialized, the next activity is to create the position request. If the position type is new, the user must provide position details. If it is an existing position, the user must provide details for that existing position and then confirm the position details. The subflow ends when **Submit** is clicked. Control is returned to the main flow from here. You can scroll to the right to view the **Submit** button.

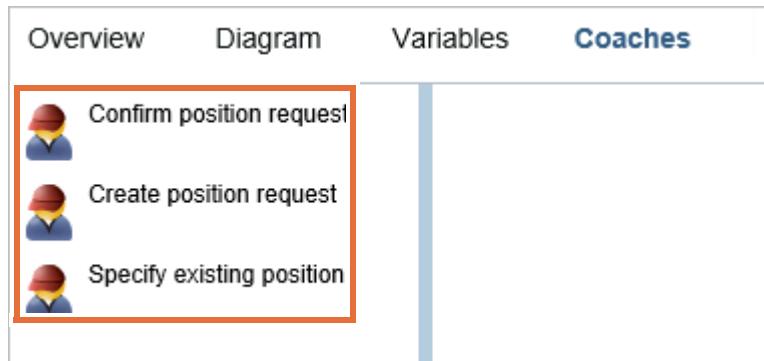


Important

You might see a request to enter a user name and password from BPMRESTAPI. Enter `pcdeadmin` for the User Name and `web1sphere` for the Password. Click **OK**.



- e. From the design pane of the **Create Position Request CSHS** service, click **Coaches**. In this example, each human task in the service relates to a *coach*. A coach is a client GUI that takes input from the actor (that is, the hiring manager) for the successful completion of the task.



- f. At the top of the editor pane, you see a drop-down list that contains a breadcrumb trail of the business process artifacts that you currently have open. Explore the options in the list to discover which artifacts are currently active.

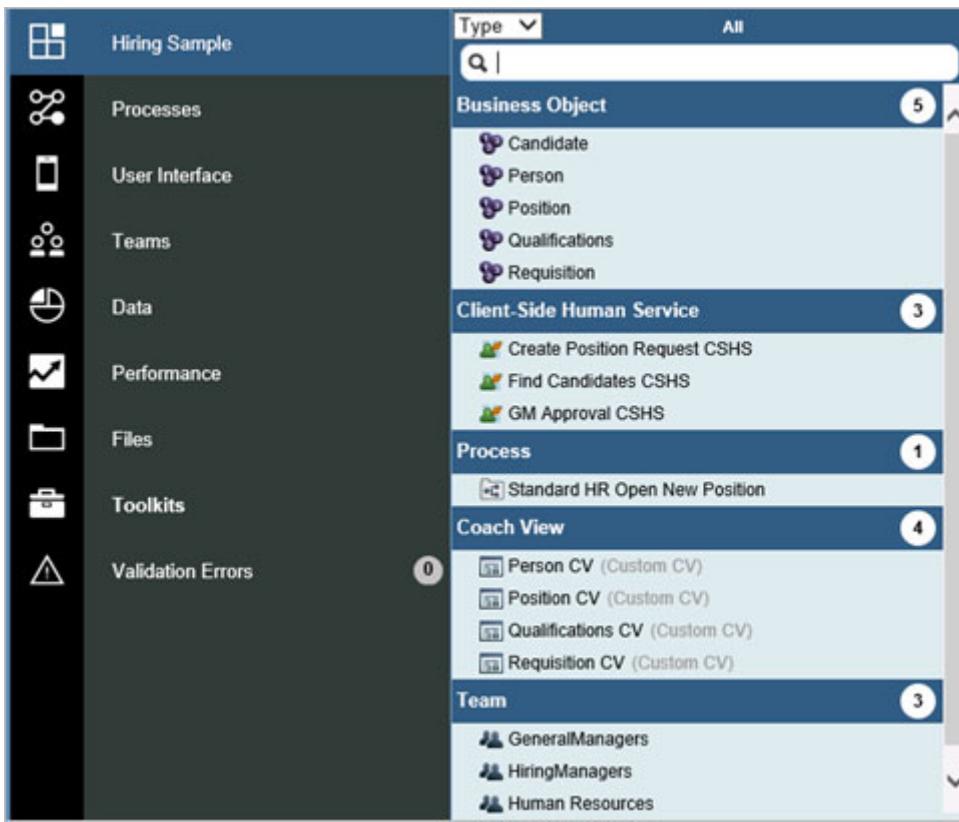


Hint

To show the type of artifact, use the icon to the left of the name of the artifact. If an artifact is changed, but not saved, it is indicated with an asterisk. To switch to a specific artifact, select it from this list. To close the artifact that is open in the editor, click X.

- g. Click **X** to close the **Create Position Request CSHS** service. If a prompt appears to save any changes, click **No**. You return to the business process definition.

3. Examine all the artifacts of the business process.
- a. In the left pane of the design window, click **Hiring Sample**. All the artifacts that are used to create this process application are displayed.



- b. The artifacts of the process application are subdivided into categories. For example, **Person**, **Position**, and **Requisition** items are classified as **Business Objects**. Take a moment to explore all the components that make up the **Hiring Sample** process application.



Note

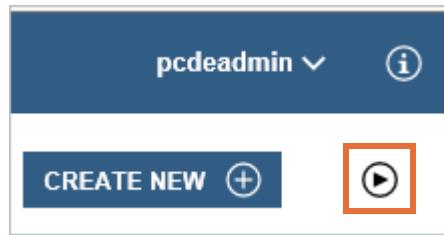
If you hover your mouse over the tools in the left pane, a plus sign is displayed to the right of each tool. To add an item to the business process, select this plus sign. To view the current contents of the process application without adding an element, double-click the element name.

4. Click anywhere in a blank space to hide the artifacts.

Part 4: Use the Playback feature to examine a running business process in IBM Process Designer Inspector

In this part of the exercise, you run the process application in the IBM Process Designer Inspector to trace the process from end to end.

- 1. In the upper-right corner of the design window, click **Run**.



- 2. The Inspector view gains focus. The business process stops at the first activity: **Submit position request**. The location of the execution can be determined in several ways. The current state of execution is shown under **Locations**.

A screenshot of the IBM Process Designer Inspector interface. At the top, it displays the title 'Standard Employee Requisition for (Standard HR Open New Position)' with a play icon. Below the title is a toolbar with icons for refresh, back, forward, and search. The main area has a dark header bar with a 'Details' section expanded, showing process details like Instance ID: 107, Process App: Hiring Sample, Status: Active, Start time: Mar 16, 2016 3:25 PM (2 minutes ago), Last action: Mar 16, 2016 3:26 PM (1 minute ago), and Due date: Mar 16, 2016 7:25 PM (in 4 hours). Below this is a 'Tasks (1)' section with one item: 'Submit position request [Token #41]'. This task is highlighted with a red rectangular box. A tooltip below the task says 'The task is assigned to pcdeadmin (pcdeadmin). It is due in 4 hours.' At the bottom, there is a 'Data' section. The entire screenshot is framed by a red border.

On the diagram, the **Submit position request** activity is marked with a token.



- 3. Complete the **Submit position request** activity.
 - a. Under **Locations**, click the **Start** arrow icon to run the task.

Screenshot of the IBM BPM Worklist interface for a process named "Standard Employee Requisition for (Standard HR Open New Position)".

The interface includes:

- Header: Standard Employee Requisition for (Standard HR Open New Position) with a "Tip" link.
- Toolbar: Includes icons for refresh, search, filter, and other process controls.
- Details section:
 - Instance ID: 107
 - Process App: Hiring Sample
 - Status: Active
 - Start time: Mar 16, 2016 3:25 PM (2 minutes ago)
 - Last action: Mar 16, 2016 3:26 PM (1 minute ago)
 - Due date: Mar 16, 2016 7:25 PM (in 4 hours)
- Tasks section: Shows 1 task.
- Locations section:
 - Submit position request [Token #41] (highlighted with a red box)
 - Submit position request (with a start arrow icon)

The task is assigned to pcdeadmin (pcdeadmin). It is due in 4 hours.
- Data section.

- ___ b. The browser opens a new empty window at first. Do not close it. It takes few minutes before the coach is displayed for the first time in this window.

Request data		Position data	
* Employment type	Contract	* Position type	New
* Department	Finance	* Location	Chicago
Hiring manager	Tom Miller	Title	Head of Product Development

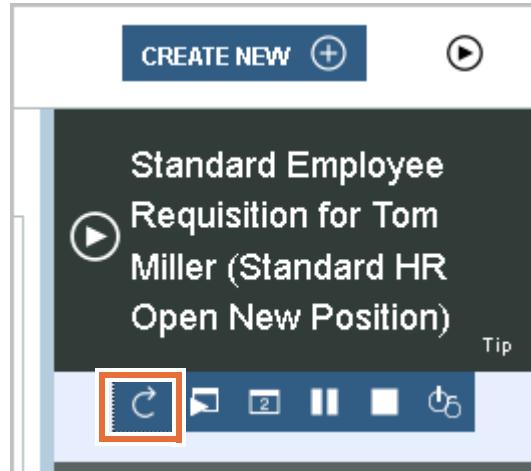
Part 5: Use a coach to work with a running business process

The **Submit position request** subtask is a series of GUI coaches that are intended to prompt the user for input to proceed to the next step. When you click the **Run** icon, a browser window opens and starts the first coach in the flow. The coach might take a few minutes to display completely.

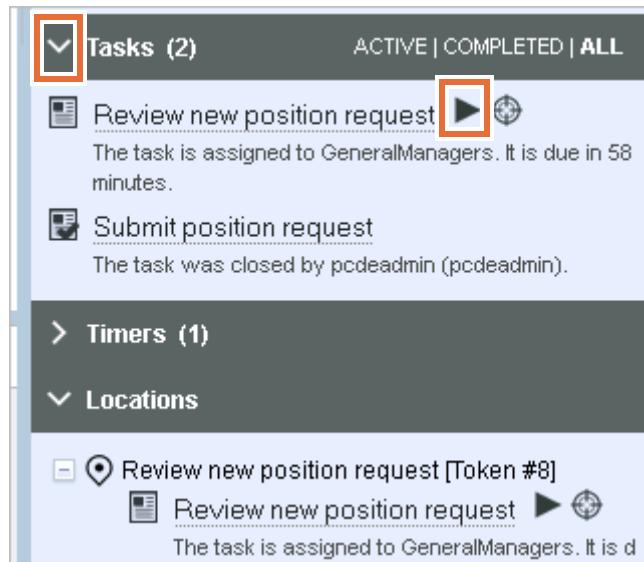
- ___ 1. In the **Position Request HS** coach, set the following values:
 - Employment Type: Full-time
 - Department: Customer Service
 - Planned starting date: Future date (for example Tomorrow's date)
 - Location: Boston
 - Position type: New
- ___ 2. Scroll down and click **Next**.
- ___ 3. Remember from exploration of the subflow that because this position is new, when the position request is created, you must confirm the position details and submit them to end this subflow. A confirmation page is displayed. Click **Submit**.

The subflow ends and displays “**The service has finished**” in the coach window. Close this browser window.

- 4. Return to the Inspector view in IBM Process Designer and click Refresh.



- 5. The status is automatically updated. Expand **Tasks**. The status of the submit requisition task is now closed, and the location moves to the next activity in the flow: **Review new position request**. Click the **Start** arrow icon to run the task.



- 6. The task starts in a new browser window. The coach might take a few minutes to display completely.

- 7. This time, the coach provides you with the option of either approving or rejecting the requisition. Select **Approved** and click **Submit**.



- 8. Close the browser window to return to IBM Process Designer. Because the requisition is approved, the flow moves to the **Find position candidates** activity.
- 9. Click the Start arrow icon to run the task. The Review Position coach opens in a new window.
- 10. Click **Add Candidates**. The coach then displays a list of candidates.
- 11. Click **Submit**. The service finishes. Close the browser window and return to the Process Designer window.
- 12. The entire process is now complete. Expand **Tasks** to view the completed tasks. The tasks that are listed correspond with the tasks in the process definition.



- 13. You can run the process again and select different options to examine the different execution states of the business process. Test the following examples:
- If the **Submit position request** activity requests an existing position, what happens?
 - If it is an existing position, do you need approval from the general manager?
 - If the general manager rejects the requisition, then what happens?
 - What is the result of the Find job candidates activity?

- 14. When you complete exploring the **Hiring Sample** process application, click the down arrow next to the login name and log out of the web-based Process Designer. Close the browser.



Part 6: Wire activities together in a business process

In this part of the exercise, you examine a partially completed process application and wire activities together to form a completed business process. Because this business process application is a skeleton, it does not contain completed implementation activities.

For the rest of this course, you use the desktop version of the Process Designer.



Note

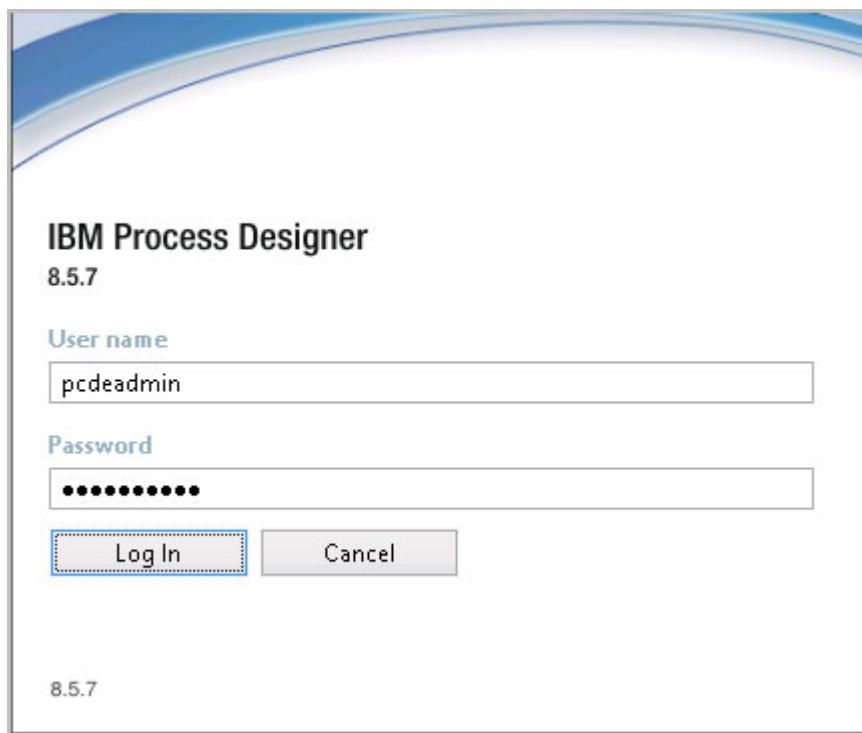
For IBM BPM on Cloud users

To obtain and install the desktop version of Process Designer, you must log in to your IBM BPM Cloud instance. At the bottom of the Process Center tile, you find a link to download Process Designer. The version that you download from IBM BPM on cloud is already configured to communicate with your IBM BPM on Cloud Process Center.

-
- 1. Start the desktop IBM Process Designer. Double-click the **Start IBM Process Designer** icon on the desktop.



2. Enter `pcdeadmin` in the **User name** field and `web1sphere` in the **Password** field, and click **Log In**.



- ___ 3. When the **Security Alert** dialog box is displayed, click **Yes** to proceed.



- ___ a. After a few moments, click **Yes** one more time.

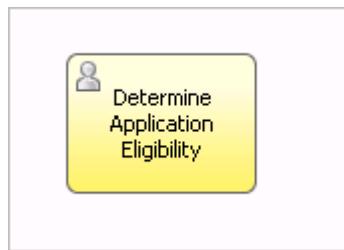


- ___ 4. In the Process Center perspective, click the **Process Apps** tab.
___ 5. Click the **Open in Designer** link that is associated with the **Account Verification Skeleton (AVS)** process application.

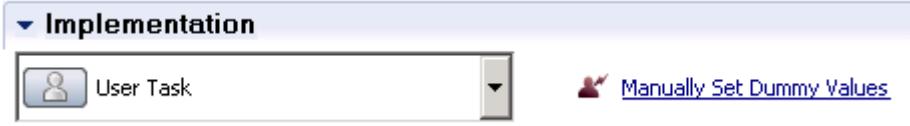


- ___ 6. Select **Processes** and double-click **AccountVerification** to open the process application in the editor.

A business process opens in the design view. You can see from the activities in the process that the process is incomplete. Several activities, such as **Determine Application Eligibility**, are not wired to the rest of the diagram.



- 7. Examine the implementation of **Determine Application Eligibility**.
 - a. Select the **Determine Application Eligibility** activity from the design editor.
 - b. At the bottom of the editor, make sure that the **Properties** tab is selected.
 - c. Click the **Implementation** option from the properties list.
 - d. A human task that is named `Manually Set Dummy Values` implements the activity.



Click the link to view the implementation of this activity.

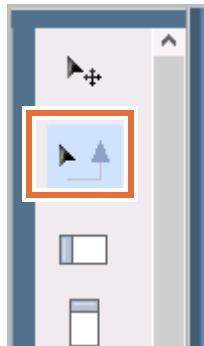
- e. The **Manually Set Dummy Values** service does only one action: to start a coach that allows the user to enter some dummy values. If you want to examine that coach in the **Coaches** tab, you can choose to do so.
- f. When you are done examining the **Manually Set Dummy Values** service, use the **X** in title bar to close it. You now return to the business process application.
- 8. Follow the process narrative to correctly wire the activities into the process.
 - a. A process narrative is an optional script that describes, in text form, how the process is expected to flow.

Process narrative for account verification

- When the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and end the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is done against the customer's credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.

- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.
- If the customer is determined to be high risk, the customer must submit more documentation. Then, the customer must seek final approval for the application.
- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

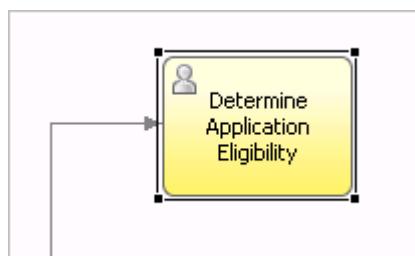
___ b. Click the Sequence Flow icon from the toolset at the right of the design view.



___ c. With the Sequence Flow selected, hover your mouse next to the **Start** element. Connection points are visible as points along the **Start** element.



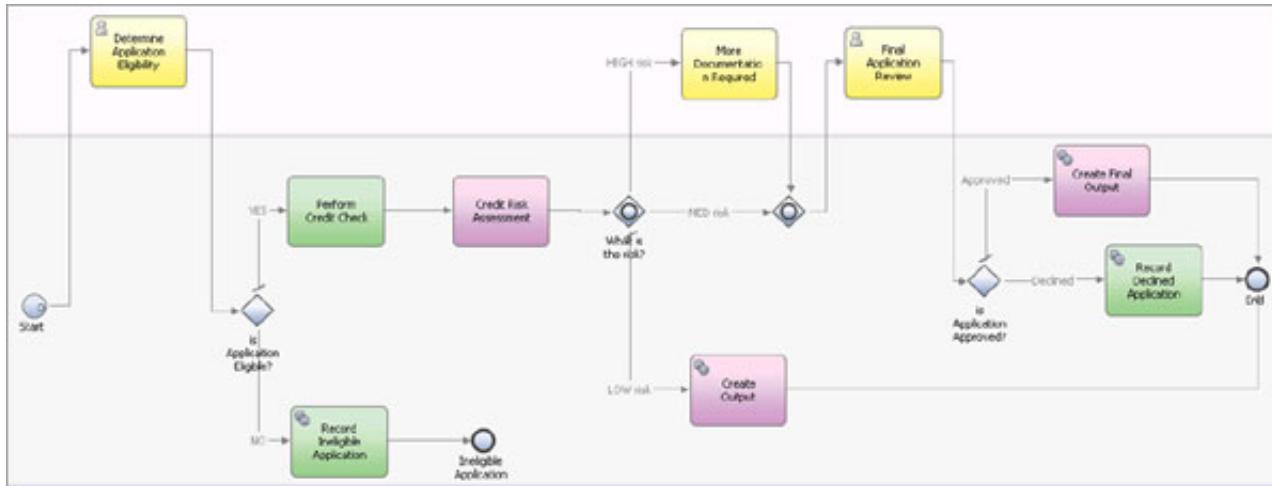
___ d. Select one of the connection points on the **Start** element. Drag your cursor to attach the wire to a connection point on the **Determine Application Eligibility** activity.



___ e. Wire the remaining activities together as suggested by the process narrative. Consider wiring the following activities and decisions:

- Determine Application Eligibility
- Is Application Eligible?
- Credit Risk Assessment
- Create Output

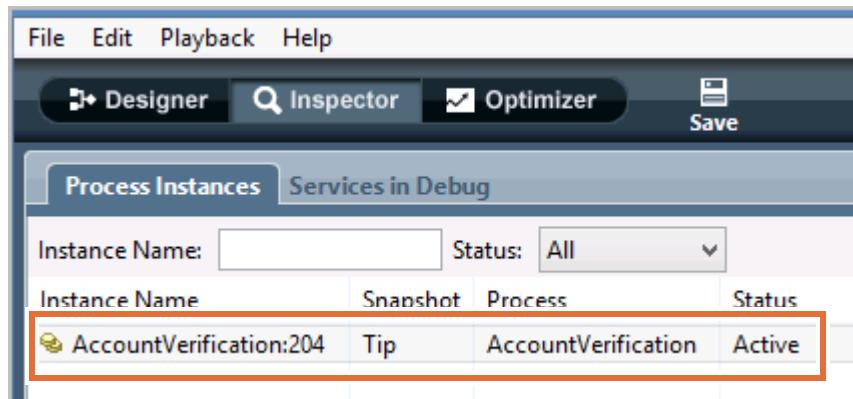
- Final Application Review
 - Create Final Output
 - Record Declined Application
- 9. Compare your solution to the following diagram. Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



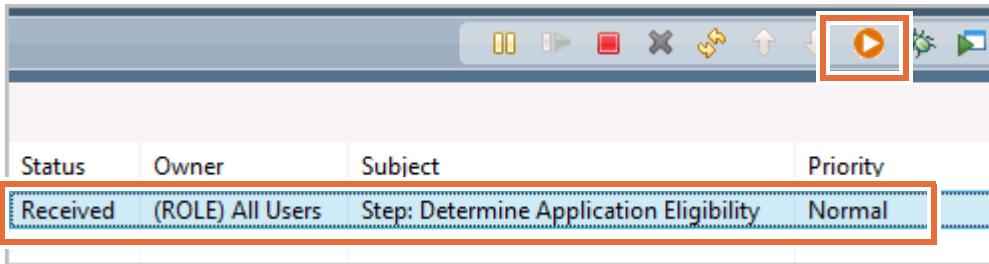
- 10. When you are done wiring the activities and decisions into the process application, click the **Save** icon to save your diagram.



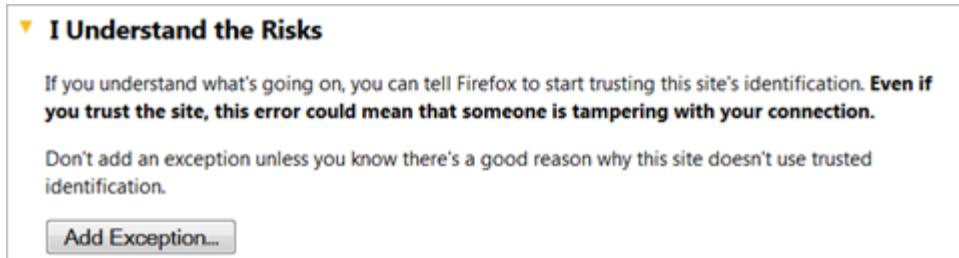
- 11. Test your process application by running it in the Inspector.
- Click the **Run Process** icon. You are asked for permission to switch to the Inspector view. Click **Yes**. The Inspector view opens.
 - The process application stops at **Determine Application Eligibility**. Remember that this activity is implemented as a human task to gather dummy data. If you do not see the **Determine Application Eligibility** activity listed, then click the AccountVerification process instance on the left.



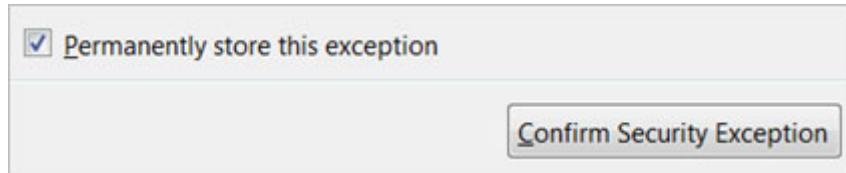
- ___ c. Select the **Determine Application Eligibility** state from the upper right and click **Runs the selected task**.



- ___ d. If you are prompted to select a user, select **pcdeadmin** and click **OK**.
___ e. The task starts in a browser window. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.



- ___ f. On the next screen, accept the defaults and click **Confirm Security Exception**.



- ___ g. The coach for collecting dummy data is displayed in a browser window. This display might take a few minutes. Set the following values for the coach and click **OK** after entering your values.
- Eligible Application: Enabled
 - Credit Risk: Low
 - Application Decision: Enabled



The service finishes. Close the browser window.

- ___ h. Return to IBM Process Designer and the Inspector view. Click the refresh icon to refresh the view.
- ___ i. The instance is complete.

Process Instances		Services in Debug			
Instance Name:	Status:	Snapshot	Process	Status	Due Date
AccountVerification:205	Tip	AccountVerification	Completed	Apr 25, 20...	205

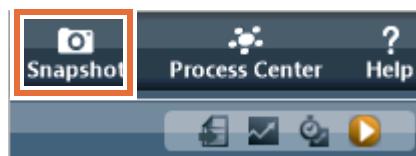
Remember that, other than the dummy values coach in the **Determine Application Eligibility** activity, none of the other activities have any implementations. If the process application is wired together properly, it completes without any errors.

12. Save your solution to a snapshot.

- ___ a. Click the Designer view from the upper left.



- ___ b. Click the Snapshot tool from the upper right.



- ___ c. Enter a meaningful name for your snapshot, such as: Skeleton Rewired
- ___ d. Enter a description for the snapshot and click **OK**.



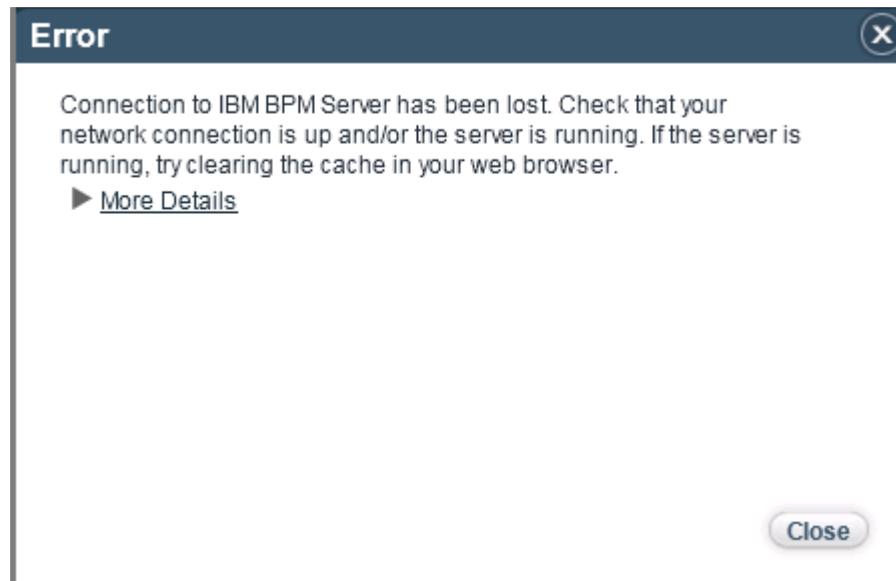
Information

A snapshot captures a precise moment in time for the process application. Snapshots allow auditing of previous process applications and reverting to a previous state. Snapshots can be managed from IBM Process Center or with the available Process Center tools in IBM Process Designer. Only snapshots can be deployed to the server.



Important

At any time while working with the Process Designer or Process Center, you might receive an IBM Business Process Manager error with a message that the connection was lost. If that occurs, do not be concerned. Click **Close** and continue with your lab exercise.



- ___ 13. To work with the snapshots of the process application, return to the Process Center perspective by clicking the Process Center icon from the upper-right corner.
- ___ 14. Click the **Account Verification Skeleton (AVS)** process application. A list of snapshots is displayed.

If you do not see the process applications in the repository view, select the **Process Apps** tab, right-click the workspace, and click **Refresh**.

Process Application	Last Changed	Created By	Status
Current	5/29/14	pdeadmin	Not Yet Installed to Process Server
Skeleton Rewired (SR)	6/16/14	pdeadmin	Not Yet Installed to Process Server
Lab 1 Start (L1S)	5/29/14	pdeadmin	Not Yet Installed to Process Server

- ___ 15. This section of the exercise is now complete. Switch back to the Process Apps tab by clicking the **Process Apps** link for the next section.

Part 7: Examine the Hiring Sample Advanced BPD and the associated users

In this portion of the exercise, you start the IBM Process Portal and explore the capabilities of the tool. In addition, if you have the associated permission, you can use dashboards to view the performance of individuals, teams, and processes.

For people to collaborate when working with business processes and required manual tasks, users and user groups must be added to the system. The user groups are assigned to the users, which again are assigned to the lanes in the business process definition (BPD) modeled in IBM Process Designer. This task is already done in this exercise for you.

- ___ 1. Start the BPD.
- ___ a. Verify that you are in the Process Center perspective and that **Process Apps** is the active tab.
- ___ b. Locate the **Hiring Sample Advanced (HSAV1)** process application but do not click it.

- __ c. On the right of **Hiring Sample Advanced (HSAV1)**, click **Open in Designer**.

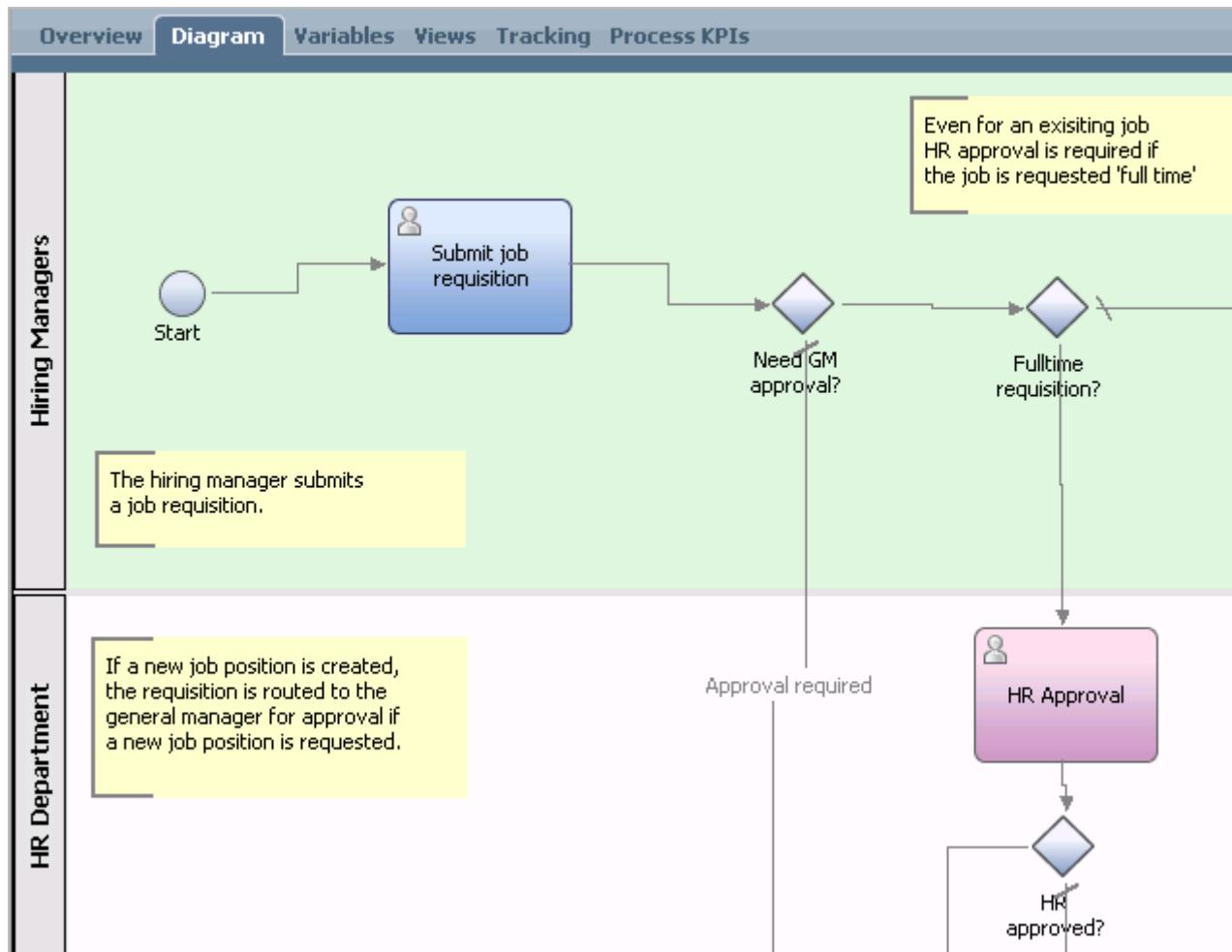
The screenshot shows the 'Process Apps' interface. At the top, there are tabs for 'Process Apps', 'Toolkits', 'Servers', and 'Admin'. Below the tabs, there is a search bar with 'Sort By: Recently Updated' and buttons for 'All' and 'Favorites'. A list of sample applications is displayed:

- Hiring Sample (HSS) ☆ ? Open in Designer
- Hiring Sample Advanced (HSAV1) ☆ ? Open in Designer
- Account Verification Skeleton (AVS) ☆ ? Open in Designer
- Procurement Sample (STPPS1) ☆ Open in Designer
- Process Portal (SYSRP) ☆ ? Open in Designer
- Heritage Process Portal (TWP) ☆ Open in Designer

- __ d. Select **Processes** and double-click **Open New Position** to open the business process application in the editor.

The screenshot shows the 'Process App Settings' interface for the 'HIRING SAMPLE ADVANCED' application. The left sidebar has categories: All, Processes (which is selected and highlighted in green), User Interface, and Implementation. The main area shows 'Business Process Definitions' with a count of 1. A list of items is shown, with the last item, 'Open New Position', highlighted with a red box.

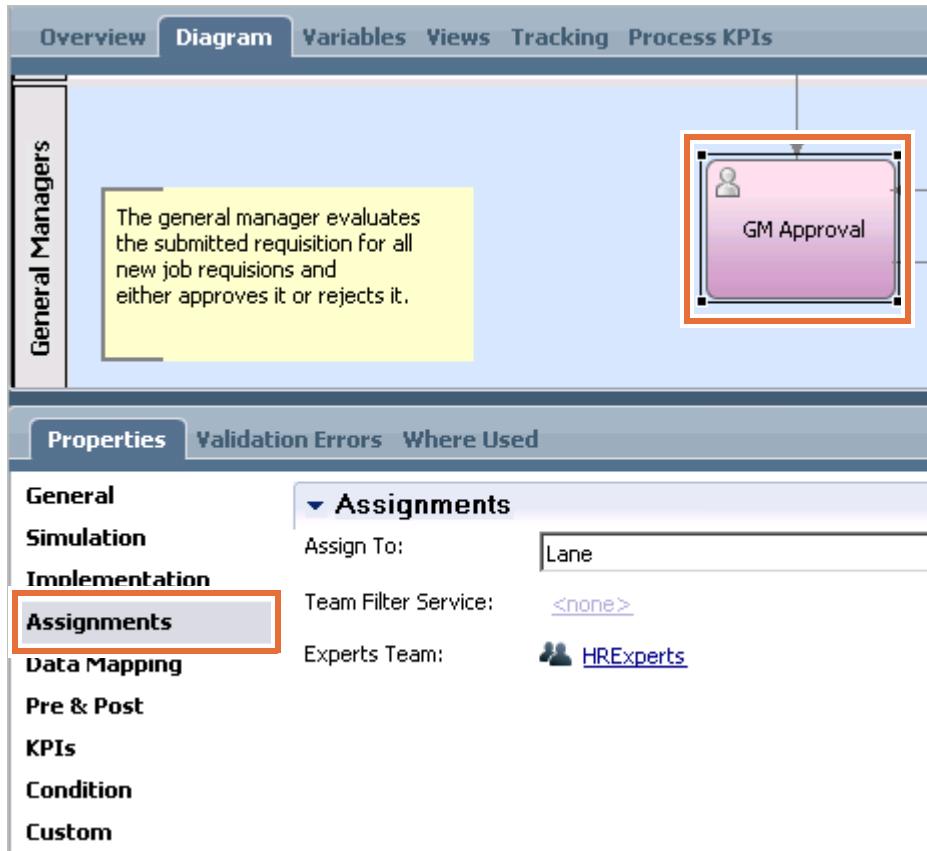
- e. The business process application opens in IBM Process Designer. Double-click the **Diagram** tab to maximize the process view. After reviewing it, double-click **Diagram** again to restore the size.



This process is an enhanced version of the process you worked with earlier in the previous section. Namely, an extra **HR Approval** task is added. This task is required if the job requisition is done for a full-time requisition by the HR department. This task is added to allow for more collaboration among the users.

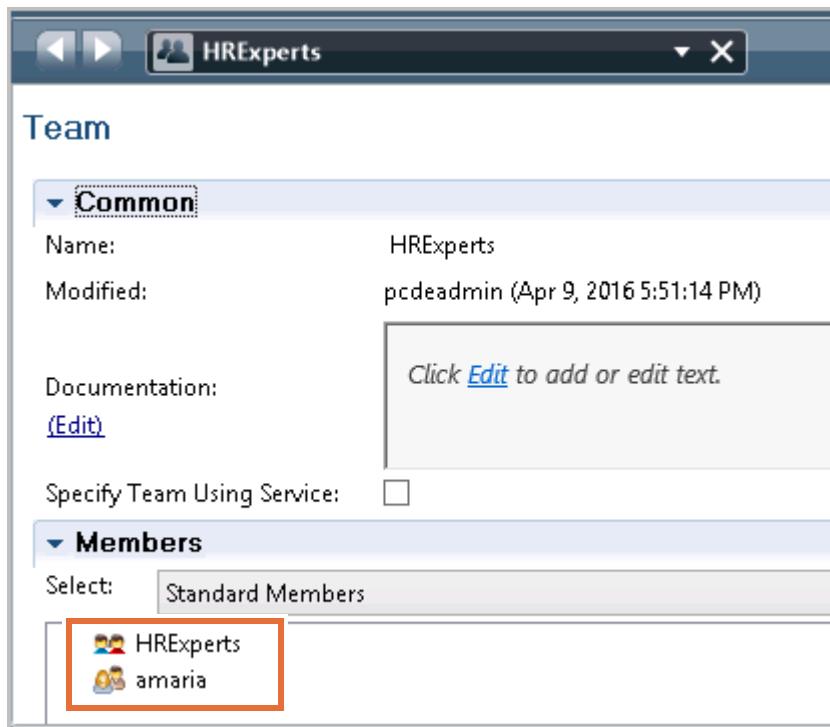
- 2. Review users who are assigned to tasks.
— a. Select the **GM Approval** task.

- __ b. In the **Properties** tab, click **Assignments**.

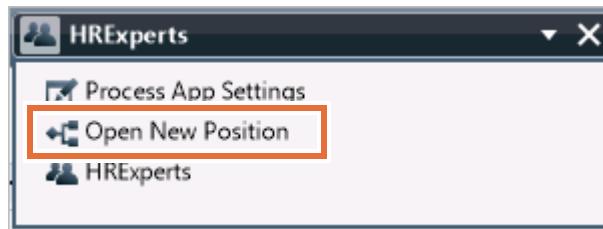


- __ c. Click **HRExperts** to view the details of the group.

- ___ d. Examine the **Team** view. The user **Maria** (amaria) belongs to the **HRExperts** group. This user is specified to be an expert on the **GM Approval** task.



- ___ e. Select **Open New Position** from the list at the top of the screen.



- ___ f. Click the **Overview** tab.



- ___ g. In the **Exposing** section, click the **HiringManagers** group.



- ___ h. Examine the members in the **Team** view. The **HiringManagers** group has three users: ajane, ajohn, and ajack. Only these members are allowed to create an instance of the process as specified in this BPD.

- ___ i. Click the Process Center icon at the upper right to switch to the Process Center view.

Part 8: Examine the BPD and start an instance

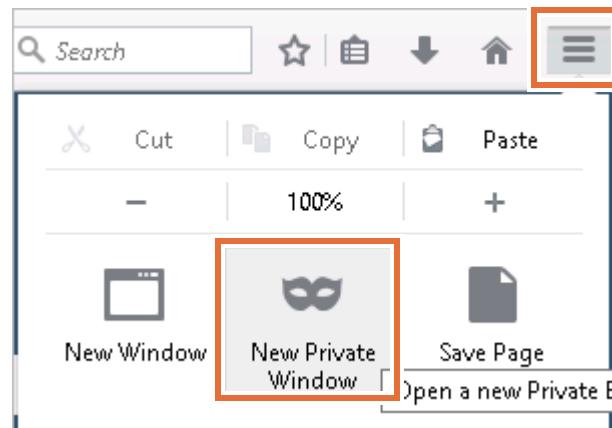


Note

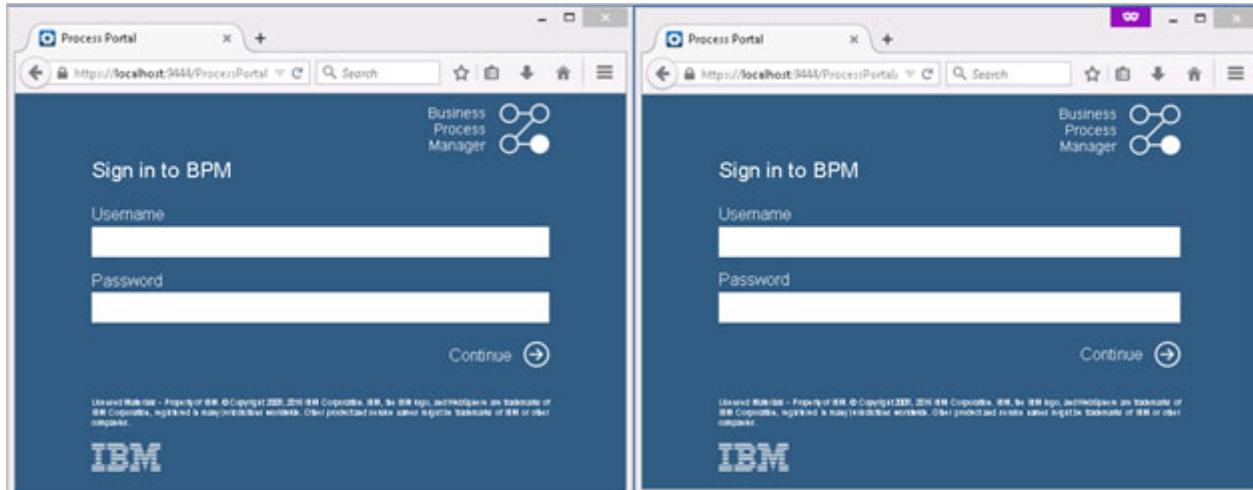
For IBM BPM on Cloud users

Click Launch in the Process Portal tile of your IBM BPM on Cloud home screen to access the Process Portal.

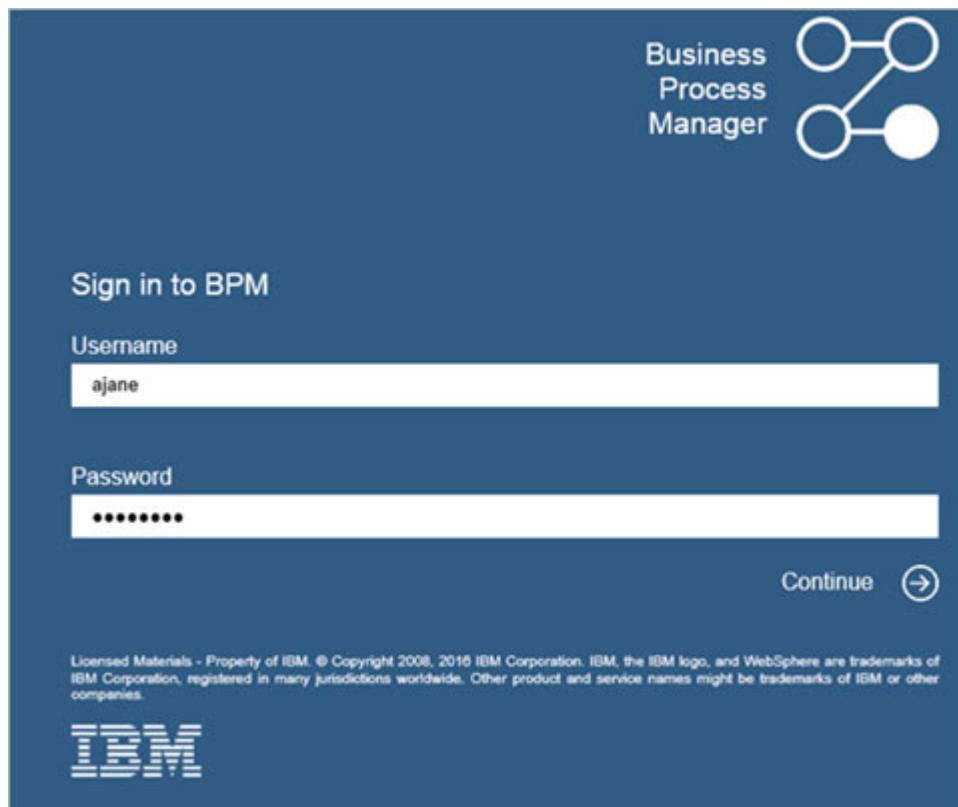
- ___ 1. Start the Process Portal.
 - ___ a. Start **Firefox** and enter `http://localhost:9081/portal` in the URL field.
 - ___ b. If prompted, add an exception and confirm the security exception. **Do not log in to the portal at this moment.**
 - ___ c. Open a new browser window by clicking **Open menu** and then **New Private Window** as shown.



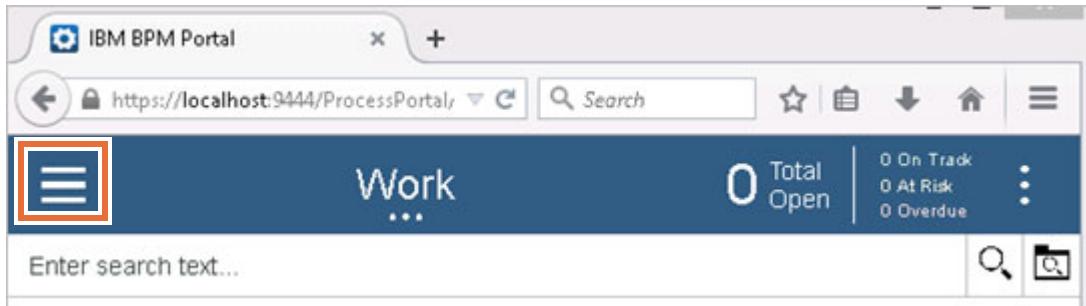
- ___ d. Enter `https://localhost:9444/ProcessPortal` in the **URL** field of the new instance of the browser and press the Enter key. **Do not log in to the portal at this moment.**
- ___ e. Rearrange the open windows so that you can see both at the same time. One method for arranging the windows is shown here; do not be concerned about the text.



- ___ f. Enter **Username** ajane and **Password** password in left window. Click **Continue**.



- __ g. The Work page opens, with no current work items. Click **Main Menu** to display the user dashboard.

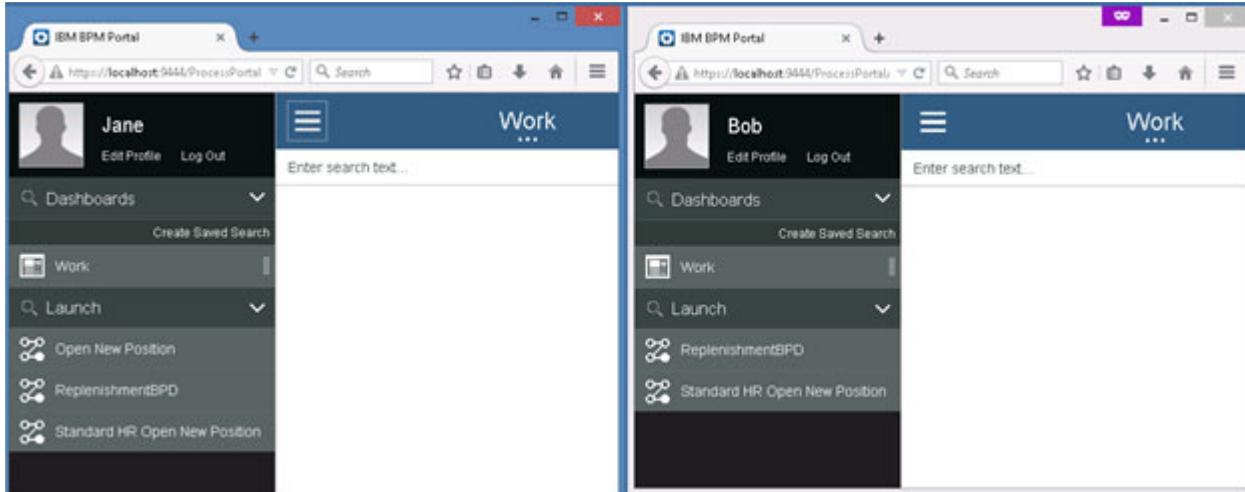


- __ h. Enter **Username** abob and **Password** password in the right window. Click **Continue**. The Work page opens, with no current work items. Click **Main Menu** to display the user dashboard.

 **Note**

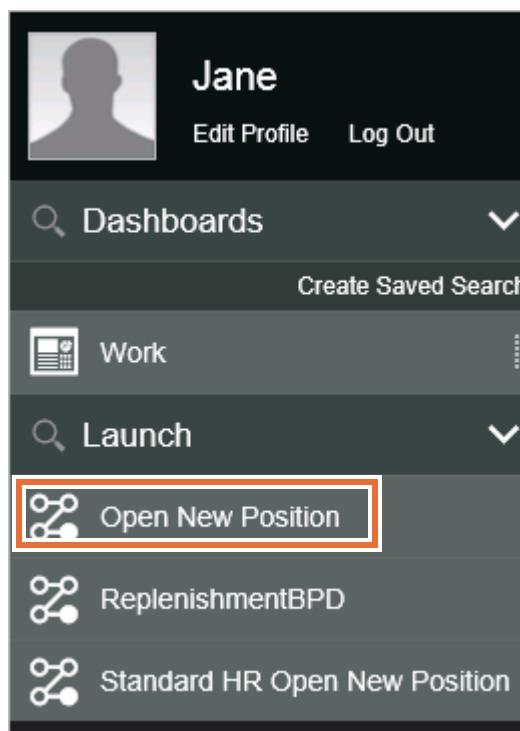
Firefox might duplicate logins when the new windows are not created before any user logs in. You enter data in two separate browser windows as two different users. Feel free to maximize and restore the size of the windows as necessary.

- __ i. You now have two open Process Portals, one for each user.



No tasks appear in the work lists of the users. **Jane** is the only user who can create an instance of the process. This action is specified in the BPD that you earlier previewed in the IBM Process Designer. The **Open New Position** task is listed in the Process Portal for the user **Jane**.

2. Use the Process Portal to start a new job requisition.
- a. In the Process Portal for user **Jane**, click **Open New Position**.



- b. If the Submit requisition task is displayed, then click it to claim it.



- ___ c. Enter the following data in the Submit Requisition task. Note: Sometimes this task activity takes a few minutes to display, but do not be concerned as this delay is normal. As soon as the display is complete, you can scroll through the windows to view the fields. Because both sessions might be arranged on your desktop, you can maximize the size of the window to enter data and then restore the size back to its original arranged location.
- Requestor: Henry Miller
 - Employment status: Contract
 - Department: Product Development
 - Location: Minneapolis
 - Position type: New
 - Job title: Software Architect

Submit job requisition

Job requisition data			Position data
Requester	Requested job position	Requested job start date and location	
Requisition number 5067125469	Employment status Contract	Planned date of job start 3/17/2016	Position type New
Requestor Henry Miller	Department Product Development	Location Minneapolis	Job title Software Architect
	Number of employees required 1		

Make your decision

Next

- ___ d. Keep the other fields at the default values and click **Next**. To locate **Next**, you can scroll to the left of the window.
- ___ e. The next coach of the first human task is presented to confirm information about the job requisition.

Submit job requisition

Job requisition data			Position data
Requester	Requested job position	Requested job start date and location	
Requisition number 5067125469	Employment status Contract	Planned date of job start 3/17/2016	Position type New
Requestor Henry Miller	Department Product Development	Location Minneapolis	Job title Software Architect
	Number of employees required 1		

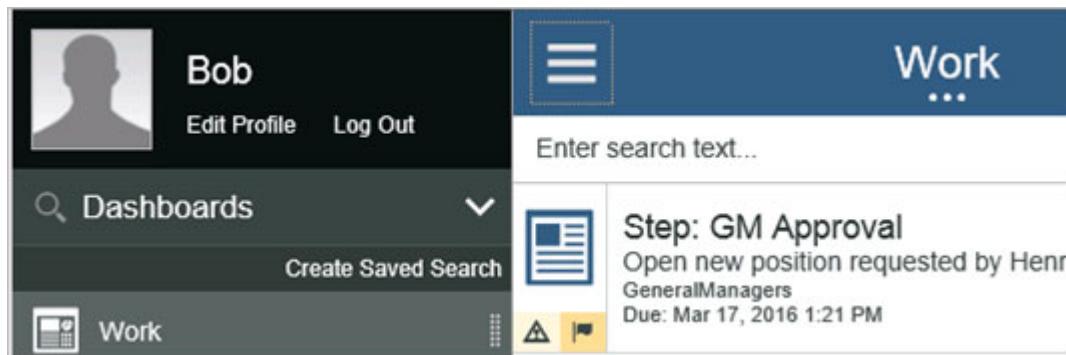
Make your decision

Back **Submit**

- ___ f. Click **Submit**. This action completes the task, and the process engine goes to the next task in the BPD. The next task in the process is the **GM Approval** task, which **Bob** is going to work on.
- ___ 3. Work with the task list for Bob.
 - ___ a. Examine the Process Portal for user **Bob**. If **GM Approval** task is not listed, then click **Work** to refresh the screen.



- ___ b. Confirm that the new **GM Approval** task is listed here.



- ___ c. Click the **Step: GM Approval** task.
- ___ d. If a window is displayed to claim the task, click **Claim Task** to work on it.
- ___ e. The task opens in the Process Portal.

 A screenshot of the "Step: GM Approval" task window. It has several sections:
 - Top: "Step: GM Approval" title and a "Position data" panel with "Position type: New" and "Job title: Software Architect".
 - Middle-left: "Job requisition data" section with three sub-sections: "Requester" (Requisition number: 5067125469, Requestor: Henry Miller), "Requested job position" (Employment status: Contract, Department: Product Development, Number of employees required: 1), and "Requested job start date and location" (Planned date of job start: 3/17/2016, Location: Minneapolis).
 - Middle-right: "Position data" panel with "Position type: New" and "Job title: Software Architect".
 - Bottom: "Make your decision" section with a note about contacting HR experts, radio buttons for "Approved?" (Approved is selected), a "Comments" input field, and an "OK" button.

- ___ f. Click the **menu** (three vertical dots) on the right to see more information about the current state of this task.

The screenshot shows a task details interface. At the top, there is a navigation bar with three tabs: 'Details' (highlighted with a red box), 'Experts', and 'Stream'. To the left of the main content area, there is a vertical sidebar with a blue header and a white body. In the main content area, there is a yellow star icon followed by the text 'Open new position requested by Henry Miller(153)'. Below this, the 'Due Date' is listed as 'Apr 4, 2016 6:27 PM'. Underneath the due date, several metadata fields are displayed: 'Department: Product Development', 'Employment Status: Contract', 'Hiring Manager: Henry Miller', and 'Location: Minneapolis'. A section titled 'Tasks' follows, containing two entries. The first entry is a completed task: 'Task: Submit requisition' with a checkmark icon, associated with user 'ajane' (represented by a small profile picture). The second entry is a pending task: 'Step: GM Approval' with a green triangle icon, currently unassigned (represented by a small square icon). Both tasks have their 'Created' and 'Completed' dates listed.

- ___ g. Click **Experts**. Maria appears in the list of available experts.

The screenshot shows the 'Experts' tab selected in the navigation bar. The main content area displays two sections: 'Experienced Users' and 'Subject Matter Experts'. The 'Experienced Users' section contains the message 'No one in the system has completed a similar task. For more assistance, contact the system administrator.' The 'Subject Matter Experts' section lists one expert: 'Maria'. Her profile picture is shown, along with her name and completion statistics: 'Similar Tasks Completed: 0' and 'Average Completion Time: 0 hours'. There is also a link labeled 'Mention'.

- ___ h. Click the menu (three vertical dots) to close the pane.
- ___ i. Click **Approved**. If you want to enter a comment, you can do so. Click **OK**. The task clears from Bob's work list and appears for Jane.

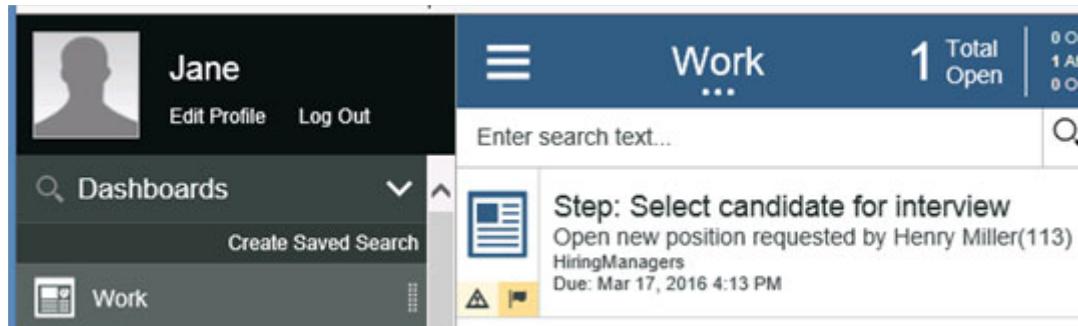
If you're unsure about your approval please contact your HR expert.

Approved Rejected

Comments
We need this help.

OK

- ___ 4. Complete the process.
- ___ a. The next step, named **Select candidate for interview**, is ready for Jane, the Hiring Manager. Click **Main Menu** again to verify that the task belongs to Jane. If the task is still not listed, then refresh Jane's browser window.



- ___ b. Click the task and claim it. The coach of the task is presented.

Last name	First name	Supervisor	Pay level	Pay type	Comments
DeFiori	Iselhar	Isidor Himmelblau	3	Exempt	Iselhar's last ratings had been fantastic.
Sadmind	Elly	Hans-Josef Schmidhweiler	6	Exempt	Elly is not an easy person. Expect major discussions especially about work/life balance.
Krachmanikovo	Uwe	Freddie T-Bone	7	Nonexempt	Uwe came back from Australia and is looking for a new job.
Truly	Zebastian	Bert Humpe	5	Exempt	Zab is very experienced, but will retire next summer.
Szczypniak	Olivera	Sabine Reibelsen	9	Nonexempt	Oli finished her master thesis two months ago and wants to get started with serious work.

- ___ c. Jane can open the menu to see the execution history of the job requisition. For example, Jane can see that Bob worked on the **GM Approval** task.

The screenshot shows a process portal interface. At the top, there are three tabs: 'Details' (selected), 'Experts', and 'Stream'. Below the tabs, a yellow star icon indicates an 'Open new position requested by Henry Miller(153)'. The 'Due Date' is listed as 'Apr 4, 2016 6:49 PM'. Below this, the 'Department' is 'Product Development', 'Employment Status' is 'Contract', 'Hiring Manager' is 'Henry Miller', and 'Location' is 'Minneapolis'. A section titled 'Tasks' lists four items:

- ✓ Task: Submit requisition by ajane. Created: Apr 4, 2016 5:24 PM. Completed: Apr 4, 2016 5:27 PM.
- ✓ Step: GM Approval by abob. Created: Apr 4, 2016 5:27 PM. Completed: Apr 4, 2016 5:49 PM.
- ✓ Step: SimCandidatesList by pcdeadmin. Created: Apr 4, 2016 5:49 PM. Completed: Apr 4, 2016 5:49 PM.
- ▶ Step: Select candidate for interview (status not yet completed).

- ___ d. It is not possible to make any selection from the list of candidates for this test. Click **Invite selected candidate to interview**. This step completes the task. The work lists for both Bob and Jane are empty.
- ___ e. If time permits, feel free to explore the other areas of the Process Portal. When you are done, close all the open Process Portal windows.
- ___ 5. Close all open browsers.
- ___ 6. Close IBM Process Designer.

7. Stop the IBM Process Center cell environment.
- __ a. Double-click the **Stop Process Center cluster** icon. Wait until it stops successfully. Press any key to close the command window.
 - __ b. Double-click the **Stop Process Center node agent** icon. Wait until it stops successfully. Press any key to close the command window.
 - __ c. Double-click the **Stop Process Center deployment manager** icon. Wait until it stops successfully. Press any key to close the command window.



Hint

You can also stop IBM Process Center from a DOS command window.

- 1) If you open a DOS command window, browse to the following directory:
C:\IBM\BPM\v8.5\profiles\dmgrProfile\bin
- 2) Type the following command to stop the IBM Process Center Deployment Manager:
stopManager.bat -username bpmadmin -password web1sphere
- 3) To stop the node agent and the cluster, browse to the following directory:
C:\IBM\BPM\v8.5\profiles\node1Profile\bin
- 4) Type the following command to stop the IBM Process Center Node agent:
stopNode.bat -username bpmadmin -password web1sphere
- 5) Type the following command to stop the IBM Process Center Single Cluster:
stopServer.bat SingleClusterMember1 -username bpmadmin -password web1sphere



Information

The Business Process Designer offers many more capabilities and benefits than the ones that are shown here. Go to <http://www.ibm.com/services/learning> and search for IBM BPM Standard to learn more.

End of exercise

Exercise review and wrap-up

In this exercise, you started and explored IBM Process Designer. You also examined a running business process in IBM Process Designer Inspector. Finally, you explored the IBM Process Portal and its collaboration capabilities.

Exercise 2. Exploring IBM Integration Designer, part I

What this exercise is about

This exercise introduces the IBM Integration Designer Eclipse-based development environment. It shows you the components in business integration projects, graphical editors, documentation capabilities, and the Task Flows view. It also explores the preferences, capabilities, product help, and product update features of the tool.

What you should be able to do

After completing this exercise, you should be able to:

- Describe IBM Integration Designer capabilities and preferences
- Use the help menu to search for assistance
- Use the Installation Manager to find product updates
- Browse the Business Integration perspective and views
- Examine the modules and libraries of a business integration project
- Browse the IBM Integration Designer graphical editors
- Use the IBM Integration Designer documentation capabilities
- Use the Task Flows view

Introduction

IBM Integration Designer is a member of a family of products that are associated with a service-oriented architecture approach to business process management. It is a complete integration development environment for building applications.

To simplify and accelerate the development of applications, the IBM Integration Designer environment provides a layer of abstraction. It separates the visually presented components that you work with from the underlying implementation. The applications that are created comply with leading industry-wide standards.

Integration developers use IBM Integration Designer to assemble business solutions by using adapters, business processes, code components, and mediation flows. Services that applications invoke can be local or remote, written in various languages, and run on various operating systems.

Applications that are created in IBM Integration Designer can be deployed to servers in the deployment stage, and maintained in the IBM Process Center

repository. IBM Process Server can run any application that is created in IBM Integration Designer, including applications that contain business processes, mediations, and adapters.

IBM Integration Designer also provides built-in, interactive tools that you can use to learn basic concepts and to create sample applications. You can use the Task Flows view to learn about integration applications. You can also create a series of applications that are designed to acquaint you with mediations, SCA, and BPEL process.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Exercise instructions

Part 1: Examine IBM Integration Designer capabilities and preferences

In this portion of the exercise, you start IBM Integration Designer and explore its capabilities in an empty workspace. An Eclipse workspace is a collection of projects and other physical resources that you are currently developing in the workbench. A workspace corresponds to a directory in the file system that contains all of the various source files and resources that are used in a project. In addition, your workspace contains any personalization preferences that you set.



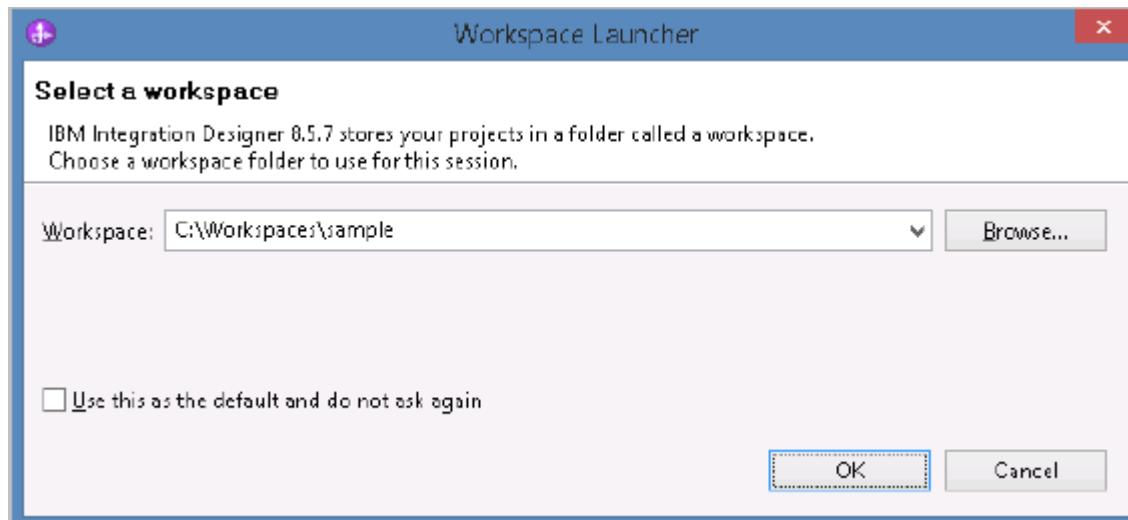
For IBM Business Process Manager on Cloud users

A version of the IBM Integration Designer that is configured to communicate with your cloud instance can be downloaded from the Process Center tile of IBM Business Process Manager on Cloud.

- 1. Start IBM Integration Designer, create a workspace, and examine the **Getting Started** tab.
- a. Double-click the **IBM Integration Designer** shortcut on your desktop.



- b. In the Workspace Launcher dialog box, change the **Workspace** location to:
C:\Workspaces\sample
Do **not** select the **Use this as the default and do not ask again** check box.



**Note**

The file path for your workspace is case-sensitive and matches the file system names. When you create a workspace in a Windows environment, ensure that your file path length does not exceed the Windows limitation of 256 characters. Failure to use short workspace file paths can lead to problems when building, deploying, or deleting your applications.

- c. Click **OK**.
- d. When the **Process Center Login** window is displayed, click **Cancel** to close the window.
- e. The Business Integration perspective is the default perspective for developing in Integration Designer. A **Getting Started** tab is displayed in the Business Integration perspective.

The **Getting Started** page includes links to help topics, samples and tutorials, basic concepts, and other resources. The **Getting Started** page is divided into the following sections:

- **What do you want to do?**

Interactive task flows teach you how to work with a process application or create a service.

- **What do you want to know?**

This section teaches you basic concepts, BPEL processes, service integration, and what is new in this version of the product.

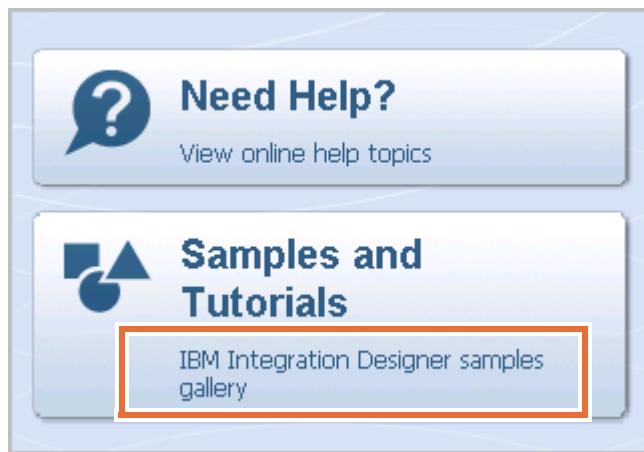
- **Other Resources**

This section provides access to developerWorks, IBM Education Assistant, support documents, and frequently asked questions (FAQ).

In addition, you have access to:

- **Help:** Online help documents
- **Samples and Tutorials:** Sample applications and tutorials that demonstrate the use of IBM Integration Designer
- **Change Environment:** Choice of development environments for working with IBM Process Server or WebSphere Data Power

- ___ f. On the **Getting Started** page, click the **IBM Integration Designer samples gallery** link.



Note the samples available. In the **More samples** section, you can click **Retrieve** to download more samples from the web (when connected).

- ___ g. Close the **Samples and Tutorials** and **Getting Started** tabs by clicking X.



When all editors are closed, a link is displayed that opens the **Getting Started** tabs for either IBM Integration Designer or IBM Process Center.

Learn more

- [Getting Started with IBM Integration Designer](#)
- [Getting Started with IBM Process Center](#)

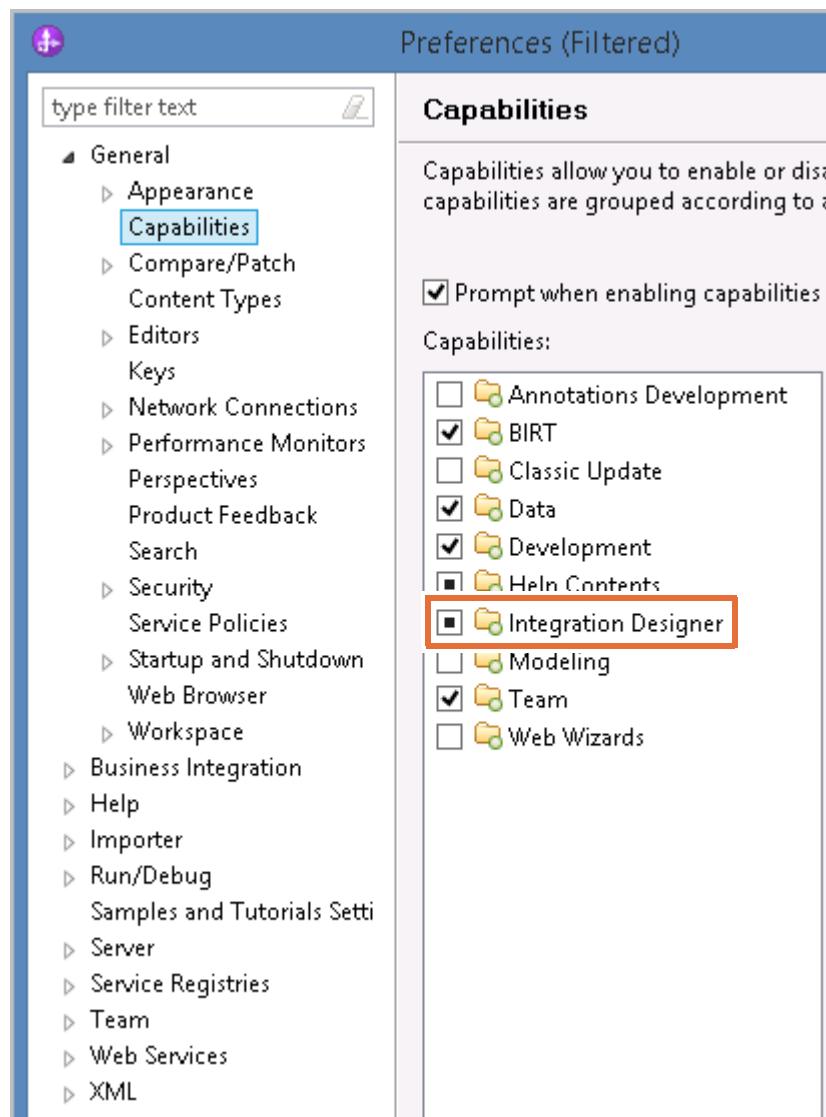
The next time that your workspace opens, the **Getting Started** tab is not displayed. You can open it by clicking **Getting Started with IBM Integration Designer** or by clicking **Help > Getting Started > IBM Integration Designer** from the menu.

Capabilities

Through Eclipse, IBM Integration Designer provides a filtering function that is known as capabilities. Capabilities are among many workspace preferences you can configure. With capabilities, you can choose to hide or show tools during application development.

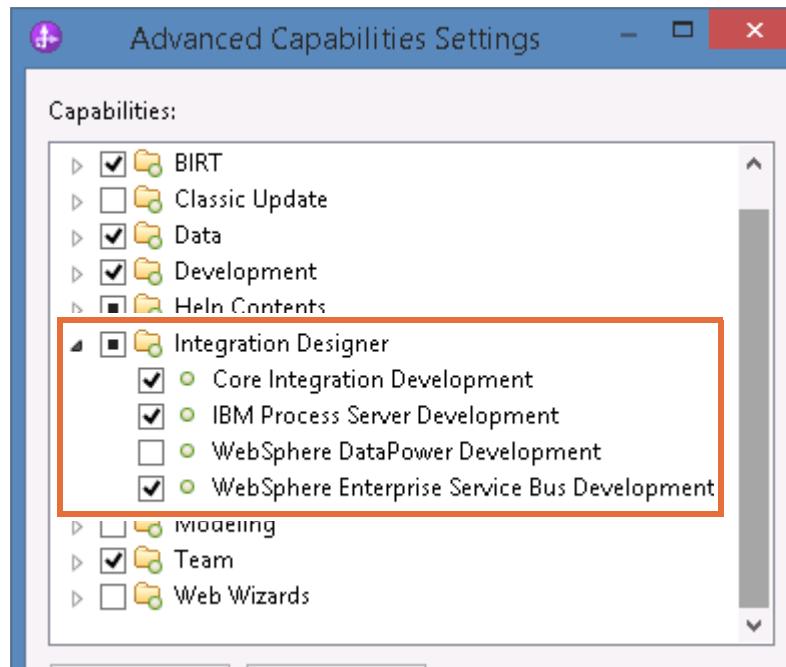
- ___ 1. From the menu options, click **Window > Preferences**.

2. View the capabilities that are currently enabled for the workbench.
- a. In the **Preferences** dialog box, expand **General** and select **Capabilities**. One category is related to the business integration tools.



- b. To see a list of the functions in this category, select the **Integration Designer** option (not the check box), and click **Advanced**.

- ___ c. In the **Advanced Capabilities Settings** dialog box, expand **Integration Designer**.



This Integration Designer category has four functions:

- **Core integration development:** Provides tools, functions, and related documentation for development of bottom-up integration of applications and services to create:
 - Libraries
 - Business objects
 - Interfaces
- **IBM Process Server development:** Includes advanced tools for top-down development of integrated business solutions to create and test components, such as:
 - Modules and module assemblies
 - Data maps
 - Customized visual snippets
 - Processes
 - Business rules
 - Human tasks
- **WebSphere DataPower development:** Includes tools, functions, and related documentation to create and test artifacts that can be deployed to WebSphere DataPower Appliances, such as:
 - DataPower libraries
 - Business objects
 - Interfaces
 - XML maps

By default, this environment is not enabled.

- **WebSphere Enterprise Service Bus development:** Includes core enterprise service bus integration development tools and also provides tools, functions, and related documentation to create and test:
 - Mediation modules and assemblies
 - Mediation flows



Note

The IBM WebSphere Enterprise Service Bus product is discontinued, and all its capabilities are incorporated into the IBM Integration Bus product.

Customers who purchase IBM Business Process Manager V8.5.7 Advanced can deploy mediation modules to the IBM Process Server runtime environment. They are not required to use a separate integration bus (enterprise service bus) environment.

- ___ d. Click **Cancel** to return to the **Preferences** dialog box.

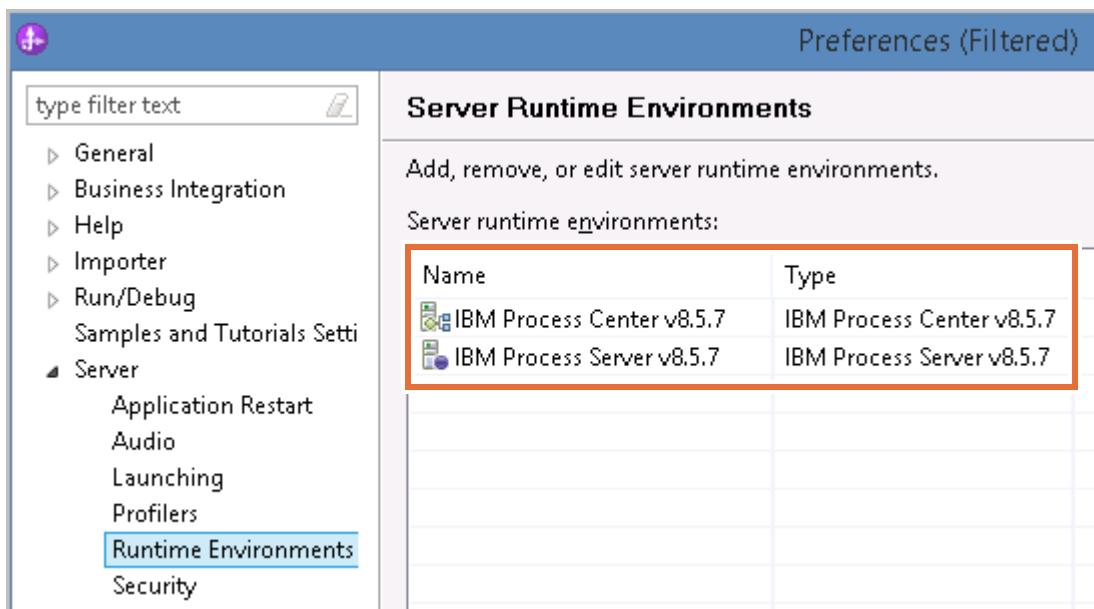
Preferences

In addition to capabilities, many other preferences can be configured. For more information about each of the preferences, see the online help. In this portion of the exercise, you examine a few of the more commonly used preferences settings.

To examine the IBM Integration Designer preferences:

- ___ 1. In the **Preferences** dialog box, in the navigation pane, expand **Business Integration**. Note the available preferences, including the settings for the Integration Designer assembly editors.

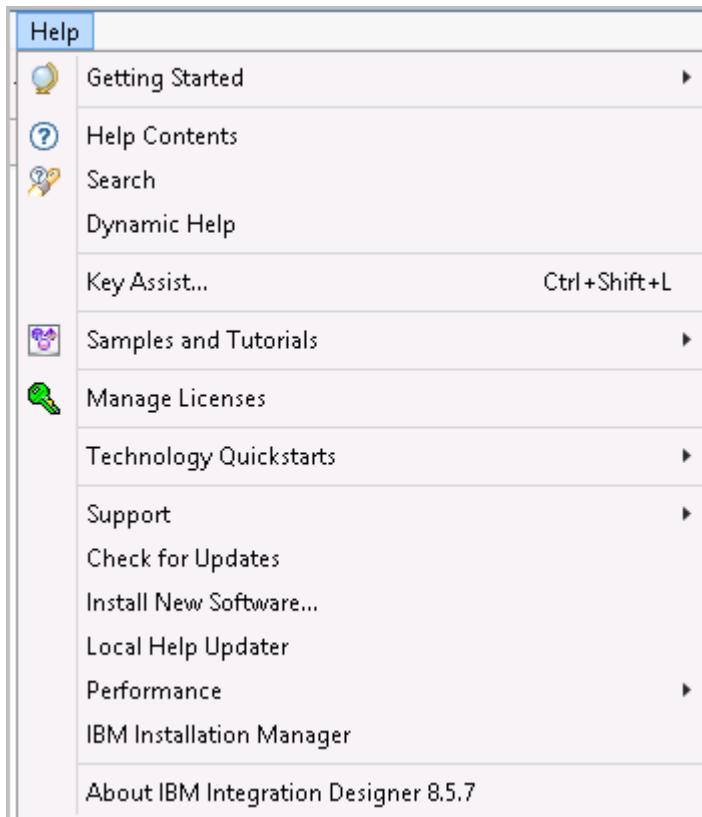
2. In the navigation pane, expand the **Server** section and select **Runtime Environments**. Note the available runtime environments and the ability to add and remove them.



3. Click **Cancel** to close the **Preferences** dialog box.

Part 2: Use the help menu to search for assistance

The help menu is a great resource for finding more information about all aspects of IBM Integration Designer. In this part of the exercise, you explore the options available from the help menu.



The help menu in IBM Integration Designer contains the following options (you are encouraged to open each of these resources and explore as you read):

- **Getting Started:** This option opens the Welcome pages that are associated with either IBM Integration Designer or Process Center for Integration Designer. You can view either of these pages when a workspace is first started. Each page contains links to Samples and Tutorials, Task Flows, Reference Materials, and Patterns.
- **Help Contents:** This option opens the IBM Knowledge Center documentation on IBM Integration Designer on the local computer.
- **Search:** This option starts a help view that can be used to search multiple information sources concurrently. You can search the web, the tutorials gallery, eclipse.org, IBM developerWorks, and the IBM Knowledge Center documentation on WebSphere Application Server.
- **Dynamic Help:** This option provides information relevant to the current task you are working on in the workbench.
- **Key Assist:** This option provides you with a list of relevant keyboard shortcuts. Press Esc to close the list.
- **Samples and Tutorials:** This option provides links to the IBM Integration Designer application samples available from the Getting Started page.
- **Manage Licenses:** This option provides status and update capacity for the current license.
- **Technology Quickstarts:** This option provides links to tutorials. Some of these tutorials are as follows: tutorials that are viewed like movies, play-and-learn tutorials that guide you through a simulation, do-and-learn tutorials for hands-on learning, and tours that contain multimedia displays that teach you conceptual information.
- **Support:** This option provides you with several useful links to external support resources such as the IBM user forums, IBM developerWorks, the IBM Education Assistant, IBM Technical Support, and the IBM Support Portal.
- **Check for Updates, Install New Software, Local Help Updater, and IBM Installation Manager:** Each of these options provides access to the IBM Installation Manager, which walks you through installing new or updating existing functions.
- **Performance:** This option provides features such as immediately reducing the amount of memory that is used, generating diagnostic information, and enabling system profiling.
- **About IBM Integration Designer 8.5.7:** This option displays the version information and information about installed features and plug-ins.

**Note**

In addition to the resources provided directly in IBM Integration Designer, you can also download and install the IBM Support Assistant from www.ibm.com. You can use the support assistant to search for an answer to your question or problem in many different locations at the same time. You can also get speedy access to critical product information or run free troubleshooting and diagnostic tools on a troublesome application. You can shorten the amount of time it takes to resolve an issue with automated data gathering and submission tools.

Part 3: Use the Installation Manager to find product updates

The IBM Installation Manager utility is used to install IBM Integration Designer product updates (and other Eclipse-based IBM product updates). The IBM Update Installer is used to install the IBM Process Server test environment. For more information about updating the test environment, see the product documentation.

**Warning**

It is important that no software updates are made in this exercise. The purpose of this section is to demonstrate how the updates are done and not to make those updates to this image. Making any software updates can cause unpredictable image behavior and lab results.

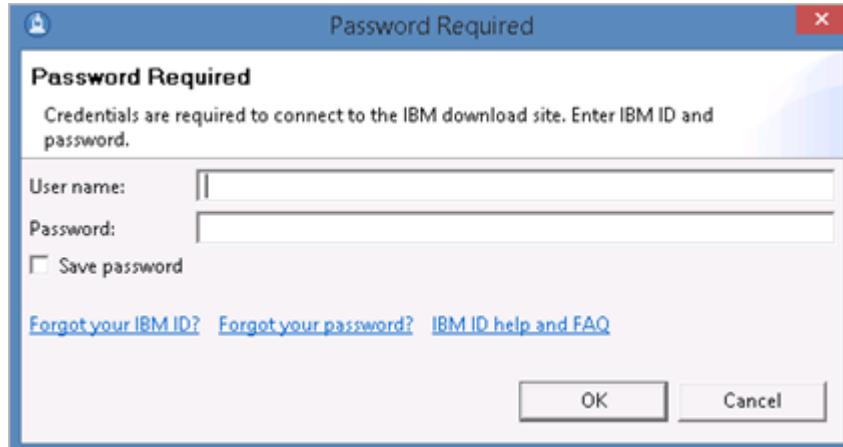
To check for IBM Integration Designer product updates:

- 1. Start the IBM Installation Manager utility by clicking **Help > IBM Installation Manager**.
- 2. Click the **Update** icon. You might receive errors such as: The following repositories are not connected.

**Important**

These steps are designed to show you the update process. You do not update IBM Integration Designer in this course. IBM Business Process Manager V8.5.7 Advanced is used to test the exercises and course materials. Updating your software can produce unexpected results.

- ___ 3. If you receive a warning message that the resources are not connected, click **OK**. If no errors are displayed, then at the Update Packages dialog box, click **Next**.
- ___ 4. If a dialog box is displayed prompting you to enter credentials, then click **Cancel** to close the window. The dialog box might open more than one time. If it does, then click **Cancel** again to close the Password Required dialog box.



- ___ 5. Click **Cancel** to close the Update Packages window. The purpose of this exercise is to explore the update feature, not to update the WebSphere tools.
- ___ 6. Click **File > Exit** to close Installation Manager.

Part 4: Browse the Business Integration perspective and views

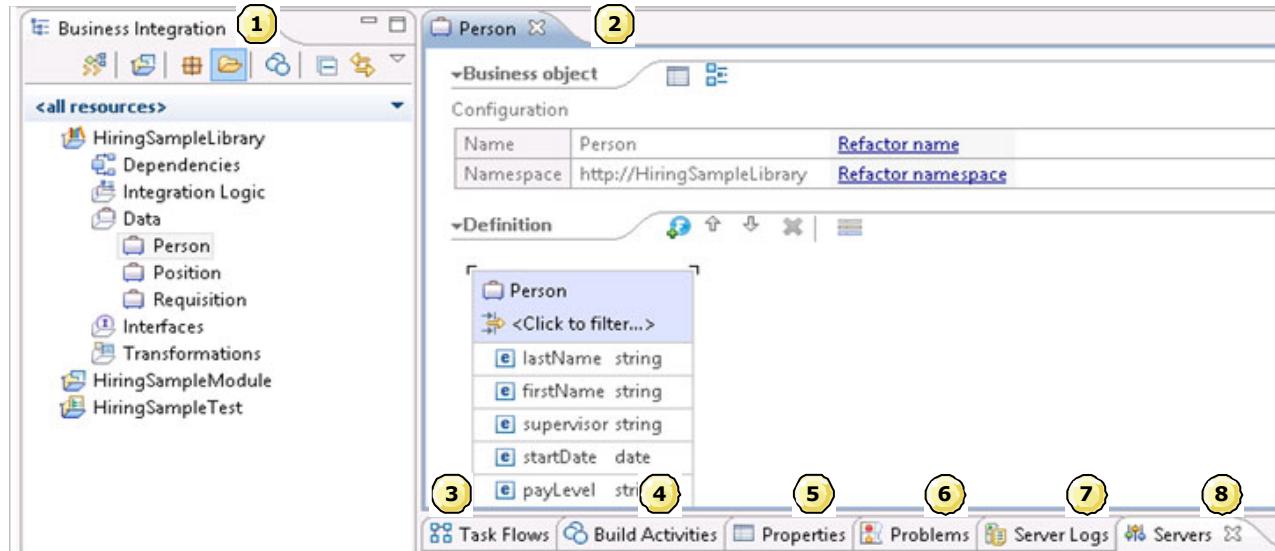
In this portion of the exercise, you browse the different views and perspectives in IBM Integration Designer. A perspective is a specific arrangement of views in the workbench. The user's active perspective controls which views are shown on the workbench page and their positions and sizes. A view is typically used to browse a hierarchy of information, open an editor, or display properties for a selected object. Users can switch between perspectives as they change tasks.

Each perspective provides a set of functions that are aimed at accomplishing a specific type of task or working with specific types of resources. Therefore, opening a different perspective opens a different group of views. For example, the Java perspective combines views that you would commonly use while editing Java source files. The Debug perspective contains the views that you would use while debugging applications.

- ___ 1. Examine the Business Integration perspective.

By default, when you start IBM Integration Designer, the Business Integration perspective is opened. Most of your integration development is done in the Business Integration perspective. Notice that since you just created a workspace, it is empty. The Business

Integration perspective has the following views and panes (you are encouraged to open each of these components and explore as you read).



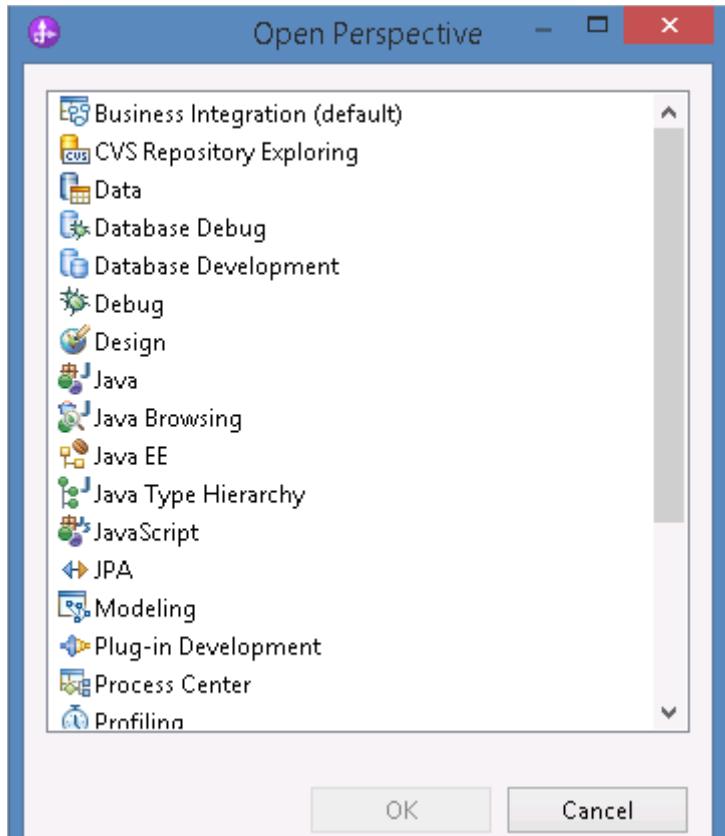
- 1) **Business Integration** view: This view provides a logical view of the key resources in each module, mediation module, and library. Non-SCA projects are displayed in the Business Integration view. You can use this view to edit projects without switching perspectives.
- 2) **Editor pane**: When you open a resource from the Business Integration view with an editor, the resource is displayed in the editor pane.
- 3) **Task Flows** view: Task flows provide a way to learn related concepts and tasks in an interactive manner. A task flow puts together all of the different tasks that are required to achieve a goal. Task flows are presented as a series of grouped tasks, where each task is a link. When you click the links, wizards and editors start along with information windows so that you are learning while you work on tasks.
- 4) **Build Activities** view: This view helps you manage builds. You can view the build and server status of business integration projects.
- 5) **Properties** view: In this view, you can see information about a selected artifact. When using the Properties view with editors, you are able to modify properties of elements you select.
- 6) **Problems** view: This view helps you debug errors by providing message text. You can see more help for the problem message by selecting a message and pressing F1.
- 7) **Server Logs** view: This view is used to display the contents of the server console and server log files. It automatically displays console output for each server that is started.
- 8) **Servers** view: Use the Servers view to create test servers (local or remote). Use the view for testing and deploying business integration applications and for viewing the status of deployed modules.

Browse other perspectives

In addition to the Business Integration perspective, several other perspectives are available in IBM Integration Designer. These additional perspectives are designed to accommodate different development tasks.

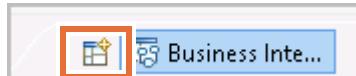
To switch perspectives:

- 1. Click **Window > Open Perspective > Other** to see a list of other perspectives.



- 2. Choose the perspective that you want to open, and click **OK**. The open perspectives are listed at the upper-right corner of the workbench window.

You can also open a new perspective by clicking the **Open Perspective** icon.



- 3. To switch perspectives, click **Window > Navigation > Next Perspective** or click the right-pointing double arrows **>>** in the upper-right corner of the workbench. (It might be necessary to have several perspectives open to work on these actions. You cannot use the double arrow method in the Process Center perspective.)
- 4. To close a perspective, click **Window > Close Perspective**, or right-click the perspective in the upper-right corner of the workbench window and click **Close**.

Customizing a perspective with views

Perspectives offer predefined combinations of views. To open a view that is not included in the current perspective, click **Window > Show View** from the menu bar. In addition to adding a view to a perspective, you can maximize views, minimize views, move views, dock views, close views, open views, and create fast views.

If you want to minimize and maximize views, you can do so in several ways:

- Double-click the title bar to maximize a view so that it fills the workbench window. Double-click the view again to restore it to its original size.
- Use the minimize and maximize icons that are attached to the view (click the restore icon after maximizing or minimizing it). A minimized view is visible as a toolbar on the perspective. Use restore or maximize to reestablish the view.



- Right-click the title bar of the view and click **Minimize** or **Maximize** from the menu. After maximizing the window, right-click the title bar and click **Restore** to restore the window to its original size.



Hint

If you close IBM Integration Designer, the last used perspective is preserved. In other words, if you add and format views in a perspective, the new layout is preserved when you close and reopen Integration Designer. However, if you change perspectives, your customized layout is lost. To save your new layout as a customized perspective, click **Window > Save Perspective As**.

A perspective can be customized to fit personal needs and saved. A simple relocation of a view might be all that you need for it to be more productive. You can click the title bar at the top of a view and drag it to a new location. As you move the view around the workbench, the mouse pointer changes to a drop cursor. The drop cursor indicates where the view would be docked in case you release the left mouse button.

To change the location of a view:

- 1. Click the title bar of the **Build Activities** view, hold the left mouse button down, and drag the view. Drag the view to the left, right, top, or bottom border of another view. You can also drag the view outside of the workbench area to turn it into a detached view.
- 2. When the view is in the location you want, release the left mouse button.
- 3. (Optional) If you want to save your changes, click **Window > Save Perspective As** from the main menu bar.

**Note**

You can also use the menu to move a view by right-clicking anywhere in the title bar of the view. This menu provides shortcut options for turning a view into either a “fast” or a “detached” view.

Fast and detached views

A fast view provides a quick means to display and hide a frequently used view. When a fast view is created, its icon is placed in the bottom, fast view toolbar. Click that icon to display or hide the view. Fast views work like other views except that they do not take up space in your workbench window.

To create a fast view:

- 1. Click **Help > Search** from the menu options. It might take a few minutes for indexing to complete, and its status is displayed at the lower right of the Help view.
- 2. Right-click the title bar of the **Help** window and click **Fast View** from the menu. (You can also drag an open view to the fast view toolbar in the lower-left corner of the status bar.)
- 3. To access the fast view, click the icon in the lower-left corner of the status bar:



- 4. To hide the fast view again, click the minimize icon in the title bar of the view, or click outside the fast view window. If you click the **X** to close the view, the view is removed from the fast view toolbar.

**Note**

Any of the views in the current perspective can be displayed as fast views by clicking the **Show View as fast view** icon.

The fast view toolbar can be docked on either side of the workbench window.

When a view is detached, the view is free to “float” in the development environment, much like a separate window. To close a detached view, use the **X** in the upper-right corner of the detached view.

Part 5: Examine the modules and libraries of a business integration project

In this portion of the exercise, you browse a business integration project. The first step in developing an application in IBM Integration Designer is to set up one or more projects to hold your resources. Projects are the largest structural units in your workspace.

Types of projects

A project is an organized collection of folders or packages. Projects are used for building, version management, sharing, and organizing resources that are related to a single work effort. The projects that you work with most in IBM Integration Designer are modules, libraries, and mediation modules, but several types of projects are available:

- **Integration solution:** An integration solution is a nondeployable project that is used solely to reference other projects in the workspace, such as modules, mediation modules, libraries, component test projects, Java projects, and simple projects. Integration solutions help you organize groups of related projects and more easily do common actions on the projects, such as sharing them in a team environment. When you create an integration solution, an integration solution diagram is automatically generated. You can open the integration solution diagram in the integration solution editor and view the relationships between the related projects that are referenced in the integration solution.
- **Modules:** Modules provide the business services for your application, which are modeled as Service Component Architecture (SCA) components that are wired together in a module assembly. Modules are the basic units of deployment to the IBM Process Server runtime environment.
- **Mediation modules:** Mediation modules provide mediation service applications, which intercept and modify messages that are passed between existing services (providers) and clients (requesters) that use those services. Mediation modules can be deployed on IBM Integration Bus (formerly WebSphere Enterprise Service Bus) or IBM Process Server. You learn more about mediation modules later in this course.
- **Libraries:** Often interfaces, data types, transformations, business calendars, and web service ports must be shared so that resources in more than one module can use them. The library is a project that is used to store these resources. Libraries are associated with modules through dependencies.
- **Component test project:** To test modules and applications in the workspace, you might want to create test suites that contain sample data. A component test project is a project that is kept separate from your working, production data. It contains testing tools such as test suites, execution traces, server configurations, emulators, and collections of sample data called “data pools”.
- **Modules and library dependencies:** When developing and deploying integration applications, it might be necessary to declare dependencies for your modules, mediation modules, and libraries. Use the dependency editor to manage these required resources.
- **Namespaces:** A namespace is a logical container in which all names are unique. A name can be displayed in multiple namespaces but cannot be displayed twice in the same namespace.
- **Other projects:** You can also bring Java projects into your application. In some cases, you might choose to do bottom-up development by designing Java resources and then importing them into your IBM Integration Designer application. If you have Java code that is used in a business integration module or mediation module, you can create a

Java project. You can add that Java project as a dependency in the module that uses the code.



Note

You can easily tell mediation modules from integration modules in the Business Integration view by examining the icons. The icon for a mediation module is a folder with an envelope. The icon for an integration module is a folder with a document.

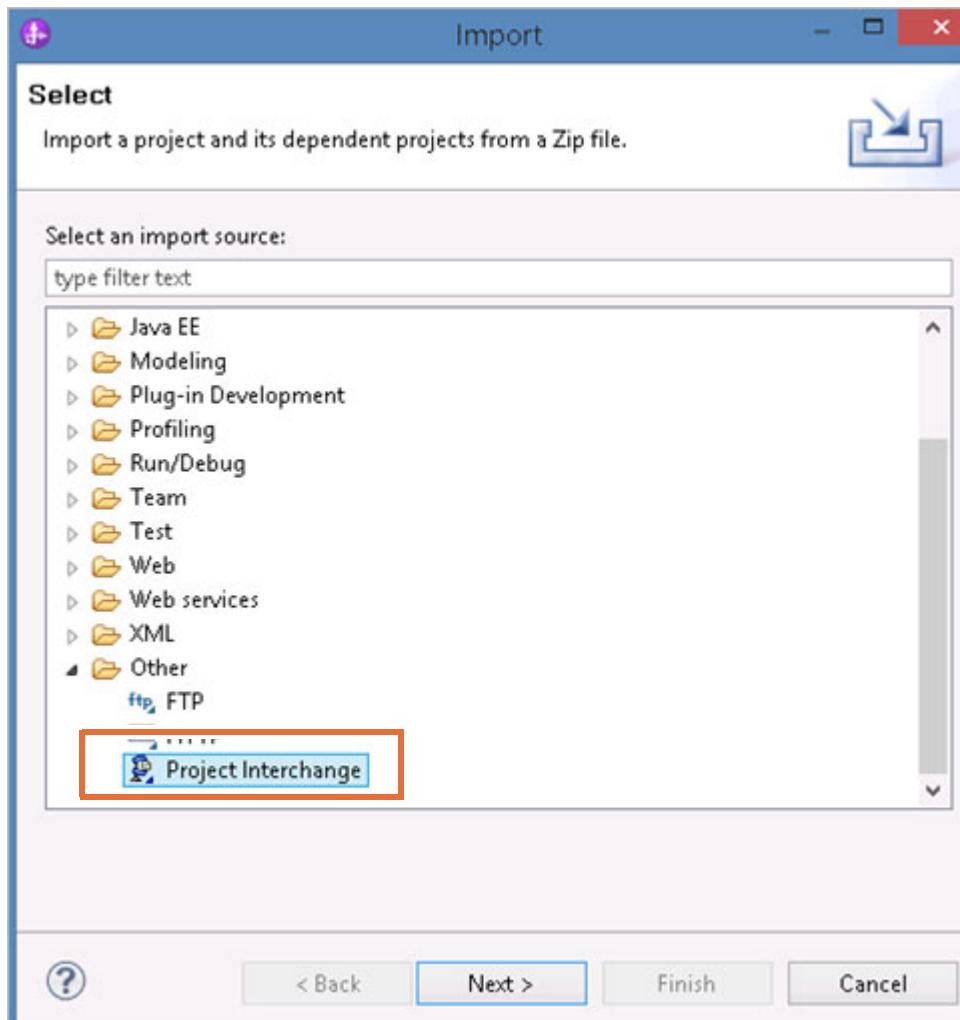
The structure of business integration modules

The Business Integration perspective has a Business Integration view, which provides a logical view of the key resources in each module, mediation module, and library. Within each project, the resources are categorized according to type. Logical resources that are displayed in the navigation tree do not necessarily have a one-to-one mapping to files. For example, a business process that is displayed in the integration logic section might comprise several different physical files. You can also use the Business Integration view to browse through Java and Java EE resources.

To browse the resources in a business integration module:

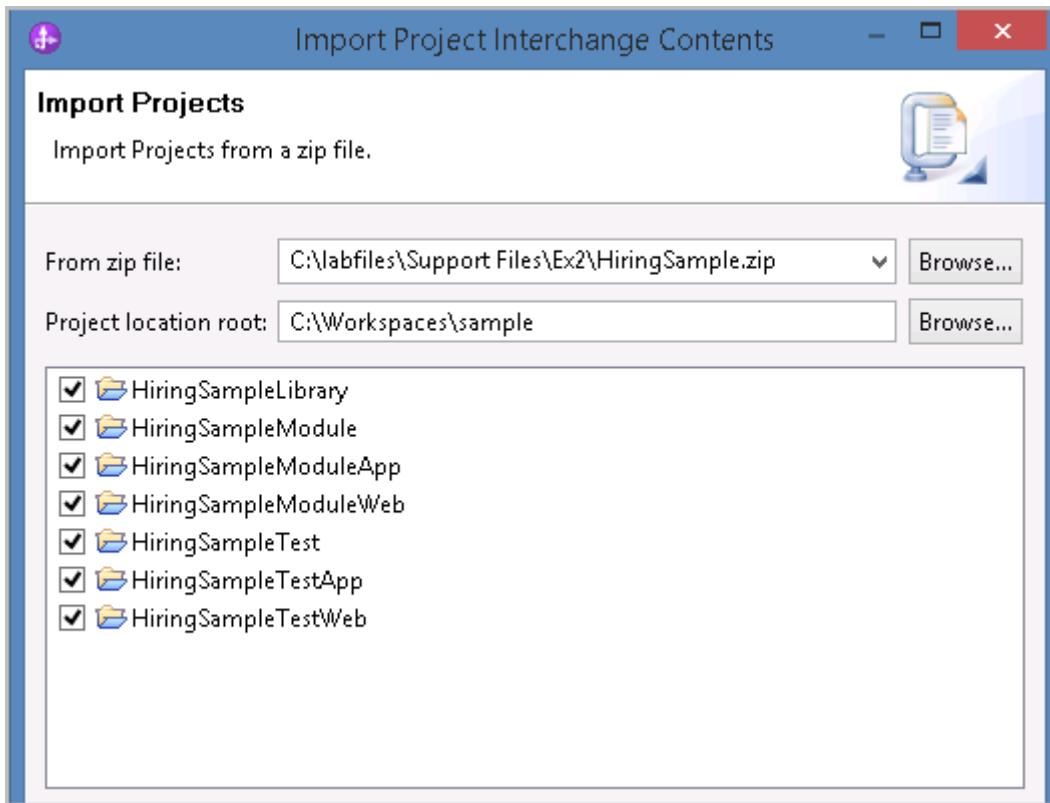
- ___ 1. If you opened another perspective in the previous section, switch to the **Business Integration** perspective.
- ___ 2. Import the **HiringSample** process application.
 - ___ a. Right-click within the Business Integration view and click **Import**.

- ___ b. In the Import window, select **Other > Project Interchange**.



- ___ c. Click **Next**.
- ___ d. In the **Import Project Interchange Contents** dialog box, select **Browse** next to the **From zip file** field.
- ___ e. Browse to the `C:\labfiles\Support Files\Ex2` directory and select **HiringSample.zip**.

- __ f. Click **Open**. The contents of the hiring sample project interchange file are displayed.



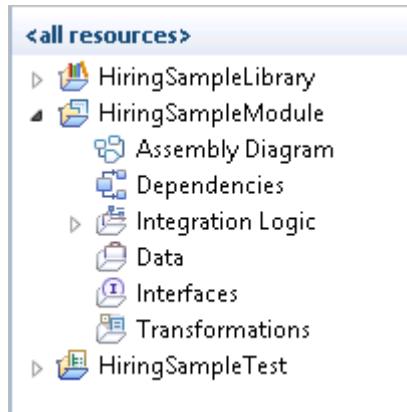
- __ g. Click **Finish**. Allow a few moments for Integration Designer to import the project file. After the file is imported, the workspace is built. Wait until the status reaches 100% at the lower right of the Integration Designer. When the workspace is built, the status progress disappears.



Important

This Hiring Sample project that you are importing is similar in its structure and function to the hiring sample copy process application that you explored in the previous lab. However, this import is a clone, not the same process application.

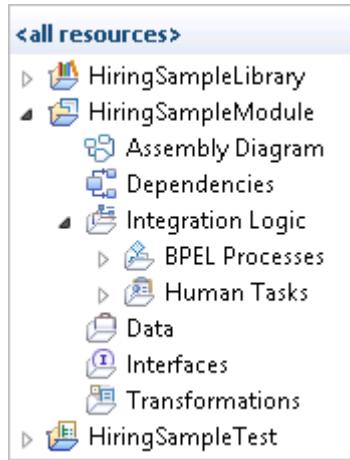
3. Click the arrow to expand the contents of **HiringSampleModule**.



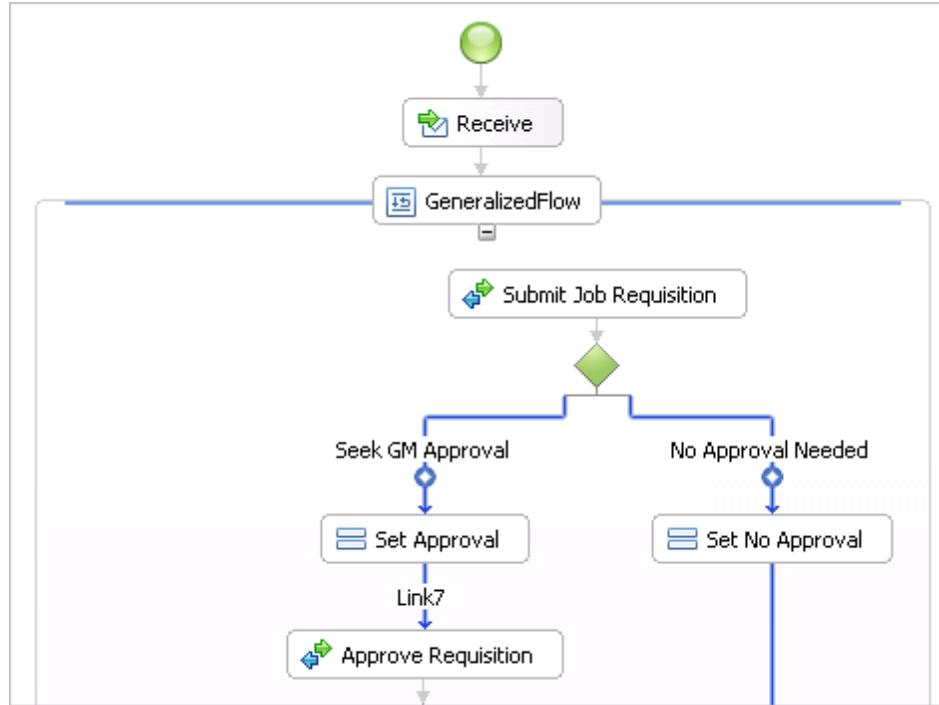
Each business integration module consists of the following resources:

- **Assembly Diagram:** The building blocks of a business solution are Service Component Architecture (SCA) components that are wired together to form modules that can be deployed to IBM Process Server. You can use the IBM Integration Designer assembly editor to build applications by assembling the SCA components on an assembly diagram.
- **Dependencies:** Double-click **Dependencies** to open the dependencies editor to see and manage the dependencies on other modules or libraries.
- **Integration Logic:** The Integration Logic section contains all of the artifacts that do specific business tasks. Business processes, human tasks, and mediation flows are some examples of integration logic implementations.
- **Data:** The Data section contains all of the data representations. For example, it contains business objects and user-defined simple types.
- **Interfaces:** The Interfaces section contains WSDL interfaces that define the operations that a service exposes and makes available to callers.
- **Transformations:** The Transformations section contains data transformations such as data maps and Extensible Stylesheet Language (XSL) transformations.

- ___ 4. Expand the **Integration Logic** section in the **HiringSampleModule** and examine the artifacts.



- ___ 5. The **Integration Logic** section contains **BPEL Processes** and **Human Tasks**. Expand **BPEL Processes** and double-click **OpenNewPosition**. The BPEL process opens in the editor.

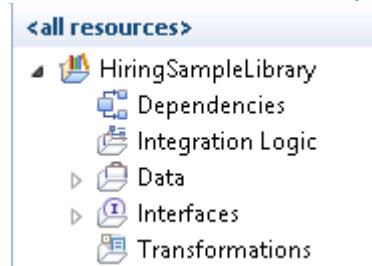


- ___ 6. Take some time to examine the steps of this business process. Can you determine the intended activities along each step of the process? Do not change the process.
___ 7. Close any open tabs when you are done.

The structure of libraries

Each of the artifacts you create in a library can also be included in a module. For example, you can create a business object in the Data section of a module, and you can create a business object in the Data section of a library. However, artifacts that are created in a module cannot be shared among projects, while artifacts created in a library can be shared among projects.

- 1. Examine the artifacts in the **HiringSampleLibrary**.
 - a. In the Business Integration view, expand **HiringSampleLibrary**.
 - b. Note the artifacts that can be included in the library:



- **Dependencies:** Double-click **Dependencies** to open the dependencies editor to see and manage the dependencies on other modules or libraries.
- **Integration Logic:** Integration Logic contains all of the artifacts that do specific business tasks. Business processes, human tasks, and mediation flows are some examples of integration logic implementations. When defined in a library, these artifacts can be reused in other projects.
- **Data:** The Data section contains all of the data representations. For example, it contains business objects and user-defined simple types. You can define these data representations in a library to use them in other applications.
- **Interfaces:** The Interfaces section contains WSDL interfaces that define the operations that a service exposes and makes available to callers. Interfaces in a library can be used in multiple modules by several components.
- **Transformations:** The Transformations section contains data transformations such as data maps and XSL transformations. Data transformations that are defined in a library are available for use by multiple applications.

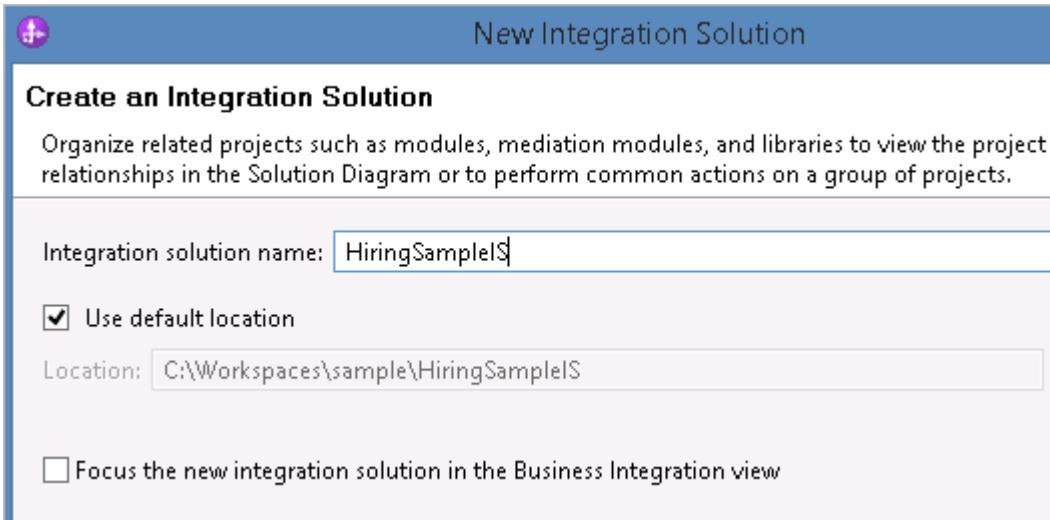
Adding an integration solution

An integration solution is a nondeployable project that is used solely to reference other projects in the workspace, such as modules, mediation modules, libraries, component test projects, Java projects, and simple projects. You use the loan application to create an integration solution. This exercise is intended to introduce you to integration solutions. In a later exercise, you have the opportunity to explore integration solutions in detail.

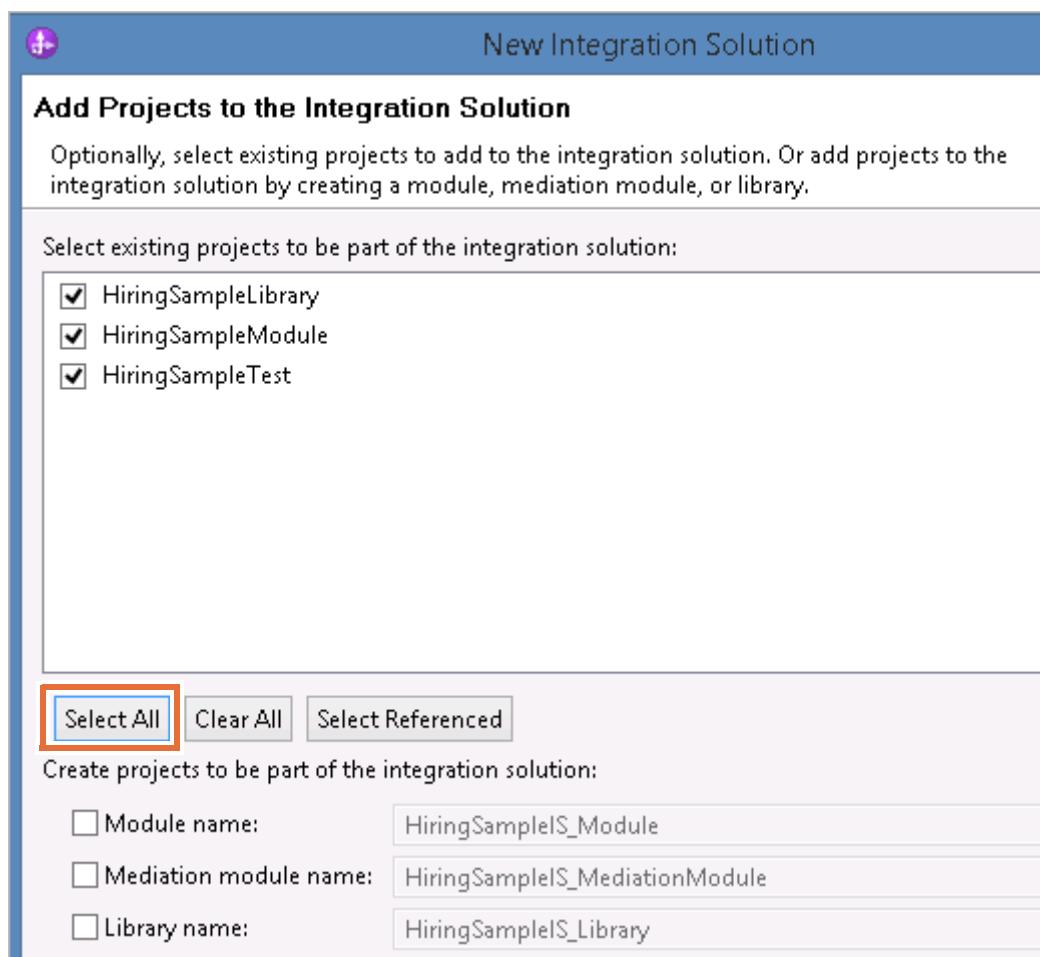
To create an integration solution:

- 1. Click **File > New > Integration Solution**.

- 2. In the **Integration solution name** field, enter: HiringSampleIS

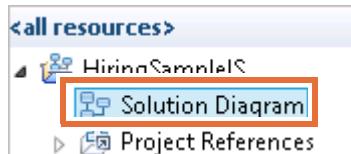


- 3. Ensure that the **Focus the new integration solution in the Business Integration view** check box is not selected and click **Next**. If you have multiple integration solutions, this option hides all of them except the one you are creating.
- 4. Click **Select All** to add all of the existing projects to the integration solution.



- 5. Click **Finish**.

- ___ 6. Expand **HiringSampleIS** and double-click **Solution Diagram**.



- ___ 7. In the solution diagram editor, click the **Show All Projects** icon. This action changes the view from the default: "Only Show Modules."



The solution diagram is a graphical representation of the artifacts that were selected when you created the integration solution.



- ___ 8. As time permits, feel free to experiment with the other options in the integration solution feature.
 ___ 9. Close the **HiringSampleIS - Solution Diagram** tab.
 ___ 10. Save your changes.

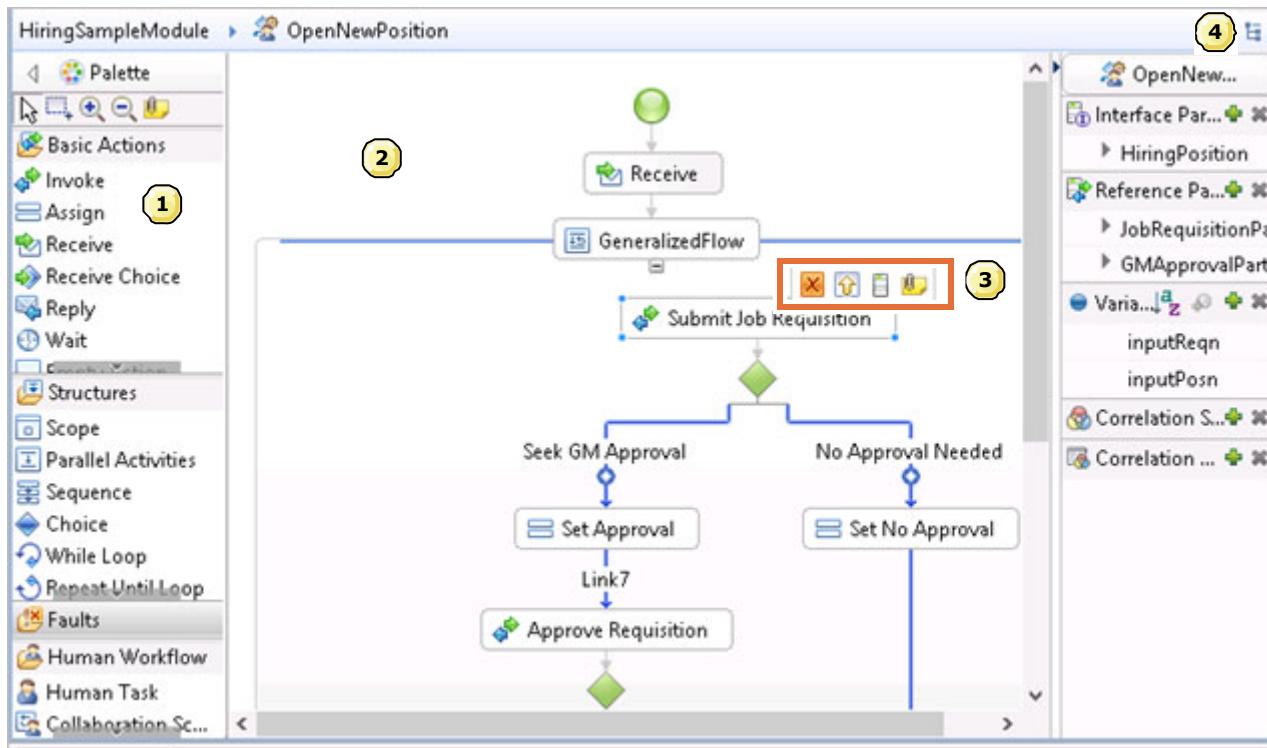
Part 6: Browse the IBM Integration Designer graphical editors

In this portion of the exercise, you browse two of the more commonly used IBM Integration Designer editors. An editor is typically used to edit or browse a document or input object. Modifications that are made in an editor follow an open-save-close model, much like an external file system editor. The platform text editor and Java editor are examples of workbench editors. Each editor has the same basic structure, but the specific areas that are available differ depending upon the object that you are editing.

To open editors in IBM Integration Designer:

- ___ 1. Open **OpenNewPosition** in the business process (BPEL) editor.
 - ___ a. In the Business Integration view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.
 - ___ b. Double-click **OpenNewPosition** to open the business process (BPEL) editor, if not already open.

- ___ c. The business process editor is divided into the following components. Feel free to explore as you read.

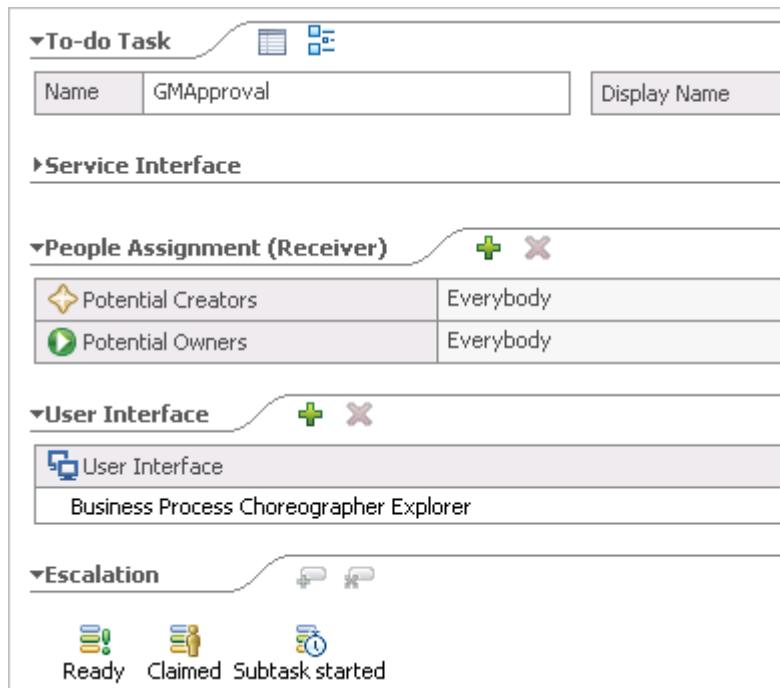


- 1) **Palette:** The palette is the area to the left of the canvas that houses the activities you drag onto the canvas to build your process. The icons are organized under several different headings that act as toggles. Click the heading, and the icons remain hidden until you click that heading again. The complexity of the palette is reduced and hides those icons that you rarely use.

You can gather frequently used items into a favorites folder by right-clicking the icon and clicking **Add to Favorites**. (If the favorites folder does not exist, it is created.) To hide the text labels when you are familiar with the icons, right-click the palette, and toggle the **Show Labels** setting. To increase the size of the icons, right-click the palette, and toggle the **Use Large Icons** setting.
- 2) **Canvas:** The canvas is the white, empty area in the middle of the editor that you use to compose your business process. When you drag an activity from the palette onto the canvas, the icon beside your cursor has a plus (+) symbol, and you can decide where you want to drop the activity. When the cursor becomes a crossed-out circle, continue moving the cursor until it becomes a plus sign again.
- 3) **Action bar:** The action bar is a miniature dialog box that is displayed next to certain activities when you select them, and it contains a series of one or more icons that are relevant to that activity.
- 4) **Tray:** The tray displays the partners, variables, correlation sets, and correlation properties that are associated with your process (each of these

items is defined later in the course). To see the interfaces and operations that are associated with the partners, click the small gray arrow beside the name of the partner. To create an item, click the green + icon; or to remove one, highlight it and click the red X icon. Click the gray horizontal arrow to collapse or expand the tray.

- ___ d. After examining the components, close the process editor by clicking X on the title bar. If you changed any components, do not save them.
- ___ 2. Open the **GMAApproval** human task editor.
 - ___ a. In the Business Integration view, expand **HiringSampleModule > Integration Logic > Human Tasks**.
 - ___ b. Double-click **GMAApproval** to open the human task editor.



- ___ c. View the options in the human task editor. After examining the task, close the editor.

Part 7: Use the IBM Integration Designer documentation capabilities

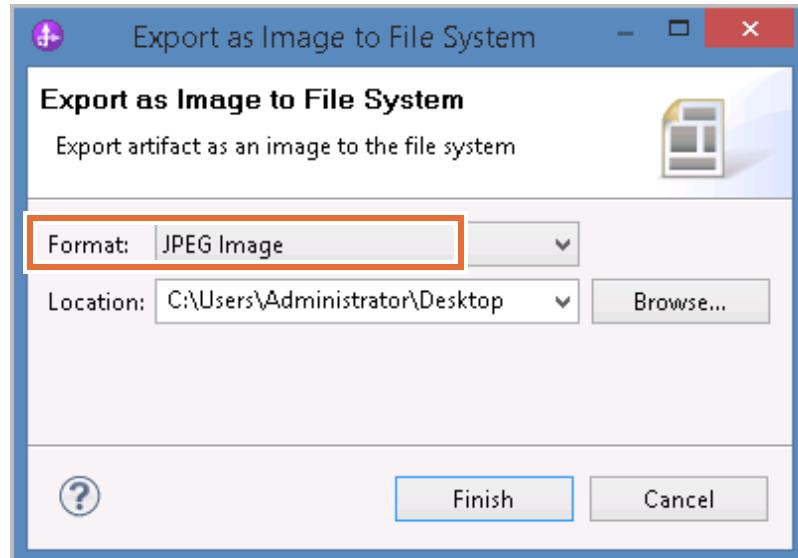
In this portion of the exercise, you examine some of the project documentation features in IBM Integration Designer.

Exporting a business process image

If you want to share a business process with others who do not have IBM Integration Designer, you can export the process as an image for them to review. You can also use this feature to create project documentation for later review or for governance. To export a business process as an image file:

- ___ 1. In the Business Integration view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.

- ___ 2. Right-click **OpenNewPosition** and click **Export as Image** from the menu.
- ___ 3. In the **Export as Image to File System** dialog box, enter the following information:
 - For **Format**, select **JPEG Image** from the list. Other supported files include **PNG Image**, **SVG Document**, and **TIFF Image**.
 - For **Location**, click **Browse**, go to your desktop, and click **OK**.

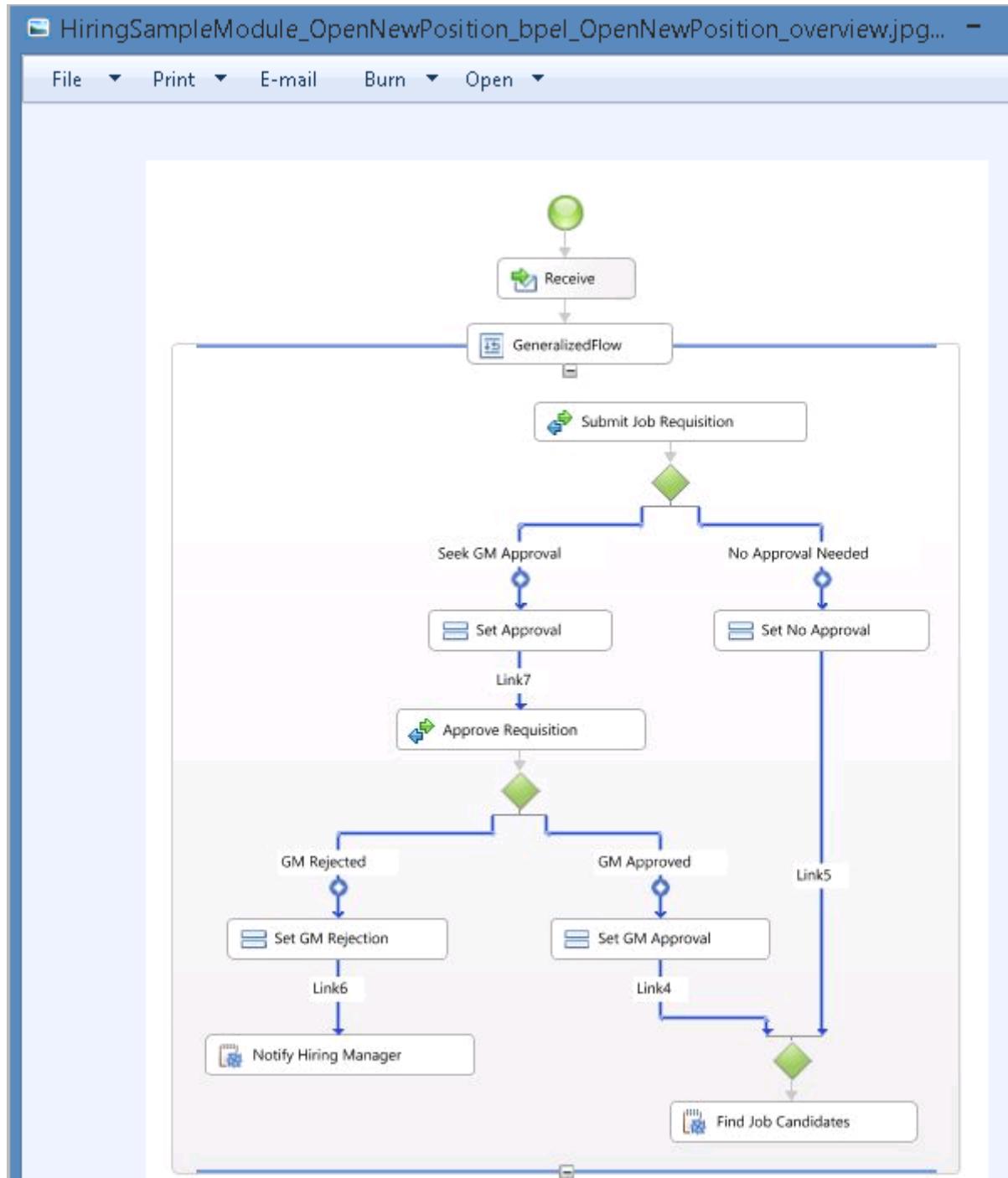


- ___ 4. Accept the remaining default options and click **Finish**.
- ___ 5. When you receive the **Export as Image Result** dialog box, the message **The following images have been exported successfully** is displayed.

The following images have been exported successfully:			
Image	Logical Artifact	Project	File
C:\Users\Administrator\Desktop\HiringSampleModu...	OpenNewPositi...	HiringSampleMod...	OpenNewPosition.bpel

- ___ 6. Click **OK**.
- ___ 7. Locate the newly saved image on the desktop and double-click **HiringSampleModule_Open newPosition_bpel_Open newPosition_overview.jpg**.

8. The image file loads in the Windows Photo Viewer.



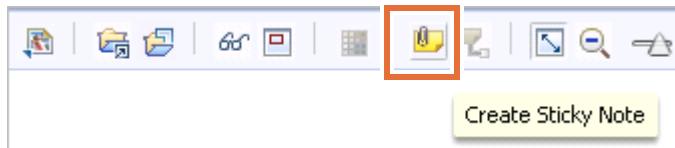
9. Close the Windows Photo Viewer and maximize IBM Integration Designer.

Creating sticky note documentation

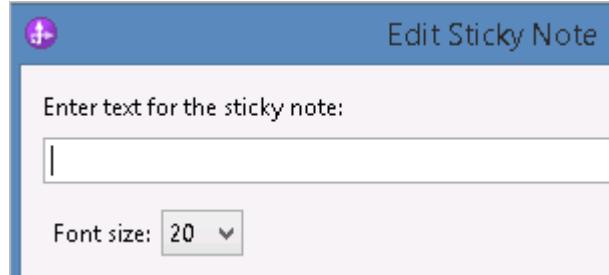
In addition to exporting project components as images, you can also create embedded documentation by adding sticky notes. Sticky notes can be used to create reminders for developers or to create task lists by using Javadoc task tags.

To create sticky note documentation in your integration solution diagram:

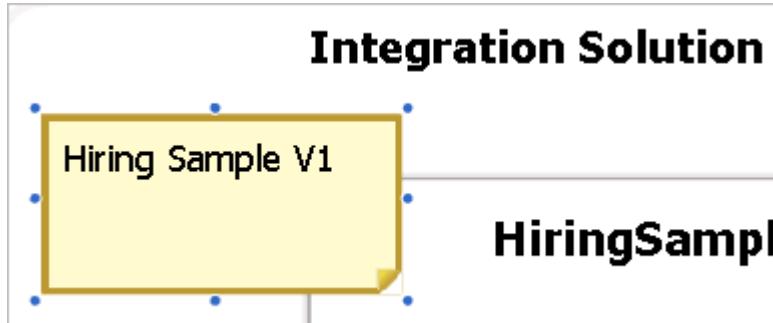
- ___ 1. In the Business Integration view, expand **HiringSampleIS** and double-click **Solution Diagram**.
- ___ 2. Click the **Create sticky note** icon:



- ___ 3. Click the diagram in the location where you want to place the note.
- ___ 4. To open the note editor, right-click the note and click **Edit Sticky Note** from the menu.



- ___ 5. In the **Enter text for the sticky note** field, type: **Hiring Sample V1**
- ___ 6. Click **OK**.



- ___ 7. You can resize the sticky note by selecting it and dragging one of the handles.
- ___ 8. Save your changes and close the solution diagram.

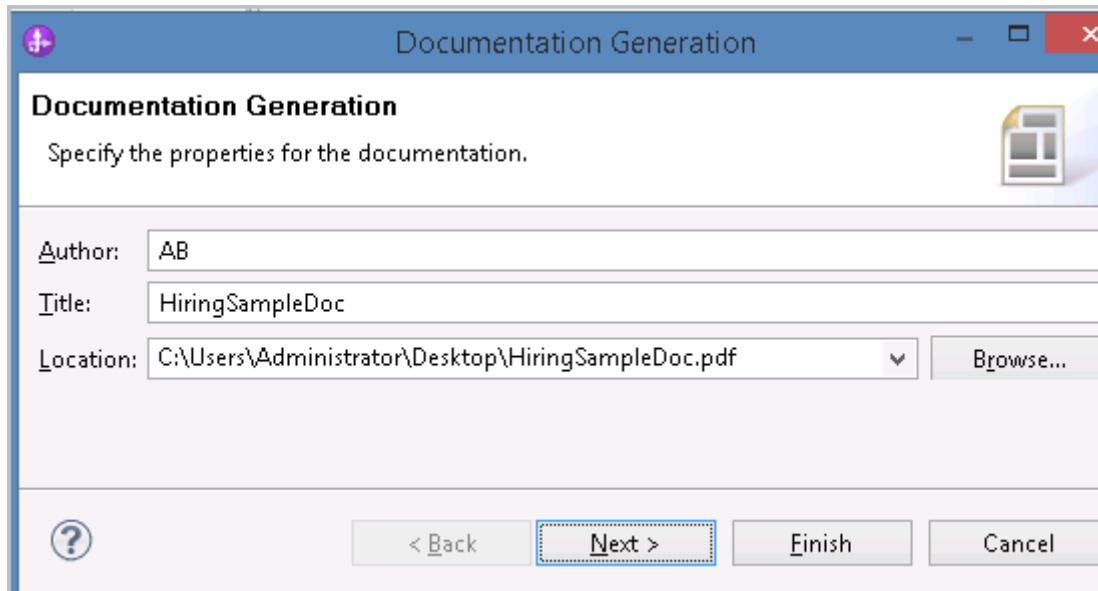
Generating documentation reports

In addition to images and sticky notes, you can generate project documentation reports. You can generate documentation for the artifacts in your modules, mediation modules, and libraries. You can customize the contents of the documentation by selecting multiple resources from one or more modules and libraries so that all of the information is written to a single report, in a PDF file. You can also specify the layout and fonts for the output.

To generate a documentation report:

- ___ 1. In the Business Integration view, in the Projects window, right-click **HiringSampleModule** and click **Generate Documentation** from the menu.

2. In the **Documentation Generation** dialog box, enter the following information:
- In the **Author** field, type your initials.
 - In the **Title** field, type: `HiringSampleDoc`
 - In the **Location** field, click **Browse**, browse to your desktop, type `HiringSampleDoc` in the **File name** field and click **Save**.



3. Click **Next**.
4. Ensure that **Generate documentation from the selected resource** is selected and click **Next**.
5. Accept the default **Paper size** and **Orientation** settings, note the other options on the **Layout Settings** dialog box, and click **Finish**.
6. When the report is generated, click **Yes** when you are prompted to open the generated report. Adobe Reader is installed in your lab environment, so you can view the generated report.

 **Note**

You might be required to accept the license agreement before using Adobe Reader for the first time.

7. After reviewing the report, close Adobe Reader.

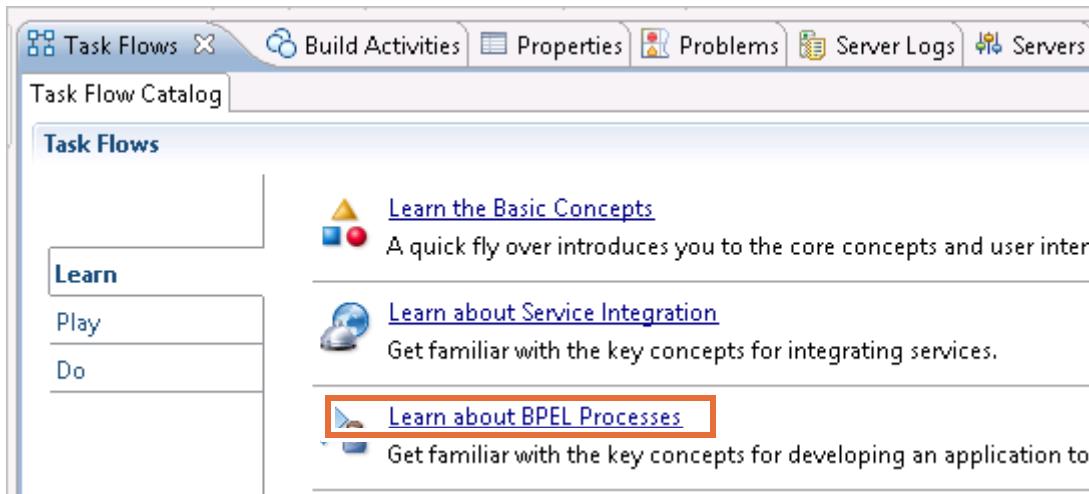
Part 8: Use the Task Flows view

Task flows are a way to learn related concepts and tasks in an interactive manner. Learning task flows briefly describes key concepts to quickly get you up to speed with IBM Integration Designer. Creating task flows shows you how to do key tasks.

A task flow puts together all of the different tasks that are required to achieve a goal such as creating a service, starting with setup or planning, and ending with testing. Task flows are presented as a series of grouped tasks; each task is a link. When you click a link, wizards and editors start, along with an information window, so that you are learning in the context of the task you are doing.

To work with the **Task Flows** view:

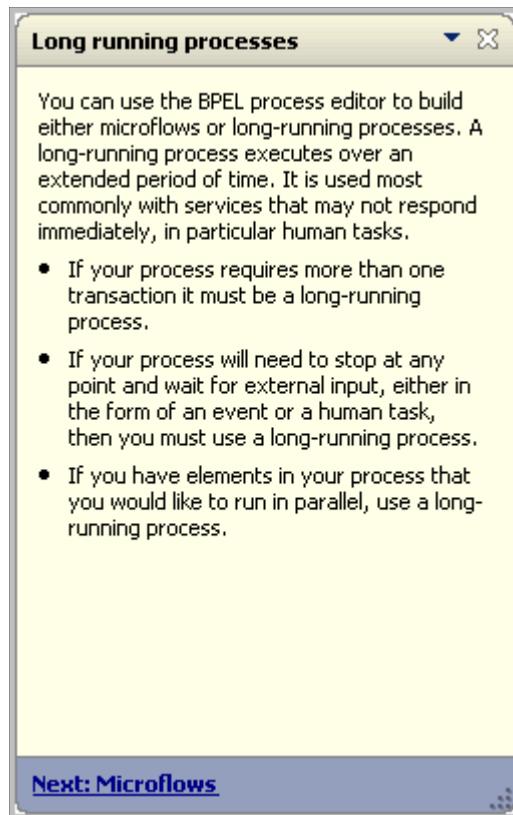
- 1. In the **Task Flows** view, open the education resources on business processes.
 - a. Switch to the **Task Flows** view.
 - b. In the **Task Flows** navigator, verify that you are on the **Learn** tab.
 - c. Click the **Learn about BPEL Processes** link.



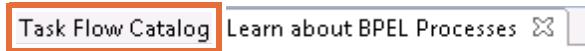
- d. After clicking the link, you are presented with a series of tasks that are involved in creating a business process. Feel free to click any of the links that interest you.



Each task that you click reveals a dialog box with information about the task.

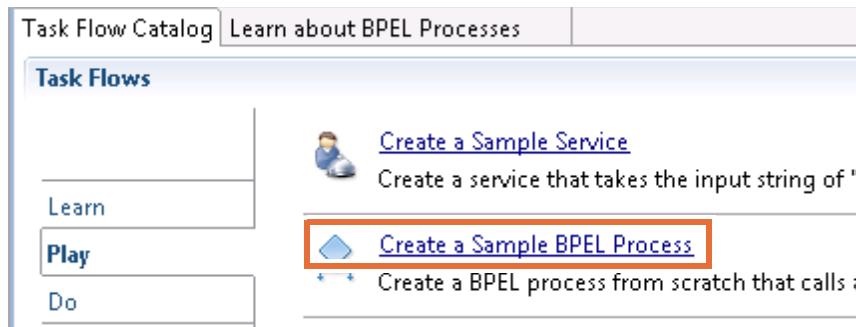


- ___ e. When you are done exploring in the **Task Flows** view, switch to the **Task Flow Catalog** tab.

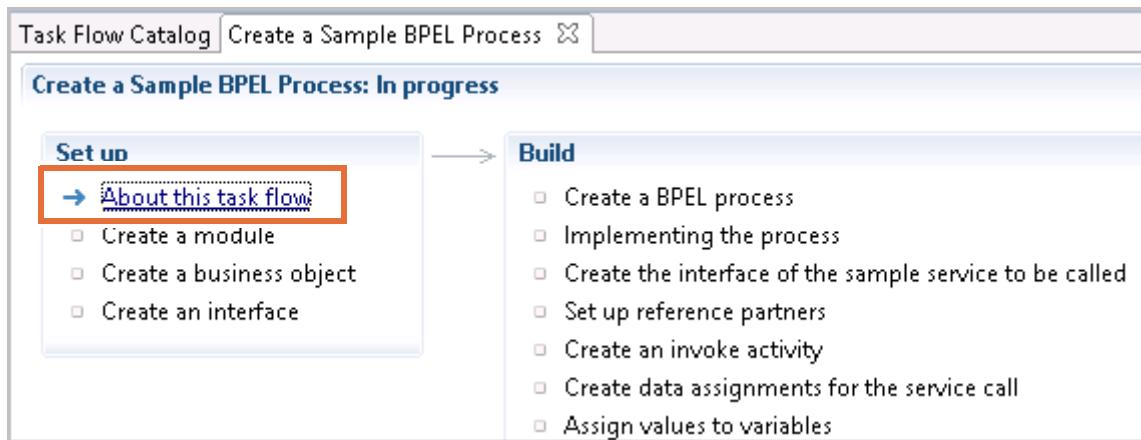


- ___ 2. Explore the interactive Task Flows samples.

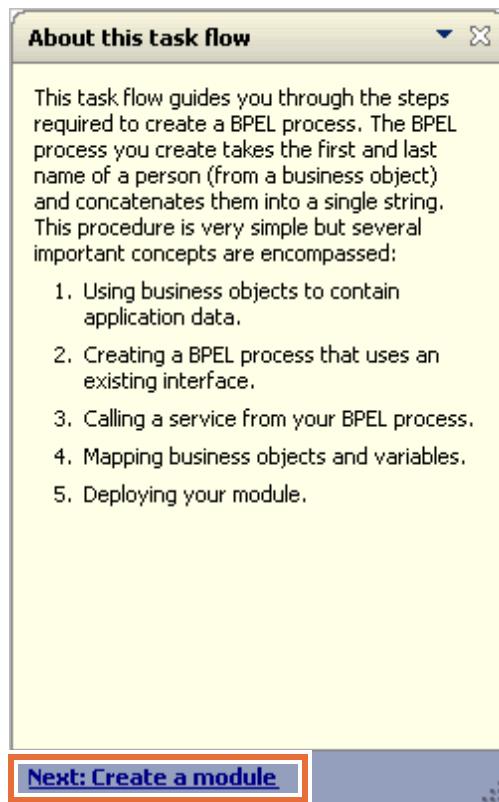
- ___ a. In the **Task Flows** navigator, switch to the **Play** tab.
___ b. Click **Create a Sample BPEL Process**.



- ___ c. When you click a link for one of the interactive samples, you are presented with a series of tasks that are involved in creating the relevant sample. You are expected to work on the tasks one at a time, in the order presented. Click **About this task flow**.

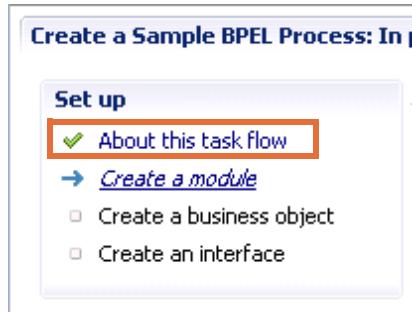


- ___ d. You are presented with a dialog box that explains the task. At the bottom of the dialog box is a link to the next task. Click **Next: Create a module**.



- ___ e. Clicking the link automatically opens the New Module dialog box for input. Close the dialog box and click **Cancel** in the New Module dialog box.

- ___ f. As you complete a task, your status is reflected in the Task Flows view. Each completed step is checked.



- ___ g. As time permits, you can continue creating the sample. Do not proceed past the tasks in the **Setup** section. If you have time at the end of the lab, you can return to this sample and continue.
- ___ 3. When you are finished exploring the sample, close IBM Integration Designer. You use a different workspace in the next exercise. Do not save any changes.

End of exercise

Exercise review and wrap-up

In this exercise, you worked with IBM Integration Designer and explored its capabilities and preferences. You used the Installation Manager to find product updates. You also browsed through several IBM Integration Designer graphical editors, views, and perspectives.

Exercise 3. Exploring IBM Integration Designer, part II

What this exercise is about

In this exercise, you use the IBM Integration Designer environment to explore various SCA components and examine them in the test environment.

What you should be able to do

After completing this exercise, you should be able to:

- Assemble an SCA application
- Examine project components
- Explore staging projects
- Use the IBM Integration Designer test environment
- Enable cross-component tracing to examine a business process in testing
- Deploy a library globally

Introduction

To simplify and accelerate the development of applications, the IBM Integration Designer environment provides a layer of abstraction that separates the visually presented components from the underlying implementation. You can use IBM Integration Designer to easily assemble an SCA application visually and generate the underlying physical resources automatically. IBM Integration Designer also includes a robust set of test components and a fully functional runtime environment.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and the IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

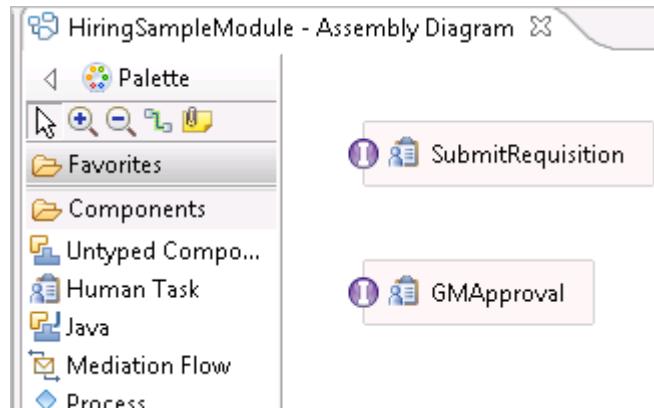
Exercise instructions

Part 1: Assemble an SCA application

In this portion of the exercise, you add SCA components to an assembly diagram and wire them together to form an application. You use a modified version of the “hiring sample” process application from the previous exercises.

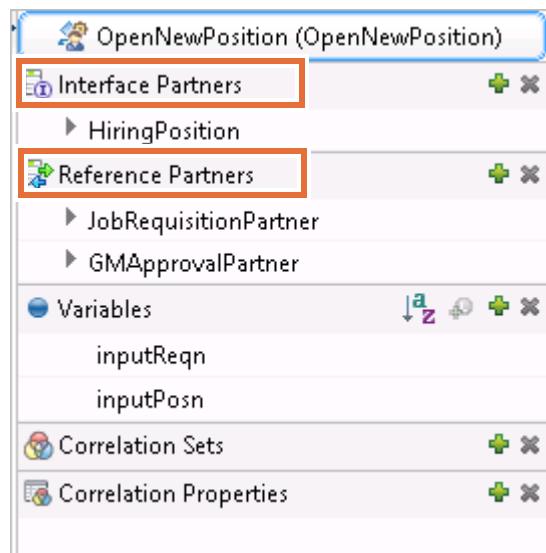
To assemble the loan application:

- 1. Open the Exercise 3 workspace.
 - a. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
 - b. Double-click the shortcut that is labeled **Exercise 3**. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of the Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
 - c. If the **Getting Started** tab is open, close it.
- 2. Assemble the SCA components.
 - a. In the Business Integration view, expand **HiringSampleModule** and double-click **Assembly Diagram**. Some SCA components are already added to the diagram: the **SubmitRequisition** and the **GMAApproval** human tasks, for example.

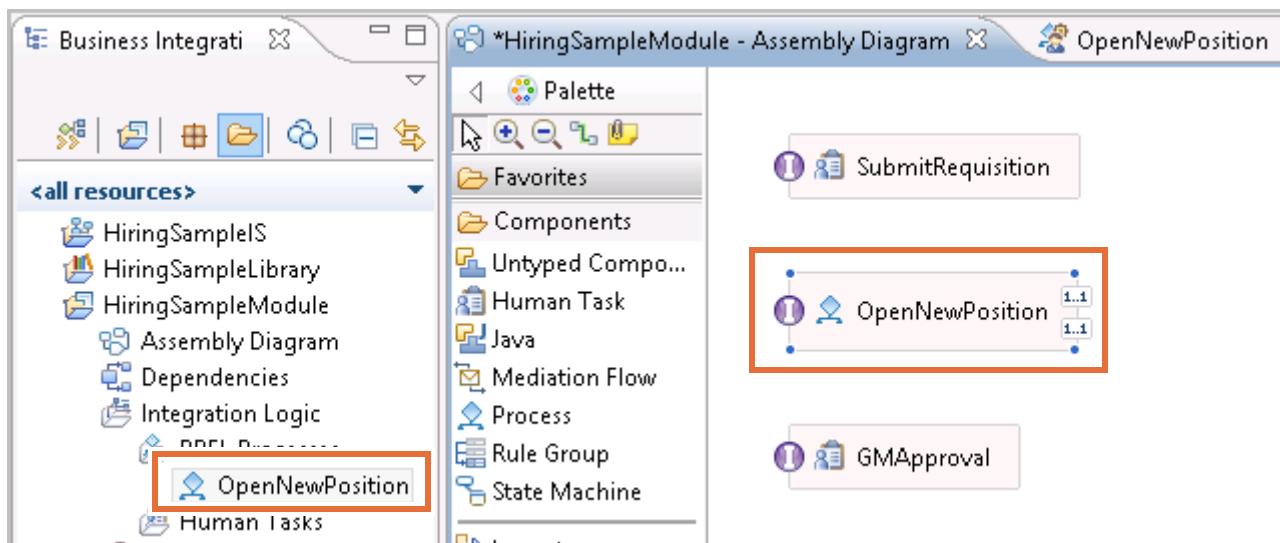


- b. In the Business Integration view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.
- c. Double-click **OpenNewPosition** to open it in the process editor.

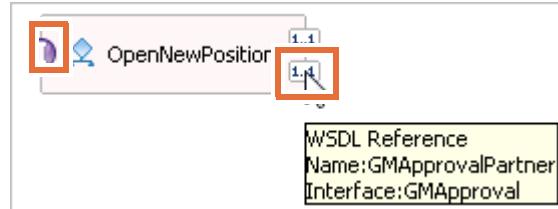
- ___ d. This process is a series of activities that are designed to work through the business logic that you tested in the previous exercises. In the **tray** to the right, note the **Interface Partners** and **Reference Partners**. The interface partner is used to invoke this process. The reference partners represent the interfaces of the services that this process invokes.



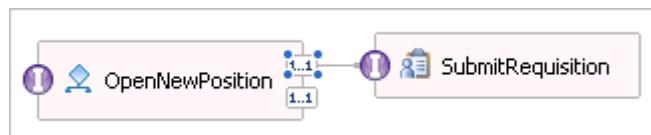
- ___ e. Switch to the **HiringSampleModule** assembly diagram.
 ___ f. Drag the **OpenNewPosition** process onto the **HiringSampleModule** assembly diagram.



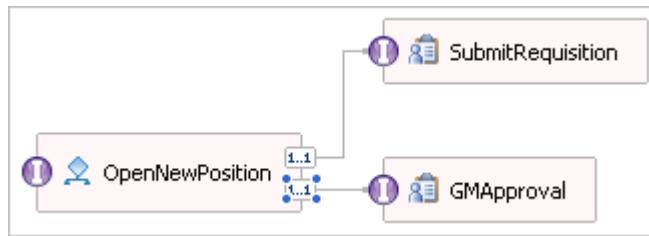
- ___ g. When you drag the process onto the assembly diagram, it is represented as an SCA component. The interface partner becomes the interface of the component. The reference partners become references on the SCA component. If you hover the mouse over the interface or references, a dialog box provides details. The BPEL process that you examined previously becomes the implementation behind the component and is represented as the implementation icon.



- ___ h. The cardinality of both references is 1..1. This cardinality indicates that the reference expects that one (and only one) service is wired to it. The process activity that invokes the service that is represented as the reference partner invokes only one possible service. Hover over the references until you locate the **JobRequisitionPartner** reference. When you locate it, right-click it and click **Wire to Existing** from the menu. This action wires the reference to the **SubmitRequisition** component.



- ___ i. Right-click the **GMAccrualPartner** reference and click **Wire to Existing** from the menu. The assembly of the application is complete.



- ___ 3. Press Ctrl+S to save your changes.

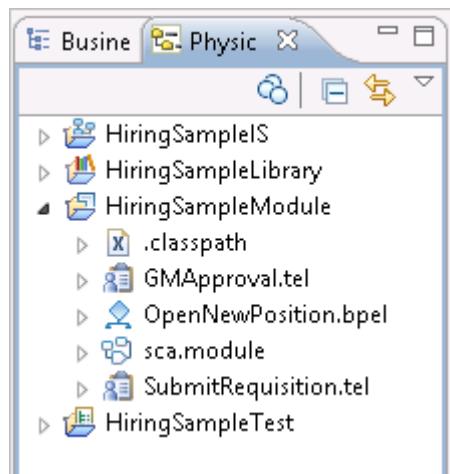
Part 2: Examine project components

In this portion of the lab, you examine some of the physical components that IBM Integration Designer generates for the visual components with which you interact.

To examine the physical components of a project:

- ___ 1. Click **Window > Show View > Physical Resources** from the menu options.

— 2. Expand **HiringSampleModule**.



The **Physical Resources** view shows you the files that IBM Integration Designer generates when you create a visual component in one of the editors. For example, the `.tel` file contains the human task rules, the `.bpel` file contains the business process, and the `.module` file contains the module assembly.

— 3. When you are done exploring, close the **Physical Resources** view.



Warning

Although it is possible to use the physical resources view to modify project components, it is intended for advanced, experienced users only. Manually changing any of the data in an `sca.module` file, or directly in a `.bpel` file, can result in unpredictable consequences.

Part 3: Explore staging projects

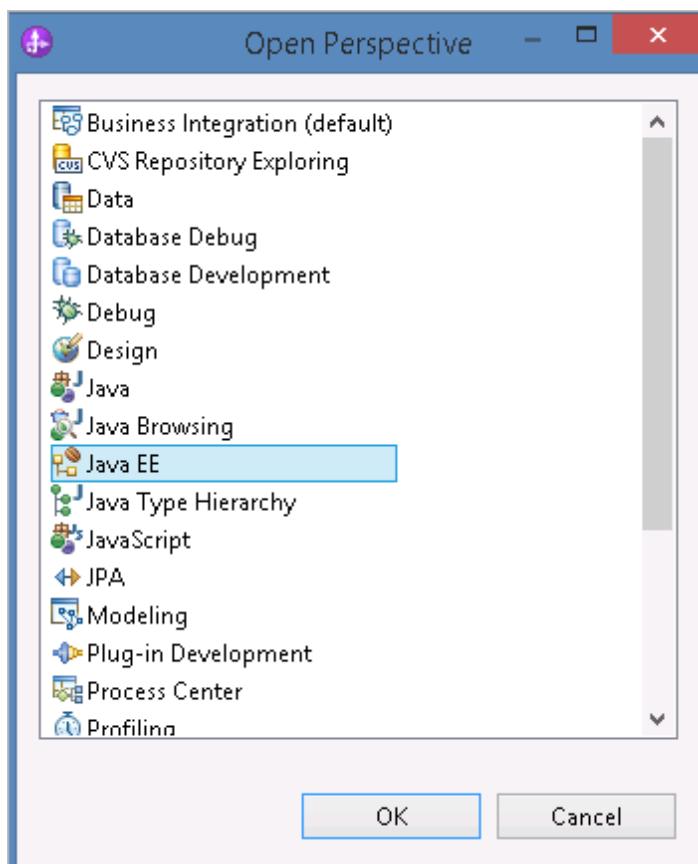
In this portion of the exercise, you examine the staging projects that IBM Integration Designer builds during automatic (or manual) builds. For any module project, two Java EE staging projects exist: an enterprise application project and a dynamic web project. If your module contains generic JMS bindings or JAX-RPC web service bindings, you also see an EJB project.

The enterprise application project can be exported as an EAR file for deployment. The enterprise application project consists of Java Platform, Enterprise Edition (Java EE) modules that can be deployed onto application servers. The modules are created from code artifacts such as web application archive (WAR) files, resource adapter archive (RAR) files, enterprise bean (EJB) JAR files, and application client archive (JAR) files. This packaging and configuring of code artifacts into enterprise archive (EAR) modules or stand-alone web modules is necessary for deploying the modules onto an application server.

A dynamic web project is a web module that contains servlets, JavaServer Page (JSP) files, and related code artifacts. After assembling a web module, you can install it as a stand-alone application or combine it with other modules into an enterprise application.

To examine staging projects:

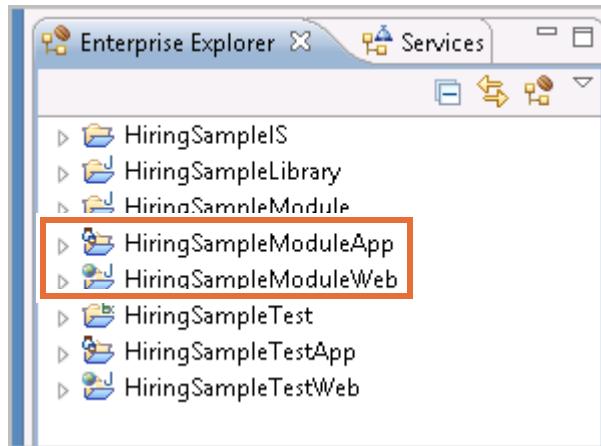
- 1. Switch to the **Java EE** perspective. Because the staging projects are platform-specific artifacts that are generated for the Java EE runtime, you must switch to the Java or Java EE perspective to view them.
 - a. Click **Window > Open Perspective > Other** from the menu options.
 - b. In the **Open Perspective** window, select **Java EE**.



- c. Click **OK**.

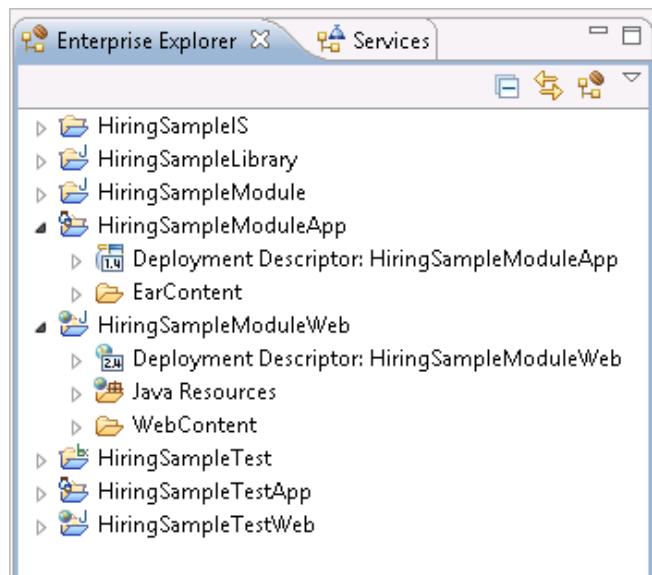
2. Examine the staging projects.

- a. In the Enterprise Explorer view, the **HiringSampleModule** has an enterprise application project and a dynamic web project. The project names are built based on the application name. Thus, the enterprise application project is named **HiringSampleModuleApp** and the dynamic web project is named **HiringSampleModuleWeb**.

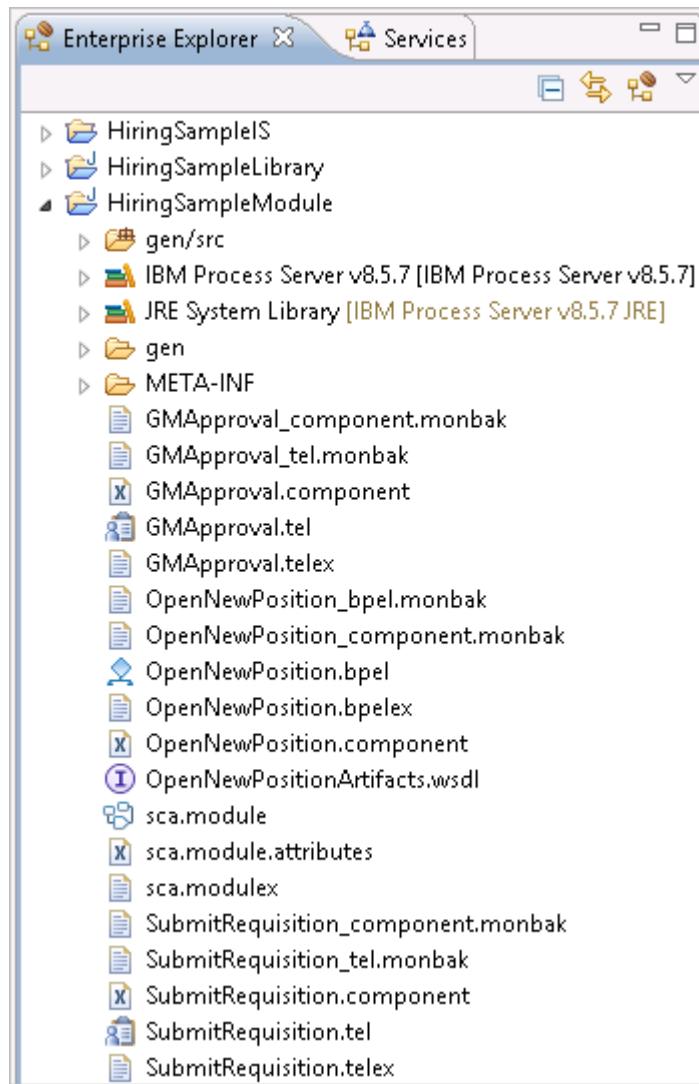


When you deploy projects to the test environment, you are deploying the applications that are represented here. These project names are displayed in the **Add and Remove** dialog box when you deploy to the server.

- b. Expand **HiringSampleModuleApp** and **HiringSampleModuleWeb** and examine the artifacts.



- ___ c. If time permits, expand **HiringSampleModule** and examine the other Java EE artifacts in the view.



- ___ d. When you are finished exploring, switch to the Business Integration perspective by clicking the icon in the upper right.

Part 4: Use the IBM Integration Designer test environment

In this portion of the exercise, you learn about the IBM Integration Designer test environment. In IBM Integration Designer, the integration test client is the designated tool for testing modules and components. You can use the sophisticated user interface of the test client to easily manage and precisely control your tests. The testing is generally done on the interface operations of your components. Additionally, you can determine whether the components are correctly implemented and the references are correctly wired.

Using the integration test client, you can test:

- An individual module
- A set of interacting modules
- An individual component
- A set of interacting components

When you begin a test by invoking an operation, the integration test client automatically detects the deployment state of the modules that are going to be tested. If any of the modules are not deployed, the deployment location wizard automatically opens. This wizard can be used to select the server where you want to deploy the modules. If the server is not already running, it is started automatically.

Test configurations

Test configurations are used to control your tests. A test configuration specifies one or more modules to test. Each of these modules might include zero or more emulators for components or references in the module and zero or more monitors for the wires in the module. When you open the integration test client, a default test configuration is automatically created that you can immediately use for testing. The default test configuration is often all that you need for testing your modules and components. However, you can choose to edit and customize the default test configuration, or you can create and edit one or more test configurations.

Emulators

You can use the integration test client to emulate components and references in your modules. During a test, when control flows to an emulated component or reference, the integration test client intercepts the invocation and routes it to the associated emulator. Two types of emulators are available:

- Manual
- Programmatic

A manual emulator is an emulator for which you must specify runtime response values for an emulated component or reference. If you are testing an entire module, the default test configuration contains manual emulators for all unimplemented components and unwired references. However, suppose that you are testing a set of components or an individual component within a module. In this case, the default test configuration contains manual emulators for any other components that were not selected for testing regardless of whether they are implemented or not. Although manual emulators are added by default, you can remove the manual emulators or redefine them as programmatic emulators.

When a manual emulator is encountered during a test, a manual emulate event is generated. When the test pauses, you can manually specify some output parameter values or throw an exception for the emulated components or references. By comparison, when a programmatic emulator is encountered during a test, a programmatic emulate event is generated. A Java program that is contained in a visual snippet or a Java snippet automatically provides the output parameter values or exceptions.

Monitors

When the integration test client generates a default test configuration or you add a test configuration, monitors are automatically added for any component wires and exports that are found in the modules of the test configuration. When you invoke an operation and run a test, these monitors listen for any requests and responses that flow over the wires and exports. If a request is detected, a request event is generated. If a response is detected, a response event is generated. These events show parameter data that flows across the wires. They are added to the test trace of events that are displayed in the Events area of the integration test client.

Although monitors are automatically added for the wires and components of your test configuration modules, you can edit the monitors and change whether they monitor requests, responses, or both. You can also remove the monitors or add more monitors as required.

Testing an individual component

In this section, you test on an individual component in the hiring sample application. You examine the emulators and monitors that the integration test client automatically configures, and you learn how to save test data.

Three different use cases in the hiring sample application are:

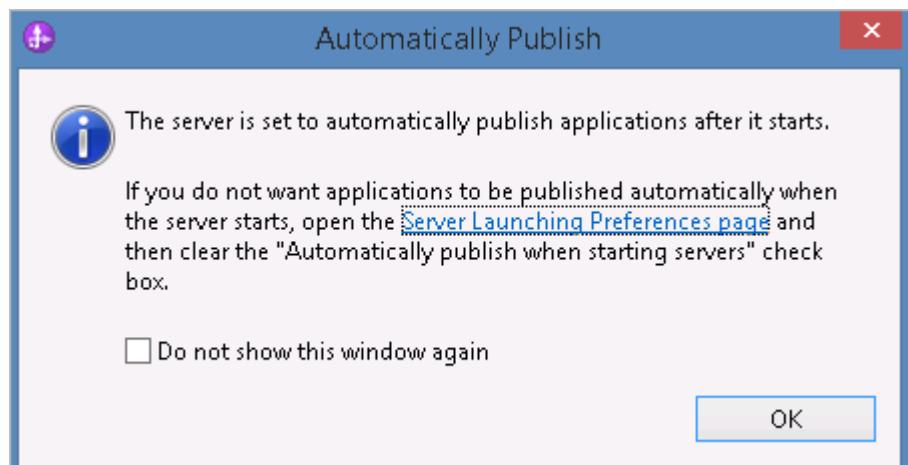
- The hiring requisition is for a new job, which does not need approval.
- The hiring requisition is for an existing job, which needs approval from the general manager.
- The general manager might approve or reject requisitions for existing job actions.

In the component test, you verify the function of the first use case where a requisition for a new job is automatically approved.

To run an individual component test:

- 1. Start IBM Process Server.
 - a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu. You can also select the server and click the **Start the server** icon.
 - b. Wait for the server to complete the startup procedure. The procedure is complete when you see the message `Server server1 is open for e-business` in the **Server Logs** view. The server status also changes to `[Started, Synchronized]` in the **Servers** view.

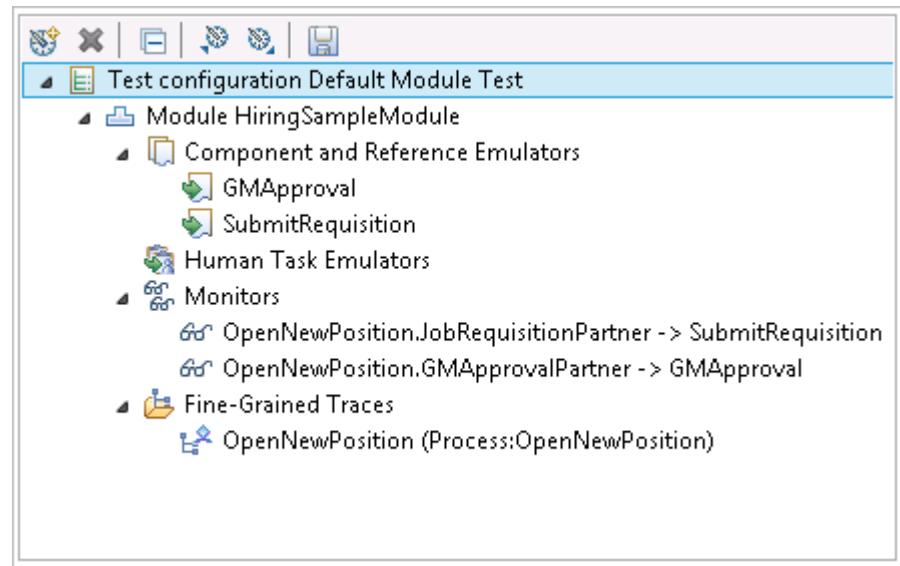
- ___ c. Click **OK** when the message is displayed that the server is set to automatically publish applications.



- ___ 2. Use the **Test Component in Isolation** option to test the **OpenNewPosition** component.
- ___ a. In the Business Integration view, expand **HiringSampleModule**.
- ___ b. Double-click **Assembly Diagram** to open the assembly editor.
- ___ c. On the assembly editor canvas, right-click the **OpenNewPosition** SCA component and click **Test Component in Isolation** from the menu. The **OpenNewPosition** component contains the process that determines whether a job requisition can be automatically approved.
- ___ d. In Integration Test Client, click the **Configurations** tab in the lower-left corner of the window.



- ___ e. Examine the emulators and monitors.



- ___ f. Switch back to the **Events** tab (in the lower-left corner of the test window) and browse to the **Initial request parameters** table on the right side of the **Detailed Properties** section.

Initial request parameters:

Value editor XML editor

Name	Type
inputReqn	Requisition
reqNum	string
requestor	string
status	string
approvalNeeded	boolean
date	date

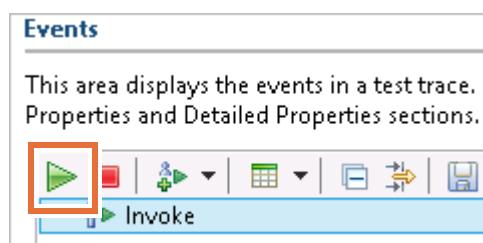
- __ g. Enter the following information in the **Value** column. To aid data entry, you can use the maximize icon to maximize the input window to full screen.
- inputReqn
 - reqNum: 001
 - department: Sales
 - location: Atlanta
 - inputPosn
 - positionType: New
 - lastName: Smith
 - firstName: John

Initial request parameters

Name	Type	Value
inputReqn	Requisition	[] 001
reqNum	string	[] 001
requestor	string	[]
status	string	[]
approvalNeeded	boolean	[] false
date	date	[] 2016-03-22
department	string	[] Sales
location	string	[] Atlanta
empNum	int	[] 0
gmApproval	boolean	[] false
gmComments	string	[]
inputPosn	Position	[]
positionType	string	[] New
replacement	Person	[]
lastName	string	[] Smith
firstName	string	[] John
supervisor	string	[]
startDate	date	[] 2016-03-22
payLevel	string	[]
payType	string	[]
notes	string	[]
jobTitle	string	[]

To edit values, start typing or press F2.

- __ h. Click **Continue** on the Events toolbar.



**Information**

If you are testing a component for the first time, several actions occur before you are able to test:

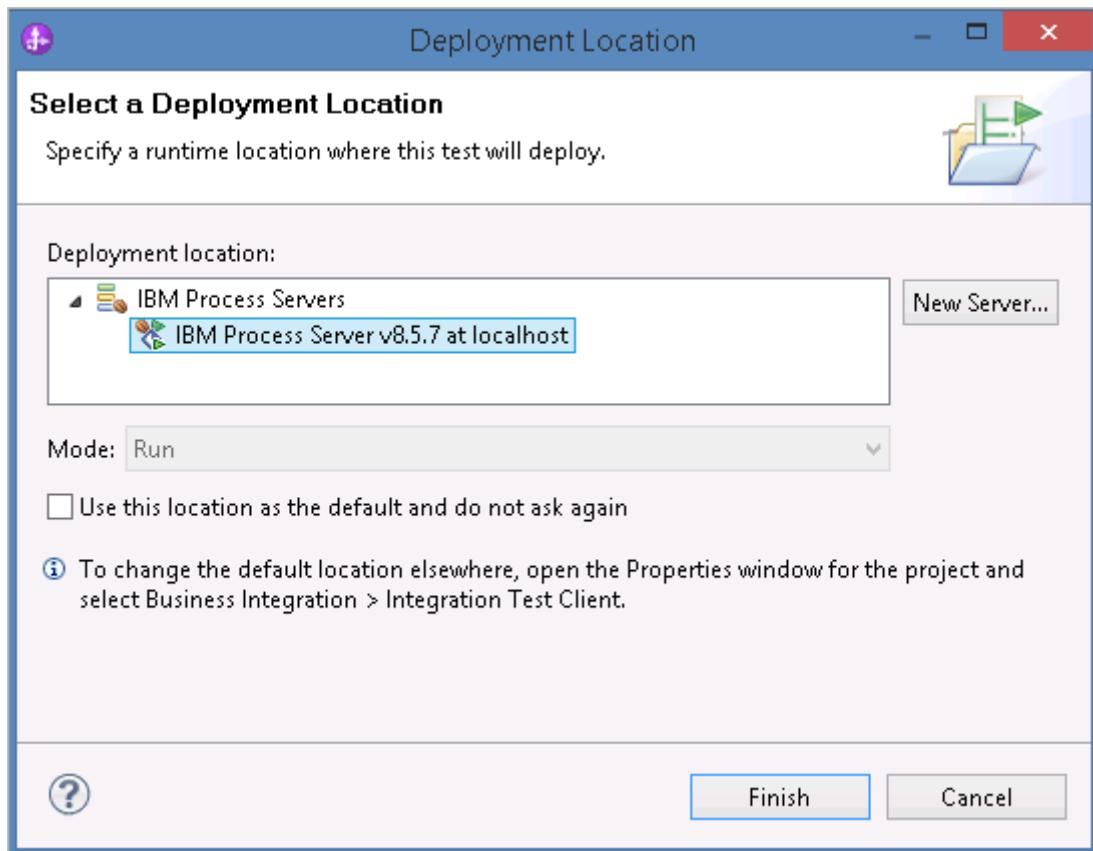
- 1) The IBM Process Server test environment starts, unless it is already started.
- 2) The **HiringSampleModule** is published to the server. You might need to specify a deployment environment.
- 3) The **HiringSampleModule** is started.

This procedure might take several minutes to complete. You might see a message that the module is going to be automatically published to the server. Select **Do not show this message again** and click **OK**.

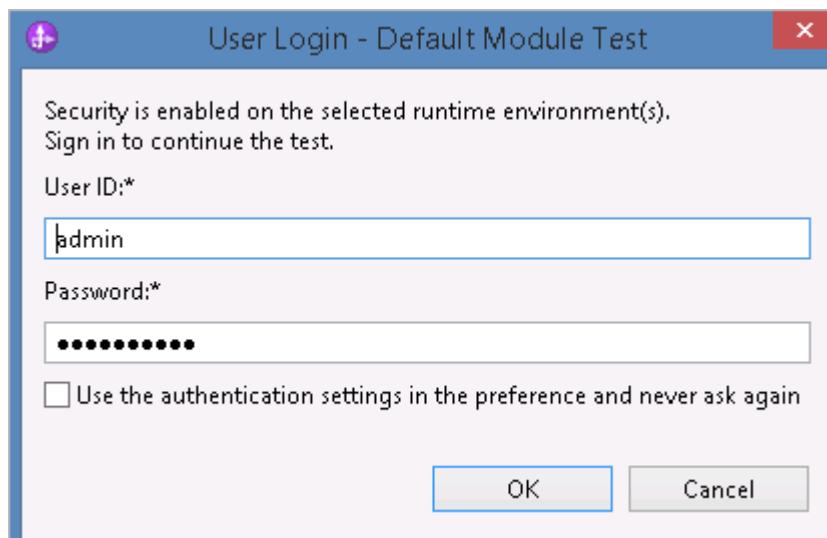
**Note**

If you maximized the input window, minimize it to locate the Events toolbar.

- i. When the Deployment Location dialog box is displayed, select **IBM Process Server v8.5.7 at localhost**.

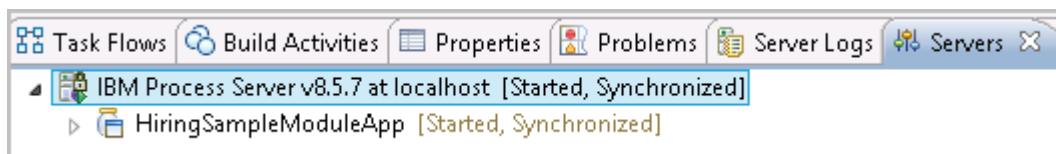


- __ j. Accept the remaining default options and click **Finish**.
- __ k. In the **User Login - Default Module Test** dialog box, accept the default options and click **OK**. By default, the **User ID** is set to `admin` and **Password** is set to `websphere` during IBM Integration Designer installation.

**Note**

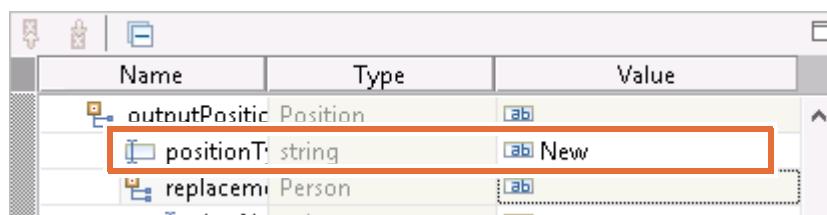
You are not selecting the **Use this location as the default** and **Use the authentication settings in the preference** check boxes because they are specific to the individual workspace. Each of your exercises has a specific workspace, so you would select these options for each exercise.

- __ l. Switch to the **Servers** view and expand **IBM Process Server v8.5.7 at localhost**. You can see the state of the modules (and the server) in the **Servers** view when the module is deployed.

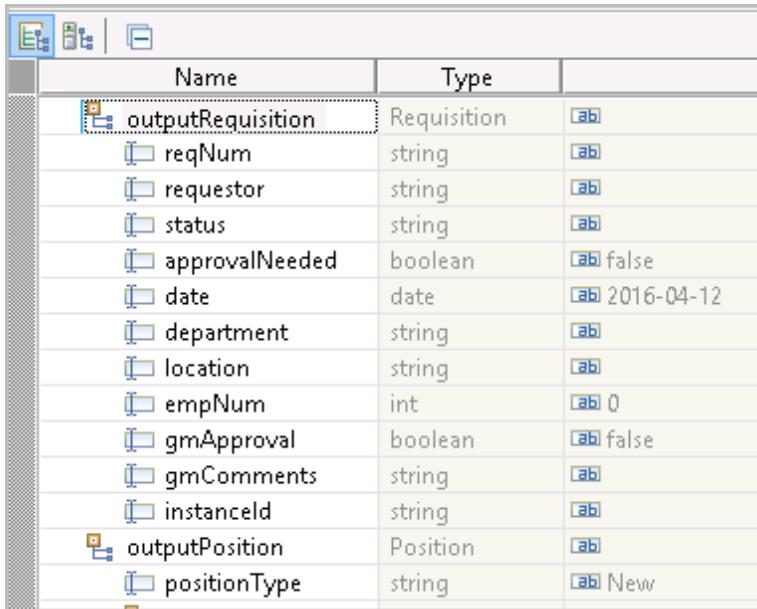


- __ m. The test stops at the human task emulator **SubmitRequisition**. As you saw in a previous exercise, a client GUI is displayed for the user to enter requisition details. For this test, the most important detail is to set the job position to `New`.

Set the **outputPosition > positionType** value to: `New`



- __ n. Click **Continue** again.
- __ o. When the test is complete, select the **Response (OpenNewPosition <SubmitRequisition:createRequisition)** event.
- __ p. Examine the **Response parameters**.

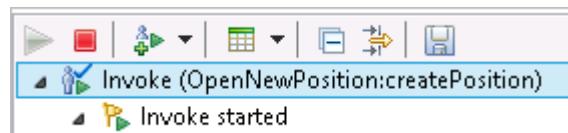


The screenshot shows a table titled "Name" and "Type" with the following data:

Name	Type
outputRequisition	Requisition
reqNum	string
requestor	string
status	string
approvalNeeded	boolean
date	date
department	string
location	string
empNum	int
gmApproval	boolean
gmComments	string
instanceId	string
outputPosition	Position
positionType	string

Because the job requisition is new, no approval is needed, so the **approvalNeeded** and **gmApproval** fields have both been set to `false`.

- __ 3. Save the test data to the data pool to reduce manual input in subsequent tests.
 - __ a. Select the top-most **Invoke** event in the **Events** pane, **Invoke (OpenNewPosition:createPosition)**.



- __ b. In the **Initial Request Parameters** section, right-click **inputPosn** and click **Add Value to Pool** from the menu. This action starts the data pool wizard.



Information

A test suite, **HiringSampleTest**, is part of the project interchange file that you imported earlier. This test suite contains a *data pool*. A data pool is a collection of test data that you might reuse when testing components for this project.

- __ c. Expand **HiringSampleTest** and select **HiringRequisitionDatapool.objectpool**.
- __ d. Click **Finish**.
- __ e. Set the name to: `NewPosition`

-
- ___ f. Accept the remaining default options and click **OK**. The next time that you run a test, you can then retrieve the data from the data pool by right-clicking the input parameter and clicking **Use Value from Pool** from the menu.
 - ___ g. Close the **OpenNewPosition_Test** tab and click **No** when you are prompted to save the test trace. The module remains deployed until you manually remove it from the server.

Testing an application

In this portion of the exercise, you test the hiring sample application. You test the application by deploying the modules to the IBM Process Server runtime and by interacting with the BPEL Process Choreographer Explorer user interfaces. Three different use cases in your application are:

- The hiring requisition is for a new job, which does not need approval.
- The hiring requisition is for an existing job, which needs approval from the general manager.
- The general manager might approve or reject requisitions for existing job actions.

Examining the test server profile

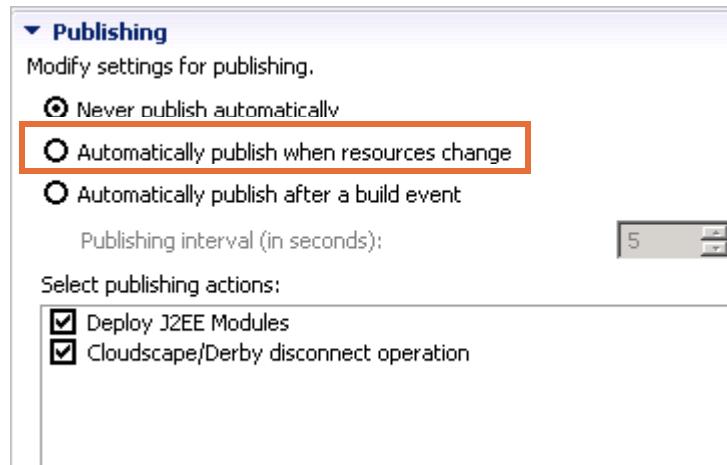
The IBM Process Server runtime that is installed in IBM Integration Designer has a default server profile. You can alter this profile, create profiles, and reset your profile back to the original state. Using profiles, you can deploy and test modules on a remote IBM Process Server or WebSphere Enterprise Server Bus runtime environment. For example, you might want to deploy your application to the preproduction environment of your organization. For more information about creating a server profile, see the product documentation.

To view the configuration of the IBM Process Server v8.5 test runtime:

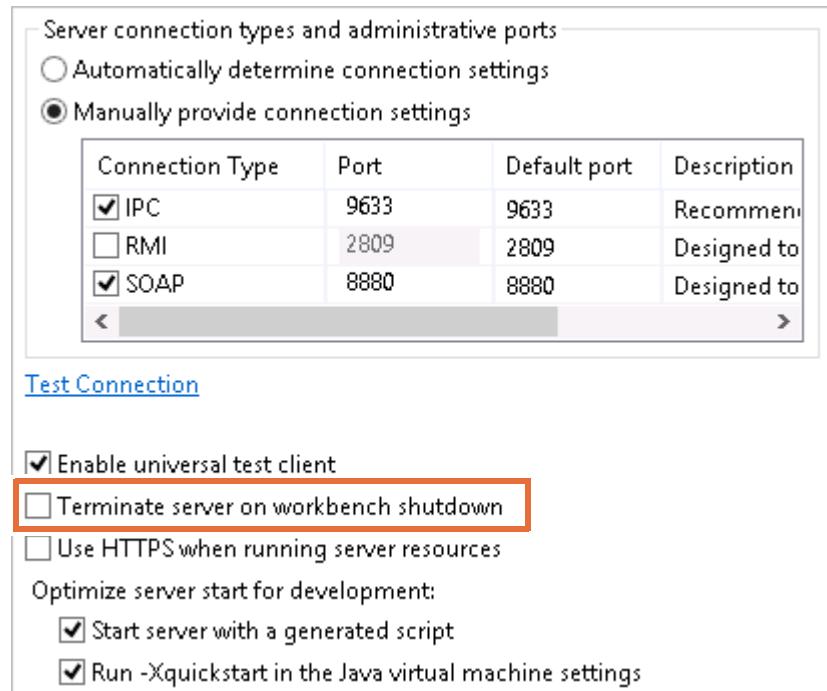
- ___ 1. Switch to the **Servers** view.
- ___ 2. Right-click **IBM Process Server v8.5.7 at localhost** and click **Open** from the menu.

3. Examine (but do not change) the server properties in the server configuration editor. In particular, note the following options.

- **Automatically publish when resources change:** This option in the **Publishing** section automatically republishes modules to the local test server if you modify them while they are deployed.



- **Terminate server on workbench shutdown:** This option in the **Server** section stops the test server when you close IBM Integration Designer.



4. After examining the server properties, close the server configuration editor.

Testing the hiring sample application

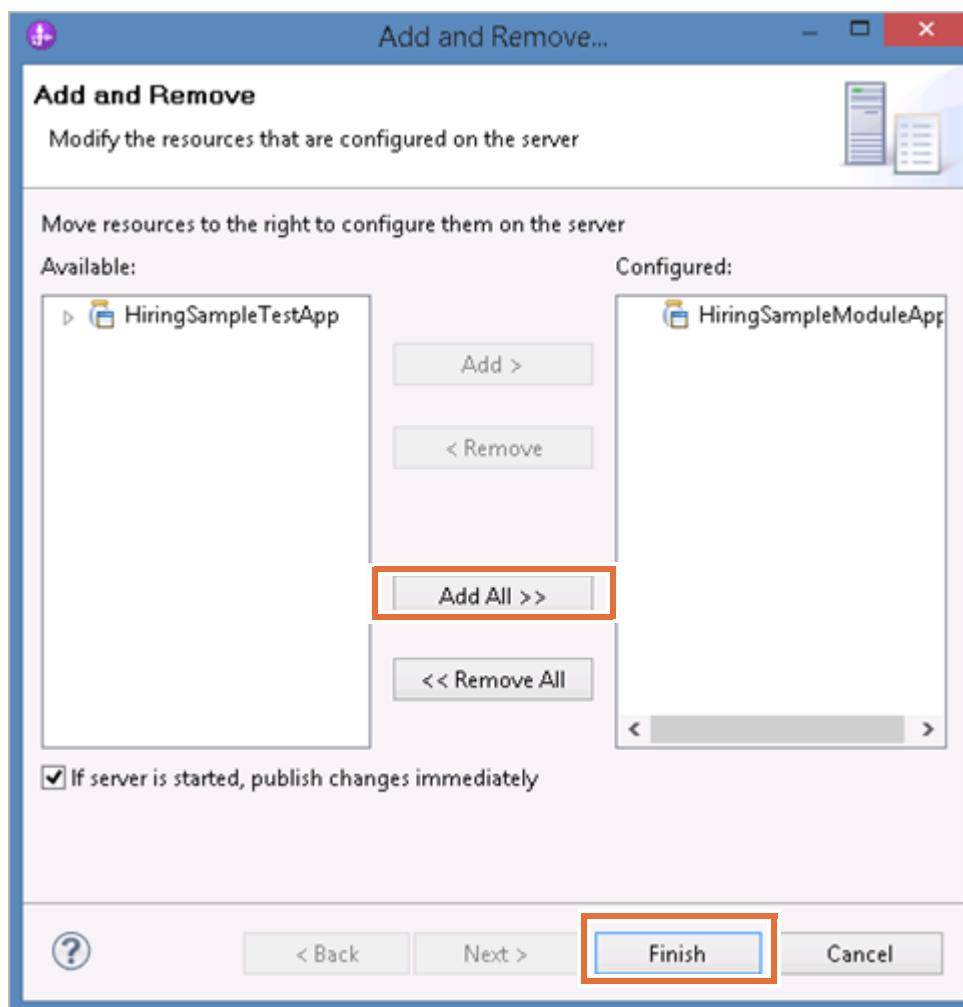
You test the hiring sample application by interacting with it through a web-based tool that is called the Business Process Choreographer Explorer. This tool is included with IBM Integration Designer

for working with process applications and human tasks. To test the application, you must deploy the application to the IBM Process Server test environment and enter test data in the user interface.

Deploying the hiring sample application

To deploy the hiring sample application to the test server:

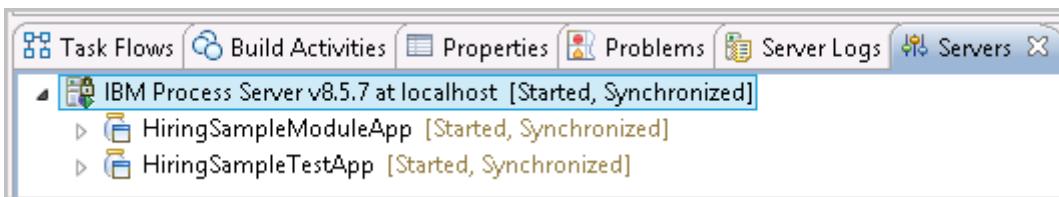
- 1. If necessary, start IBM Process Server. The server was started automatically in the previous section. If it is already started, then go to step 2.
 - a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu. You can also select the server and click the **Start the server** icon.
 - b. Wait for the server to complete the startup procedure. The procedure is complete when you see the message `Server server1 is open for e-business` in the **Server Logs** view. The server status also changes to `[Started, Synchronized]` in the **Servers** view.
- 2. Add the projects.
 - a. Right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.
 - b. Click **Add All** and then click **Finish**.



**Note**

In IBM Integration Designer, you can add projects to the IBM Process Server runtime environment without first starting the server. If you do so, when the server is started, the projects are automatically deployed and started. Similarly, if you remove projects while the server is stopped, the projects are removed the next time that the server starts. These options are valid only if the test server is local.

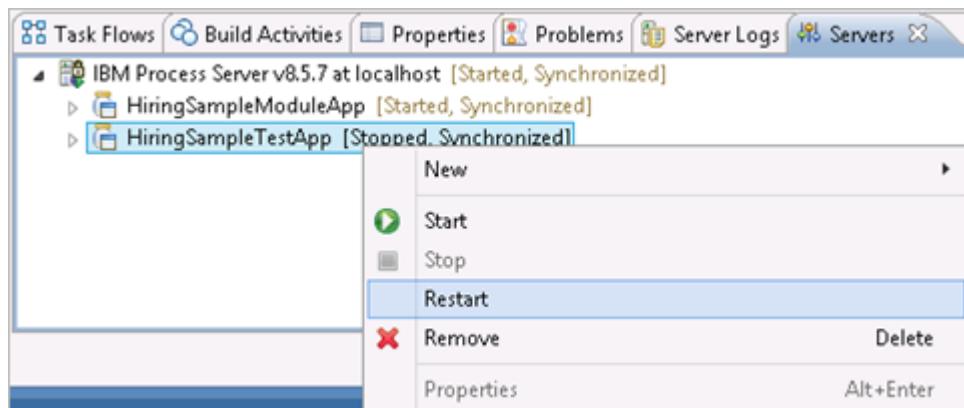
- __ c. Wait for the publishing action to complete and for the modules to start. Publishing is complete when no messages are in the status bar such as: Publishing to IBM Process Server v8.5.7 at localhost.
- __ d. When the projects are deployed and started, switch to the **Servers** view, expand **IBM Process Server v8.5.7 at localhost**, and view the modules that are currently deployed to the server. Verify that the status of all the modules added is **Started**.

**Troubleshooting**

In the version of IBM Business Process Manager Advanced V8.5.7 that was used to test these exercises, the status of the server and modules occasionally does not automatically refresh. It is possible that, while the server and modules are started, the status remains **Publishing**.

Press F5 to refresh the view.

Also, notice that often a module that is added has a **Stopped** status. To change its status, right-click the module, click **Restart** from the menu, and republish the module if prompted. Repeat the step if necessary to ensure that the module starts.



Opening the customer user interface

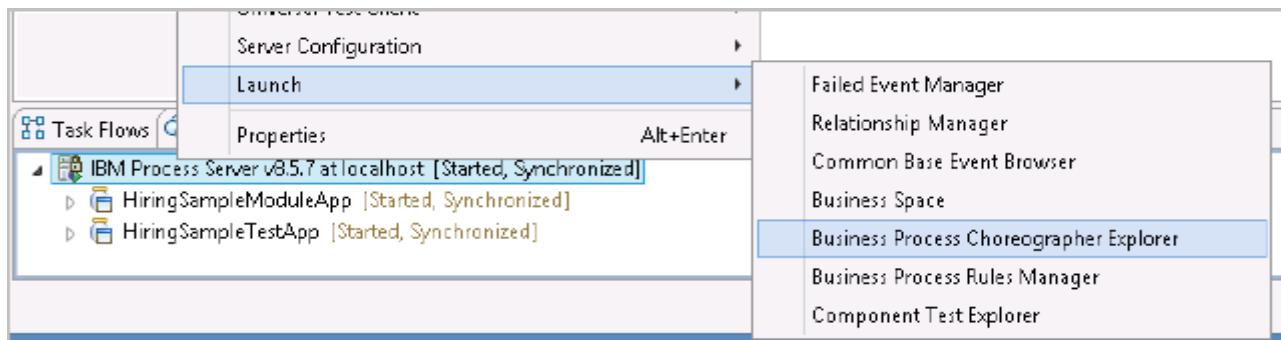


Note

For BPM on Cloud users:

Click **Launch** in the Business Process Choreographer Explorer tile to use this tool.

- 1. Open the Business Process Choreographer Explorer user interface and create an application:
 - a. In the **Servers** view, right-click the server and click **Launch > Business Process Choreographer Explorer**.



- b. If you receive security alert messages, click **Yes** each time.



Note

Alternatively, you can open a browser and type:

`http://localhost:<portnumber>/bpc`

The port number can be found in the console output. For a local IBM Process Server instance configured during IBM Integration Designer product installation, the port number of the internal web container is typically 9080.

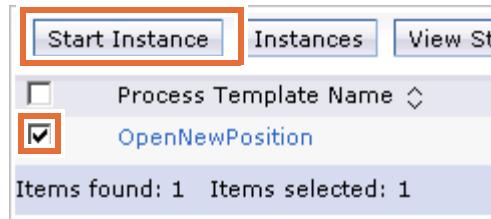
- c. Use `admin` for the user name and `web1sphere` for the password to log in to the tool.
- d. The page automatically begins at the **My To-dos** list. This list contains the human tasks that are waiting for user input. It is expected that no tasks are waiting.

Configurable menus on the left of the page work either with processes or with human tasks.

- __ e. From the **Process Templates** menu, click the **Currently Valid** link.



- __ f. Select the **OpenNewPosition** template and click **Start Instance**.



- __ g. In the **Process Input Message** section, set the **Process Name** to: `NewPositionTest`

- __ h. Click **Submit**.

Process Template Name	OpenNewPosition
Process Description	
Operation	createPosition
Process Name	NewPositionTest
Process Input Message	

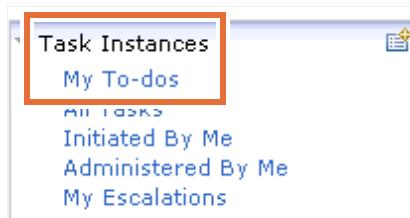


Note

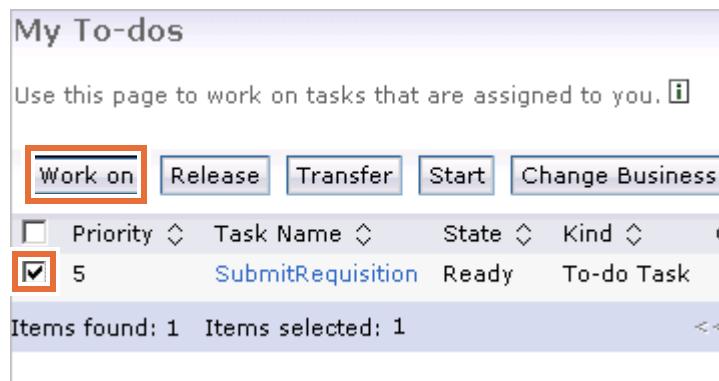
You do not have to complete any other data. It is accomplished in a later step.

- __ i. The first activity for the user is to complete the job requisition. This job requisition is implemented as a human task by the process application. The Business Process Choreographer Explorer works with processes and human tasks.

From the left side of the explorer, from the **Task Instances** menu, click **My To-dos**.



- __ j. The instance of the process application stops at the human task: **SubmitRequisition**. Select **SubmitRequisition**; then, click **Work on**.



- __ k. The **Task Message** display is divided into two sections: the input message and the output message. The input message is empty; you did not enter any information for it in the previous step.

Type the following information for the output message:

- **outputPosition > positionType**: New

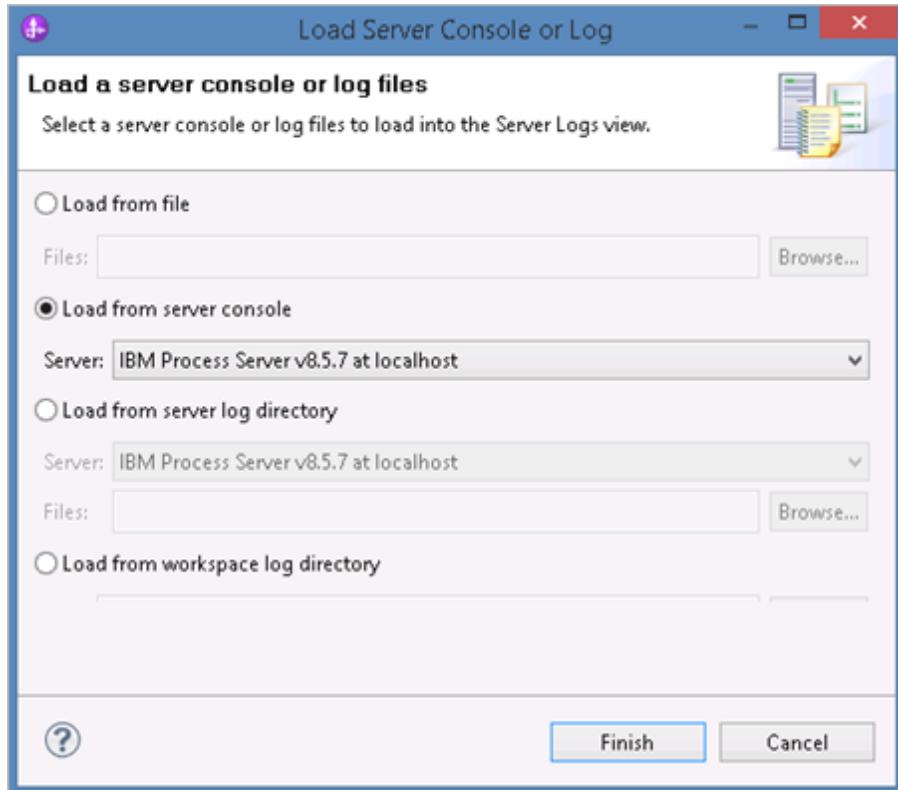
The screenshot shows a configuration interface for a task message. On the left, there's a vertical list with 'outputPosition' highlighted. To the right, there are several fields: 'positionType' is set to 'New' (highlighted with a red box); 'replacement' has a 'Add' button; 'jobTitle' and 'iId' both have empty input fields.

- __ l. Click **Complete**. Expect to see that no tasks are waiting for you. If necessary, scroll up to locate the **Complete** button.

- __ 2. Examine the **Server Logs** view.
__ a. If you do not see any messages, click the **Load Server Console or Log** icon.



- __ b. Select **Load from server console** and click **Finish**.



- __ c. Examine the server log. You see a message that the requisition was automatically approved because the position was new.

Type	Time	Thread ID	Contents
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:16.204 EDT	00000271	WSVR0221I: Application started: HiringSampleModuleApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:20.204 EDT	00000056	WSVR0200I: Starting application: TestController70
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:20.532 EDT	00000056	WSVR0221I: Application started: TestController70
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:55.720 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed
<input type="checkbox"/> Log message	Apr 15, 2016 14:04:47.673 EDT	0000024b	WSVR0200I: Starting application: HiringSampleTestApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:04:48.173 EDT	0000024b	WSVR0221I: Application started: HiringSampleTestApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:10:59.986 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed

Seeking manager approval for an existing job position

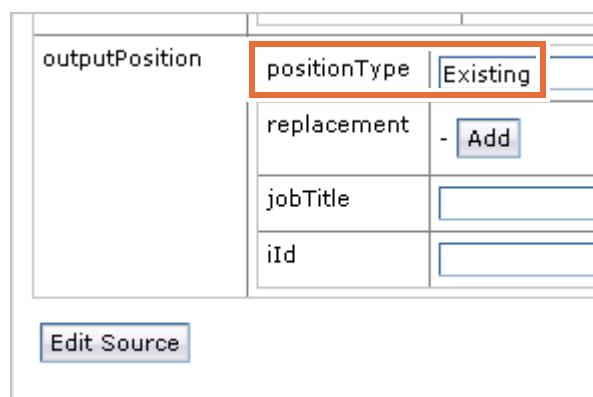
In the previous example, the job requisition was automatically approved because the position was listed as `New`. In this example, set the position to `Existing`, which forwards the job requisition to the general manager for approval.

- __ 1. Return to the Business Process Choreographer Explorer.
- __ 2. Select **Process Templates > Currently Valid**.
- __ 3. Select **OpenNewPosition** and click **Start Instance**.
- __ 4. Set the **Process Name** to `ExistingJobA` and click **Submit** to create an instance.

- 5. Select **OpenNewPosition** and click **Start Instance** again. Set the **Process Name** to **ExistingJobB** and click **Submit** to create a second instance with a different process ID. In this way, you can test two samples at the same time: one that the general manager approves, and one that is rejected.
- 6. In the **Currently Valid Process Templates** view, select **OpenNewPosition** and click **Instances**. Both of your instances are active and running.



- 7. Select **Task Instances > My To-dos**. Both instances stop at the same activity: **SubmitRequisition**. Select the first instance and click **Work on**.
- 8. Set **outputPosition > positionType** to: **Existing**. Click **Complete**.

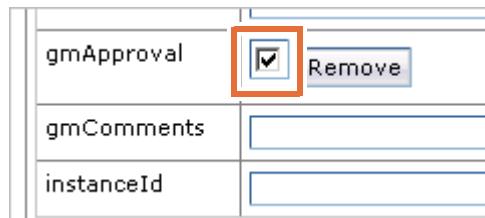


- 9. Refresh the **MyTo-dos** page by clicking **Refresh**. One process is still at the **SubmitRequisition** activity, but the one you just completed moves forward to **GMAccrual**.



- 10. Select the task instance at the **SubmitRequisition** activity and click **Work on**.
- 11. Set **outputPosition > positionType** to: **Existing**. Click **Complete**. When you refresh again, both tasks arrive at the **GMAccrual** activity.
- 12. Select the first task that is named **GMAccrual** and click **Work on**. Because the position was listed as **Existing**, the **approvalNeeded** field is enabled.

- ___ 13. In the output message, next to the **gmApproval** field, click **Add**. A check box is added for you to toggle.



- ___ 14. Leave the check box selected and click **Complete** to indicate that the general manager approved the job requisition.
- ___ 15. Examine the **Server Logs** view. A message indicates that the general manager approved the job requisition.
- ___ 16. Return to the Business Process Choreographer Explorer. Select **My To-Dos**.
- ___ 17. Select the remaining task and click **Work on**.
- ___ 18. In the output message, next to the **gmApproval** field, click **Add**. A check box is added for you to toggle.
- ___ 19. Clear the **gmApproval** check box. This action signals that the general manager denied the job requisition. Click **Complete**.
- ___ 20. Examine the **Server Logs** view again. A message indicates that the general manager rejected the job requisition.
- ___ 21. Log out of the Business Process Choreographer Explorer and close the browser.

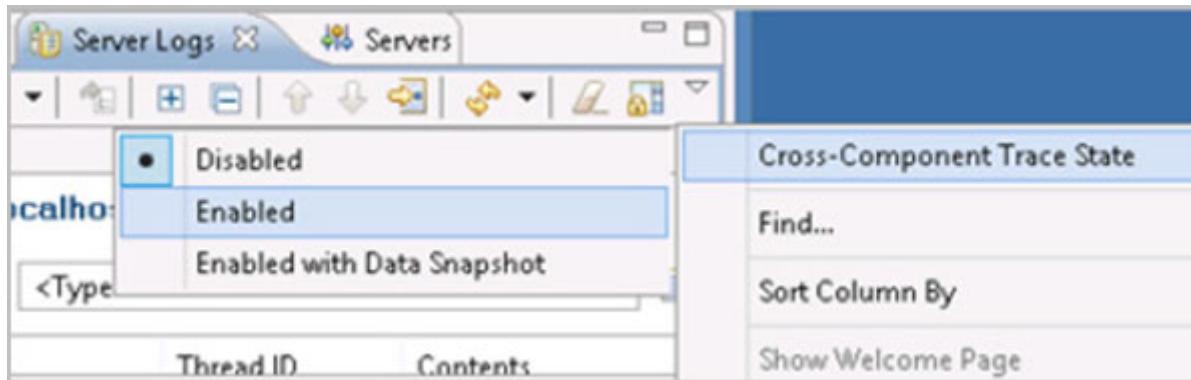
Part 5: Use cross-component testing to examine a running process application

Cross-component testing is a feature of the IBM Process Server test environment, which traces the individual components in a process application. Without the use of breakpoints or debugging, a developer can use cross-component tracing to follow the path of a running business process and examine its flow in the **Server Logs** view.

- ___ 1. Open the **Server Logs** view. In the **Server Logs** toolbar, an icon is named **View Menu**.



2. Click **View Menu** and click **Cross-Component Trace State > Enabled**.



Information

The three settings for cross-component tracing are:

- **Disabled**: Tracing is not sent to the server log.
- **Enabled**: Tracing is sent to the server log.
- **Enabled with Data Snapshot**: Tracing is sent to the server log along with capturing data that is sent in and passed between components.

3. Clear the contents of the server log by clicking the **Clear Server Console** icon.



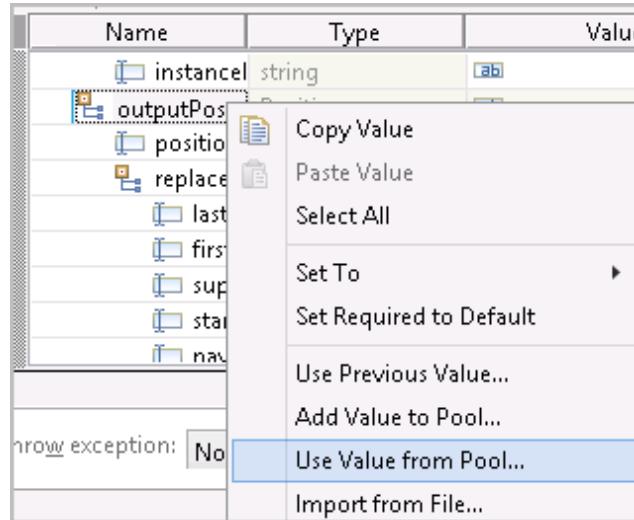
4. Open the **HiringSampleModule** assembly diagram.

- In the Business Integration view, expand **HiringSampleModule**.
- Double-click **Assembly Diagram** to open it in the editor.

5. Test the process application.

- Right-click the **OpenNewPosition** business process and click **Test Component in Isolation**.
- In the integration test client, click **Continue**.
- If you are prompted to do so, select **IBM Process Server v8.5.7 at localhost** from the list of deployment locations, and click **Finish**.
- If you are prompted to do so, enter `admin` as the User ID and `websphere` as the password.
- Click **OK**.
- The process stops at emulating the **SubmitRequisition** human task.

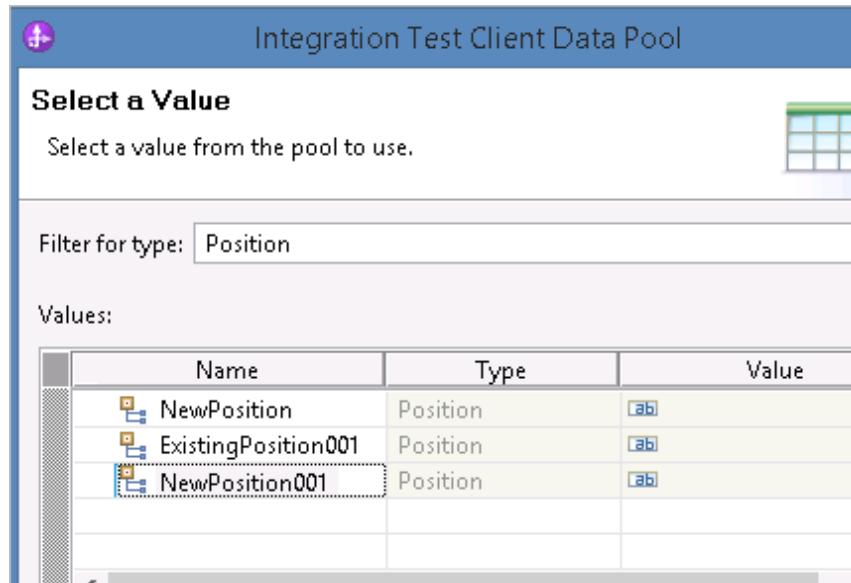
- __ g. Right-click the **outputPosition** parameter and click **Use Value from Pool**.



Important

Select the Position data object, not the Requisition data object.

- __ h. Options are available for you to select from the pool of test data. Select **NewPosition001** and click **OK**.



- __ i. Click **Continue** when the sample data is populated. The business process completes successfully.

- j. Examine the **Server Logs** view. You might choose to maximize the window so that it is easier to read.

Type	Time	Thread ID	Contents
Invocation sequence	Apr 15, 2016 14:52:43.632 EDT	00000247	
Start invoke (Op)	Apr 15, 2016 14:52:43.632 EDT	00000247	Start of the one-way invocation of operation OpenNewPos...
Start compo	Apr 15, 2016 14:52:43.710 EDT	00000059	Start of the component processing of operation OpenNew...
In BPEL	Apr 15, 2016 14:52:43.772 EDT	00000059	07e8a3dd-ddab-4640-be64-54345fc78d94 STATE 4fb376fb-...
Start BPEL	Apr 15, 2016 14:52:43.882 EDT	000002a2	Start of processing for BPEL process OpenNewPosition:_Pl...
End BPEL	Apr 15, 2016 14:52:43.929 EDT	000002a2	End of processing for BPEL process OpenNewPosition:_Pl:9...
Start BPEL	Apr 15, 2016 14:53:42.773 EDT	000002a2	Start of processing for BPEL process OpenNewPosition:_Pl...
Log	Apr 15, 2016 14:53:42.788 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed
End BPEL	Apr 15, 2016 14:53:42.819 EDT	000002a2	End of processing for BPEL process OpenNewPosition:_Pl:9...
End compo	Apr 15, 2016 14:52:43.851 EDT	00000059	End of the component processing of operation OpenNewP...
End invoke (Op)	Apr 15, 2016 14:52:43.663 EDT	00000247	End of the one-way invocation of operation OpenNewPos...

Cross-component tracing writes a trace to the Server Logs view, which displays the entire invocation sequence, from the time the SCA component was invoked at the top level to its BPEL implementation. Inside the BPEL process, you can follow how the BPEL process is calling each of the steps in turn. As each step completes, the invoking component that wraps it is closed in turn.



Hint

Cross-component tracing can be a valuable assistant when troubleshooting lab exercises. You might choose to leave cross-component tracing enabled for the labs that follow so that you can easily follow the flow of the business processes.

- 6. Close the integration test client. Do not save it.
- 7. Remove the applications from the server.
 - a. In the **Servers view**, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.
 - b. Click **Remove All** and click **Finish** to remove the projects from the server. The status bar indicates when the publish (removal) action is complete. The projects are also removed from the **Servers** view.

Part 6: Deploy a library globally

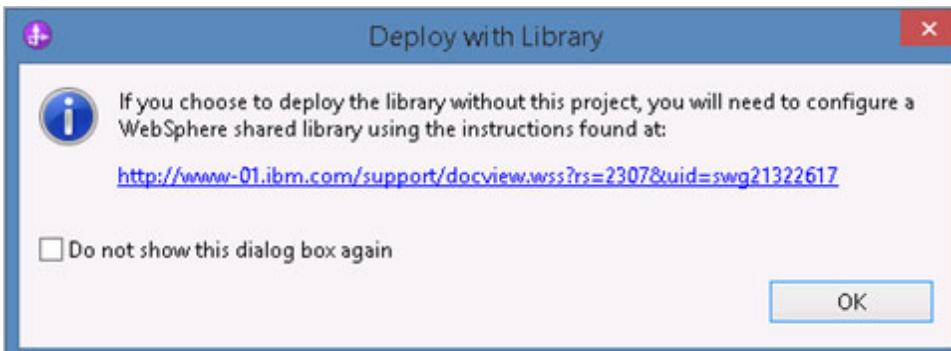
Two methods are available for deploying an SCA library. Libraries can be deployed globally (by reference) or with the dependent module (by copy). You make your choice in the dependency editor and configure the section “Sharing across runtime environments.” In this portion of the exercise, you enable the **HiringSampleLibrary** for global deployment as a WebSphere shared library.

The WebSphere shared library is a feature in the WebSphere Application Server product. It can be used to add resources to the server class loader hierarchy. This portion of the exercise

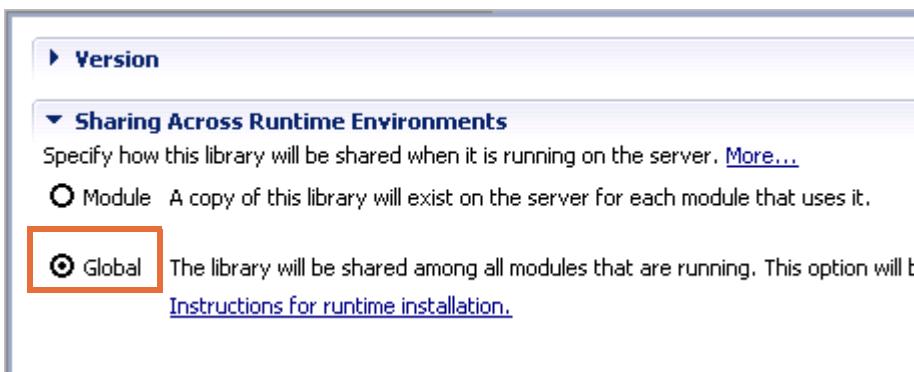
demonstrates how SCA libraries can be deployed for “by-reference” sharing so only one instance of the library exists in the runtime. The default deployment option is to package the library with each SCA module that references it.

To enable global deployment of an SCA library:

- ___ 1. Change the Sharing Across Runtime Environments setting for **HiringSampleLibrary** to Global.
 - ___ a. In the Business Integration view, expand **HiringSampleLibrary** and double-click **Dependencies**.
 - ___ b. Expand the **Sharing Across Runtime Environments** section.
 - ___ c. Click **Global**.
 - ___ d. When the **Deploy with Library** dialog box is displayed, click **OK** to close it.

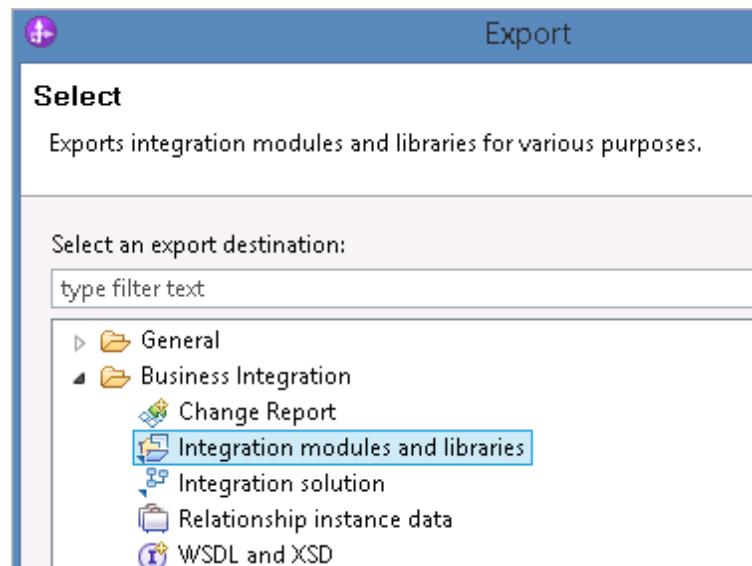


- ___ e. Verify that the **Global** option is selected and save your changes.

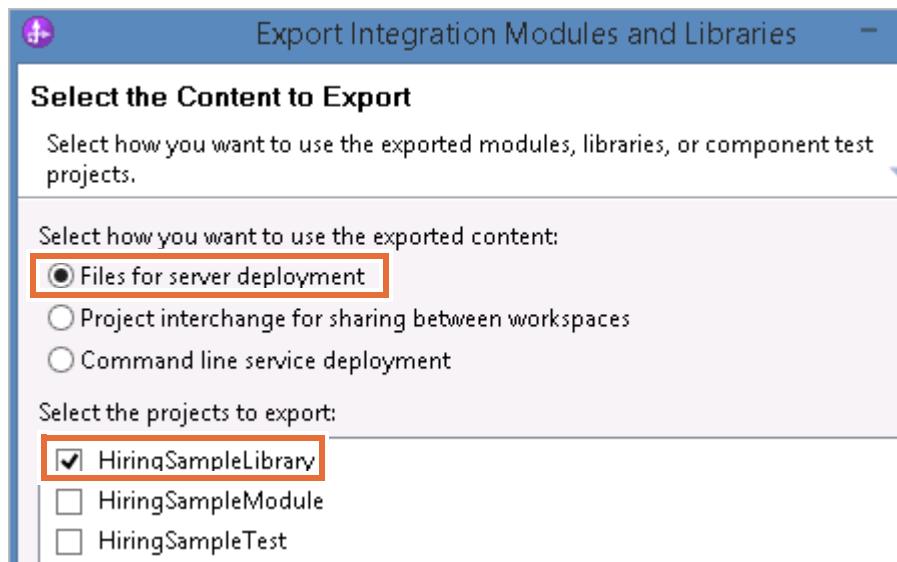


- ___ f. Close the **Dependencies** editor.
- ___ 2. Export **HiringSampleLibrary** as a .jar file so it can be deployed to the server.
 - ___ a. Click **File > Export** from the menu options.

- ___ b. In the **Export** dialog box, expand **Business Integration** and select **Integration modules and libraries**.

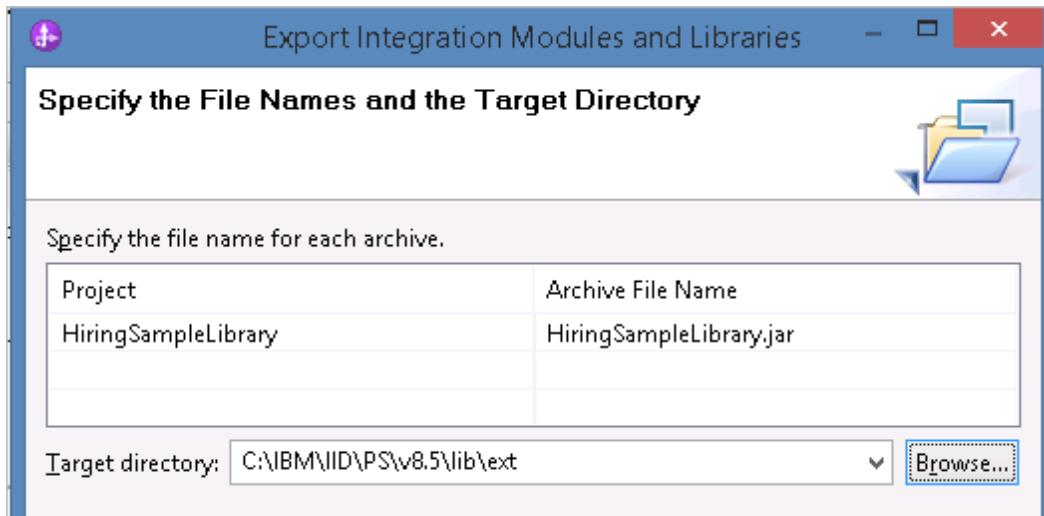


- ___ c. Click **Next**.
- ___ d. At the **Select the Content to Export** dialog box, verify that **Files for server deployment** is selected, and that **HiringSampleLibrary** is selected.



- ___ e. Click **Next**.

- __ f. At the **Specify the File Names and the Target Directory** dialog box, take the following actions:
- For **Target Directory**, click **Browse**, browse to `C:\IBM\IID\PS\v8.5\lib\ext`, and click **OK**.
 - Verify that the **Archive File Name** is set to: `HiringSampleLibrary.jar`

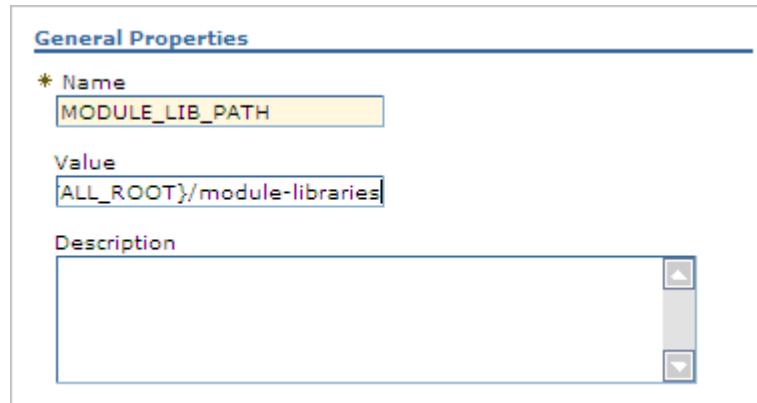


- __ g. Click **Finish**.
- __ 3. Create a folder that is named `C:\IBM\IID\PS\v8.5\profiles\qbpmaps\module-libraries` and copy the exported library from `C:\IBM\IID\PS\v8.5\lib\ext` into the new folder.
- __ a. In Windows Explorer, browse to `C:\IBM\IID\PS\v8.5\profiles\qbpmaps`.
 - __ b. Create a folder in the `\qbpmaps` directory and change the name to: `module-libraries`
 - __ c. Copy `HiringSampleLibrary.jar` from `C:\IBM\IID\PS\v8.5\lib\ext` to `C:\IBM\IID\PS\v8.5\profiles\qbpmaps\module-libraries`.
 - __ d. Close Windows Explorer.
- __ 4. Create a variable that is named `MODULE_LIB_PATH` at the node scope. The variable value is: `${USER_INSTALL_ROOT}/module-libraries`
- __ a. In IBM Integration Designer, switch to the **Servers** view.
 - __ b. Right-click **IBM Process Server v8.5.7 at localhost** and click **Administration > Run Administrative Console** from the menu.
 - __ c. If security alert messages are displayed, click **Yes** to proceed.
 - __ d. At the login page, enter `admin` in the **User ID** field and `web1sphere` in the **Password** field, and click **Log in**.

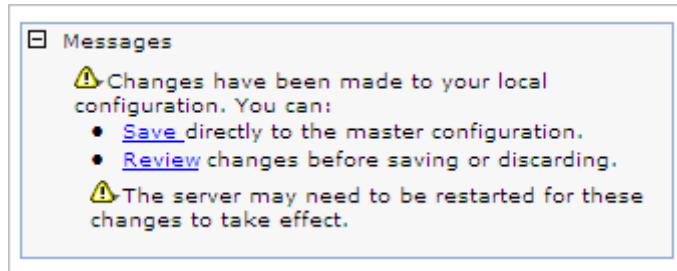
- ___ e. In the navigator pane, expand **Environment** and click **WebSphere variables**.



- ___ f. In the **Scope** section, select **Node=Node01**.
- ___ g. Click **New** to create a variable.
- ___ h. In the **General Properties** section for the new variable, enter the following information:
- In the **Name** field, type: `MODULE_LIB_PATH`
 - In the **Value** field, type: `${USER_INSTALL_ROOT} /module-libraries`

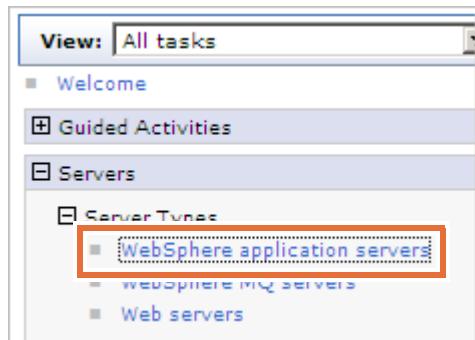


- ___ i. Click **Apply**.
- ___ j. Click the **Save** link in the **Messages** section at the top of the page to save the changes to the master configuration.



- ___ 5. Create a shared library at the node scope that is named `ModuleSharedLibraries`. The class path value for the library is `${MODULE_LIB_PATH}`.
- ___ a. In the navigator pane, expand **Environment** and click **Shared libraries**.
- ___ b. On the **Shared Libraries** page, in the **Scope** section, select **Node=Node1**.
- ___ c. Click **New** to create a shared library.

- ___ d. In the **General Properties** section for the new shared library, enter the following information:
- In the **Name** field, type: ModuleSharedLibraries
 - In the **Classpath** box, type: \${MODULE_LIB_PATH}
- ___ e. Click **Apply**.
- ___ f. Click the **Save** link in the **Messages** section at the top of the page to save the changes to the master configuration.
- ___ 6. Create a class loader for the test server and add **ModuleSharedLibraries** to the class loader as a shared library reference.
- ___ a. In the navigator pane, expand **Servers > Server Types** and click **WebSphere application servers**.



- ___ b. On the **Application servers** section, click the **server1** link.
- ___ c. In the **Server Infrastructure** section, expand **Java and Process Management** and click **Class loader**.



- ___ d. Click **New** to create a class loader.
- ___ e. Accept the default option (**Classes loaded with parent class loader first**) and click **Apply**.
- ___ f. Click the **Save** link in the **Messages** section at the top of the page to save the changes to the master configuration.

- ___ g. When you return to the **Class loader** page, click the link for the new class loader in the **Class loader ID** column. Your ID might differ from the following graphic.

		New	Delete		
					
Select	Class loader ID	Class loader order			
You have no class loaders defined.	Classloader_1266966219234	Sources:			
		Classes loaded with parent class loader first			
		Total 1			

- ___ h. In the **Additional Properties** section, click **Shared library references**.

<u>General Properties</u>		<u>Additional Properties</u>
Class loader ID	<input type="text" value="Classloader_1266966219234"/>	<input checked="" type="checkbox"/> Shared library references
Class loader order	<input type="text" value="Classes loaded with parent class loader first"/>	

- ___ i. Click **Add**.

- ___ j. For Library name, select **ModuleSharedLibraries**.

<u>General Properties</u>	
Library name	<input type="text" value="ModuleSharedLibraries"/>

- ___ k. Click **Apply**.

- ___ l. Click the **Save** link in the **Messages** section at the top of the page to save the changes to the master configuration.

- ___ m. Click **Logout** to log out of the administrative console.

- ___ n. Close the administrative console tab.

- ___ 7. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Restart** from the menu. The server must be restarted for your changes to take effect. The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.

- ___ 8. Deploy the hiring sample application modules to the test server.

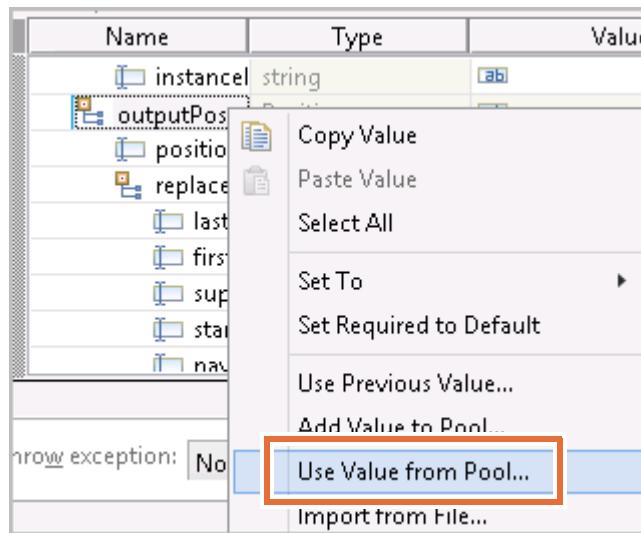
- ___ a. When the server restarts, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
- ___ b. If they are not already added, click **Add All** to add the projects to the **Configured projects** list.

- __ c. Click **Finish**.
- __ d. Wait for the publishing action to complete and for the modules to start. Publishing is complete when the status bar contains no messages such as: Publishing to IBM Process Server v8.5.7 at localhost.

The modules are started when you see these messages in the Server Logs view:

```
Application started: HiringSampleModuleApp
Application started: HiringSampleTestApp
```

- __ e. Verify that the modules started. If the status is not **Started**, then right-click the module and click **Restart** from the menu.
- __ 9. Test the application.
 - __ a. In the Business Integration view, open the **HiringSampleModule** assembly diagram, right-click the **OpenNewPosition** component, and click **Test Component in Isolation** from the menu.
 - __ b. On the test tab, click **Continue** to start testing the process application.
 - __ c. If you are prompted to do so, select **IBM Process Server v8.5.7 at localhost** from the list of deployment locations, and click **Finish**.
 - __ d. If you are prompted to do so, enter `admin` as the **User ID** and `web1sphere` as the **Password**.
 - __ e. Click **OK**.
 - __ f. The process stops at emulating the **SubmitRequisition** human task.
 - __ g. Right-click the **outputPosition** output parameter and click **Use Value from Pool**.



Important

Select the Position data object, not the Requisition data object.

- ___ h. Options are available for you to select from the pool of test data. Select **NewPosition001** and click **OK**.
 - ___ i. Click **Continue** when the sample data is populated. The business process completes successfully.

The success of this test demonstrates that the deployed modules borrow their resources from the shared library on the application server.
- ___ 10. Remove the applications from the server and stop the server.
- ___ a. In the **Servers view**, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
 - ___ b. Click **Remove All** and click **Finish** to remove the projects from the server. The status bar indicates when the publish (removal) action is complete. The projects are also removed from the Servers view.
 - ___ c. Close any windows that might be open. Do not save the test trace.
 - ___ d. Optionally, in the **Servers view**, right-click **IBM Process Server v8.5.7 at localhost** and click **Stop** from the menu.
- ___ 11. Close IBM Integration Designer.

End of exercise

Exercise review and wrap-up

In this exercise, you assembled an SCA application and examined project components. You used the IBM Integration Designer Test environment to examine projects and run a test component. You also deployed a library globally.

Exercise 4. Working with web services

What this exercise is about

This exercise demonstrates two methods for working with web services in IBM Integration Designer. First, you import an external Web Service Description Language (WSDL) file given to you by a third party. You use this interface file to integrate a web service into your application by using IBM Integration Designer. After importing the WSDL file, you test the web service. Second, you use a web service export to expose an existing IBM Process Server application. You then use the SCA programming framework to call the export.

What you should be able to do

After completing this exercise, you should be able to:

- Import an external Web Services Description Language (WSDL) file into IBM Integration Designer
- Create an SCA component from a web service interface file
- Use the integrated test client to test a web service
- Use a web service export to expose an existing IBM Process Server application

Introduction

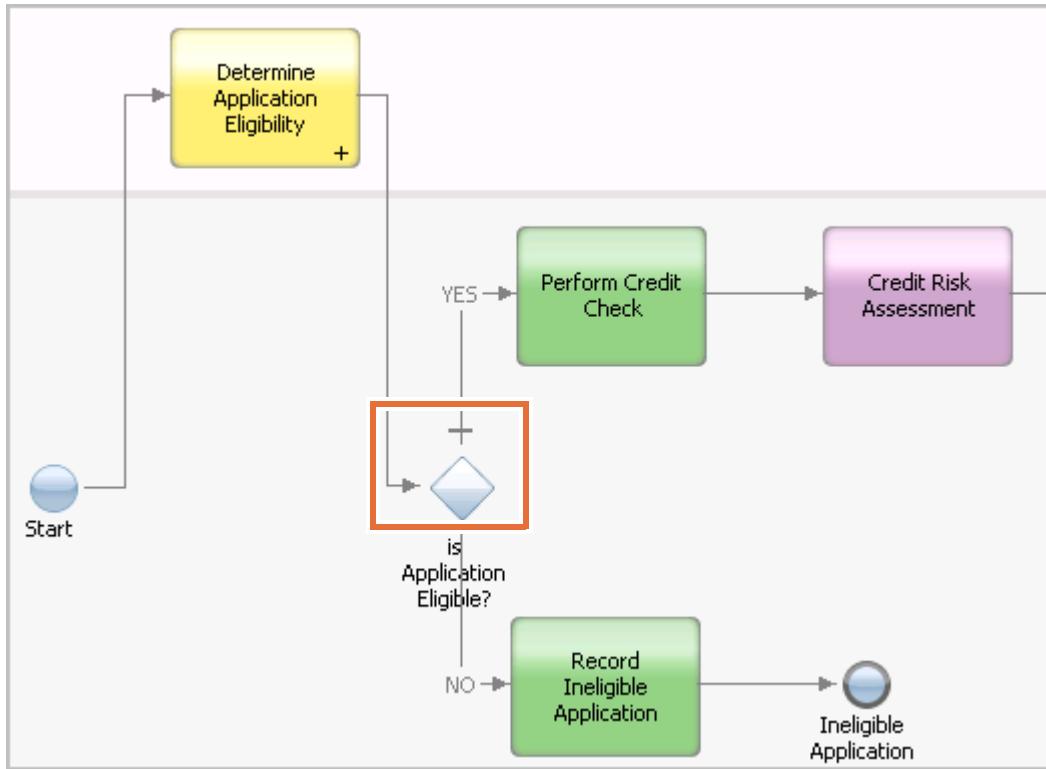
In your integration applications, you frequently invoke services that your Business Partners provide. The main advantage of web service technology is that it makes business-to-business communication simple and easy. IBM Integration Designer uses the Service Component Architecture (SCA) to facilitate the use of web service technology to connect to existing web services.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Exercise instructions

This exercise is based on the Account Verification process application, which you explored in IBM Process Designer in a previous lab.



As indicated in the account verification model and in the process narrative, when the customer submits an application, it must be tested for eligibility. If the application is eligible, the system calls an external service to do a credit check. In its implementation, this external service can realistically be a web service that another business provides to you.

In this exercise, you use IBM Integration Designer to implement the Credit Check Service from this account verification model as a web service.

Part 1: Import an external Web Service Description Language file into IBM Integration Designer

In this portion of the exercise, you import the WSDL interface of an external web service. In practice, a Business Partner is likely to offer this web service. After importing the WSDL interface, you use the IBM Integration Designer integrated test environment to examine the generated artifacts and test connectivity with the service.

Starting IBM Integration Designer and creating a workspace

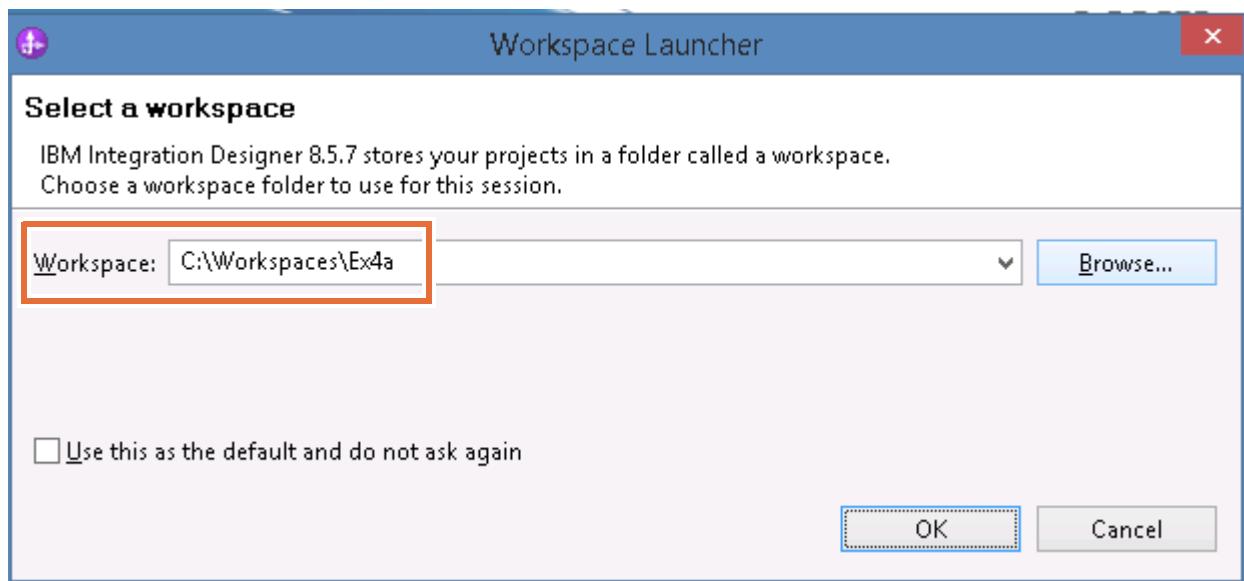
Before you begin the exercise, start IBM Integration Designer and create a workspace.

- 1. Create a workspace. This new **Ex4a** workspace does not exist in your lab environment. It is a new workspace that you create in this exercise.
 - a. Double-click the **IBM Integration Designer** shortcut on your desktop.
 - b. In the **Workspace Launcher** dialog box, change the default workspace location by entering **C:\Workspaces\Ex4a** in the **workspace** field.



Important

Do **not** select the **Use this as the default and do not ask again** check box.

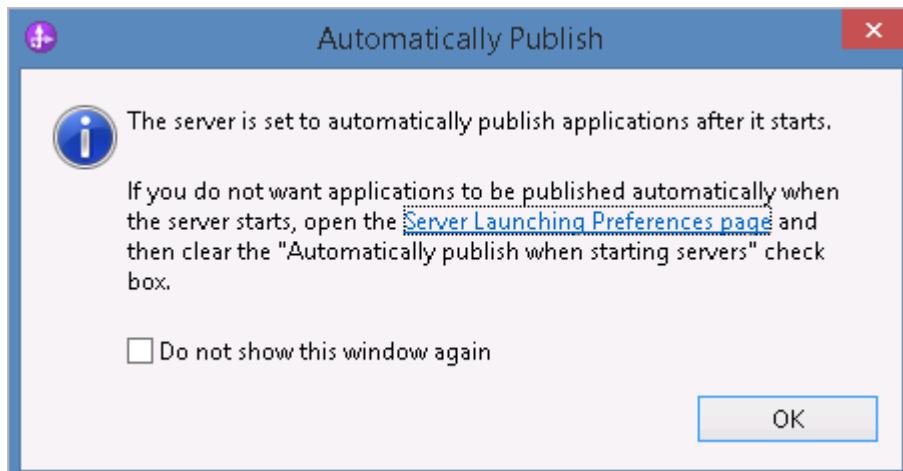


Note

The workspace path that is listed is not mandatory. You can create the workspace for this exercise in any location.

- c. Click **OK**.

- ___ d. If you get a message that the server is already set to publish, then click **OK**. You get this message when the server is already running from the previous exercise.



- ___ e. When the **Process Center Login** window is displayed, click **Cancel** to close that window. The Business Integration perspective opens.
___ f. Close the **Getting Started** tab.

Starting the prebuilt web service

Service-oriented architecture promotes flexibility and reusability of application functions. Based on the principles of service-oriented architecture, your Service Component Architecture application can interact with existing web services that other organizations, other companies, or your Business Partners provide.



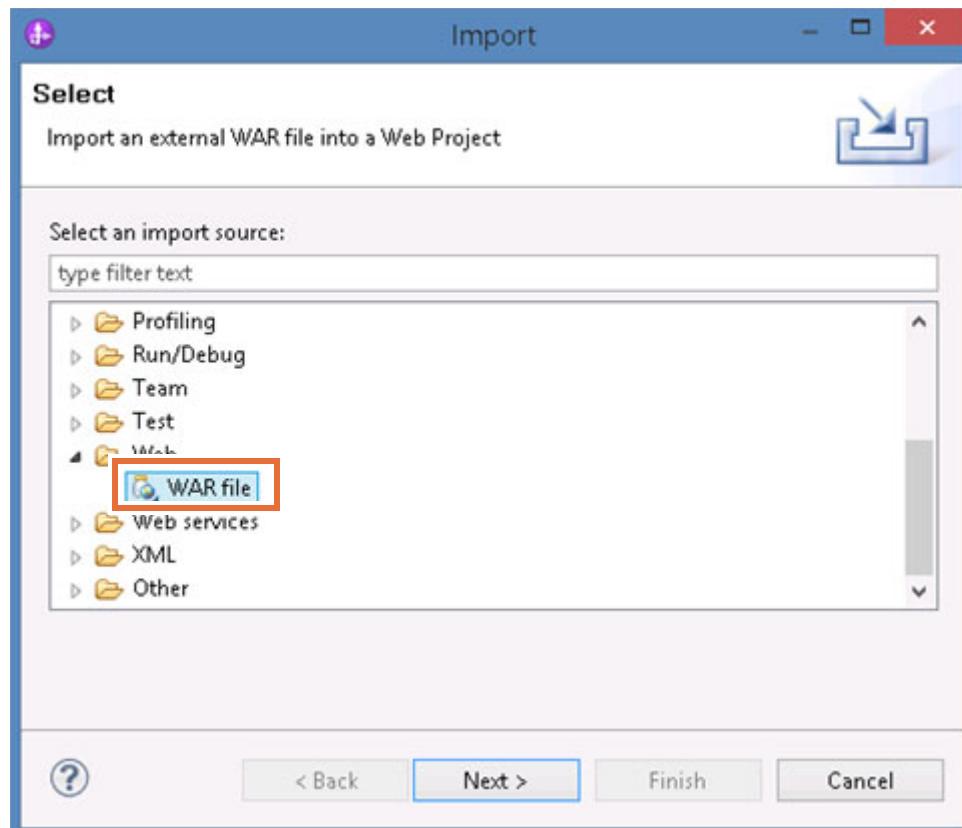
Important

For simplicity, you are going to run a prebuilt web service, **CreditReportService**, on your local IBM Process Server. In a production environment, this web service would be running in your Business Partner's environment.

To start the web service:

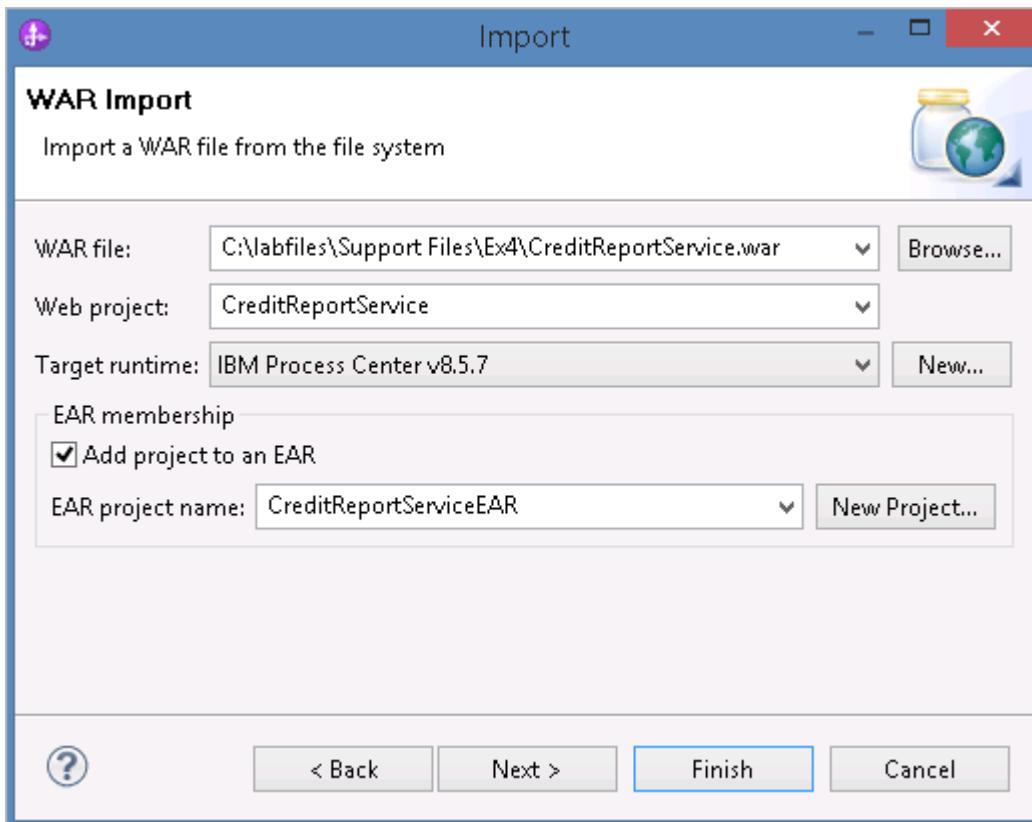
- ___ 1. Import the `CreditReportService.war` file and start the web service.
 - ___ a. Click **File > Import** from the menu options.

- ___ b. In the type selection window, expand **Web** and select **WAR file**.

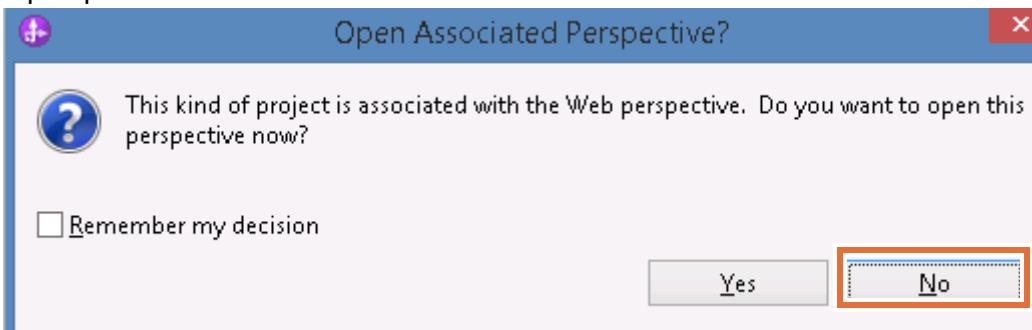


- ___ c. Click **Next**.
___ d. In the **Import** dialog box, click **Browse** beside the **WAR file** field and select C:\labfiles\Support Files\Ex4\CreditReportService.war.

- ___ e. Click **Open**. When you are returned to the WAR Import dialog box, in the **Target runtime** field, select **IBM Process Server v8.5.7**.



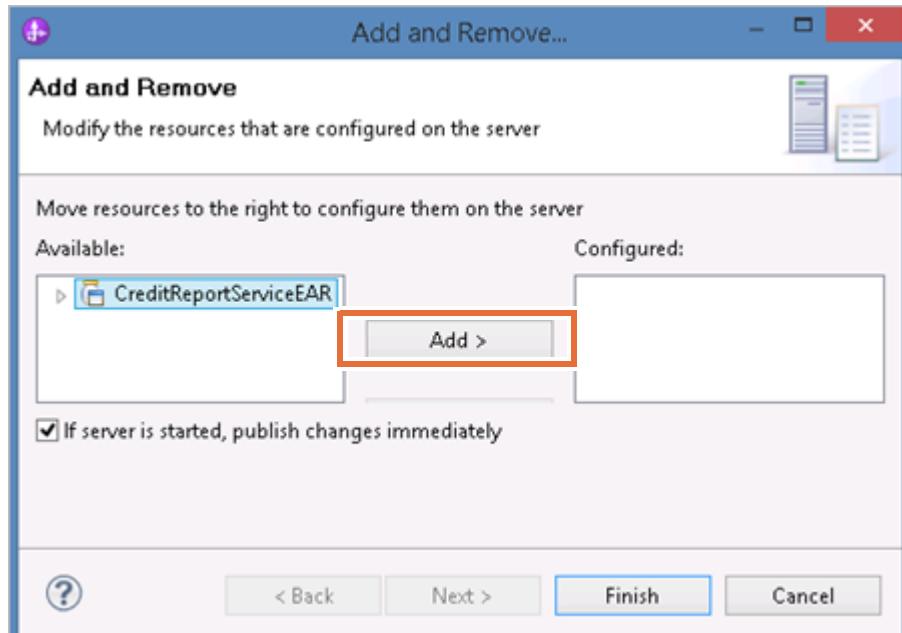
- ___ f. Accept the remaining default options and click **Finish**.
- ___ g. In the Open Associated Perspective dialog box, click **No** to avoid switching to the web perspective.



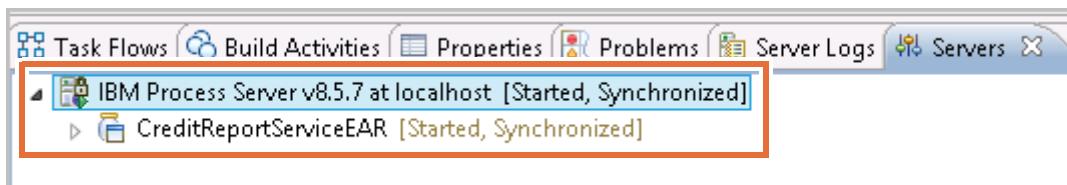
Information

Web services are not SCA artifacts, so it is necessary to switch to the appropriate perspective to work with these non-SCA components. IBM Integration Designer is built on top of Rational Application Developer technology.

2. Start IBM Process Server.
- __ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu. You can also click the **Start the server** icon to start the server. If the server is already started, then go to step c.
 - __ b. Wait for the server to complete the startup procedure. The **Servers** view shows the server state as **[Started, Synchronized]**.
 - __ c. Right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
 - __ d. Select **CreditReportServiceEAR** in the **Available** list, and click **Add**.



- __ e. When CreditReportServiceEAR is added to the **Configured** list, click **Finish** to deploy the CreditReportService web service.
- __ f. Wait for the **CreditReportService** application to start. When the application is started, the following message is displayed in the **Server Logs** view: Application started: CreditReportServiceEAR. The **Servers** view also shows the status of **CreditReportServiceEAR**.



- __ g. If the **CreditReportServiceEAR** application has a status of **Stopped**, then right-click the module and click **Restart** in the menu. If prompted to republish the module, republish it.
- __ h. Wait for the server status to change to **Started, Synchronized**. If the server has a status of **Started, Publishing**, then clicking the server refreshes the status to **Started, Synchronized**.

Part 2: Create a Service Component Architecture component from a web service interface file

In this portion of the exercise, you import a WSDL file that describes the CreditReportService web service. Then, you test your connectivity with the service that is using the IBM Integration Designer test environment. When your connectivity is confirmed, you implement the SCA components that are used to connect to the service.

To import the WSDL interface:

- ___ 1. Review the WSDL file that describes the CreditReportService web service.
 - ___ a. In Windows Explorer, browse to C:\labfiles\Support Files\Ex4\.
 - ___ b. Right-click **CreditReportService.wsdl** and click **Open with**.
 - ___ c. Click the **More options** link in the window that opens and then double-click **WordPad**.
 - ___ d. Examine the WSDL file.

CreditReportService.wsdl defines an operation **calculateCreditScore** that accepts an input message of type **calculateCreditScoreRequest**.

```
<wsdl:operation name="calculateCreditScore">
    <wsdlsoap:operation soapAction="calculateCreditScore"/>

    <wsdl:input name="calculateCreditScoreRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
```

The **calculateCreditScoreRequest** message is associated with data type **calculateCreditScore**.

```
<wsdl:message name="calculateCreditScoreRequest">
    <wsdl:part element="intf:calculateCreditScore"
        name="parameters"/>
</wsdl:message>
```

The **calculateCreditScore** data type is a complex data type that includes **CreditCheckDetail**.

```
<element name="calculateCreditScore">
    <complexType>
        <sequence>
            <element name="request" nillable="true"
                type="impl:CreditCheckDetail"/>
        </sequence>
    </complexType>
</element>
```

The definition of **CreditCheckDetail** contains the elements `acctNumber`, `companyName`, `creditScore`, and `dateRequested`.

```
<complexType name="CreditCheckDetail">
    <sequence>
        <element name="acctNumber" nillable="true"
            type="xsd:string"/>
```

```
<element name="companyName" nillable="true"  
type="xsd:string"/>  
  
<element name="creditScore" type="xsd:int"/>  
  
<element name="dateRequested" nillable="true"  
type="xsd:string"/>  
</sequence>  
</complexType>
```

**Note**

The web service address that you use to test the service is:

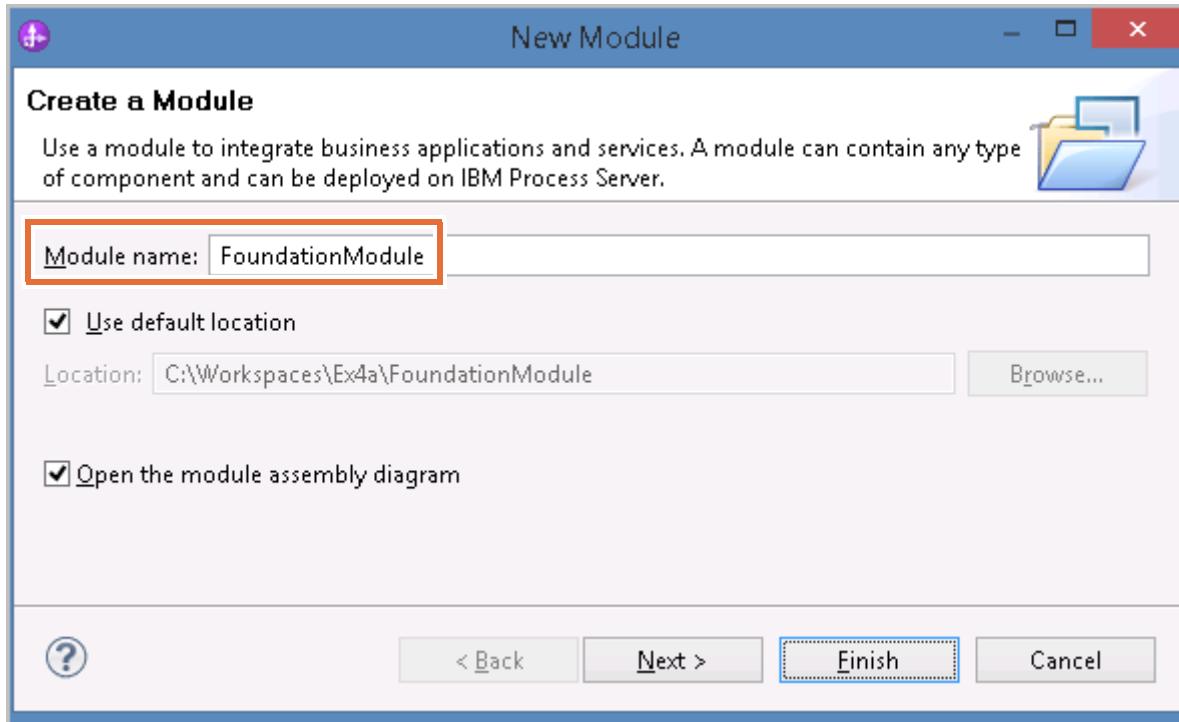
```
<wsdlsoap:address location="https://localhost:9443/CreditReportService/  
services/CreditReportService"/>
```

In a production environment, the address would contain

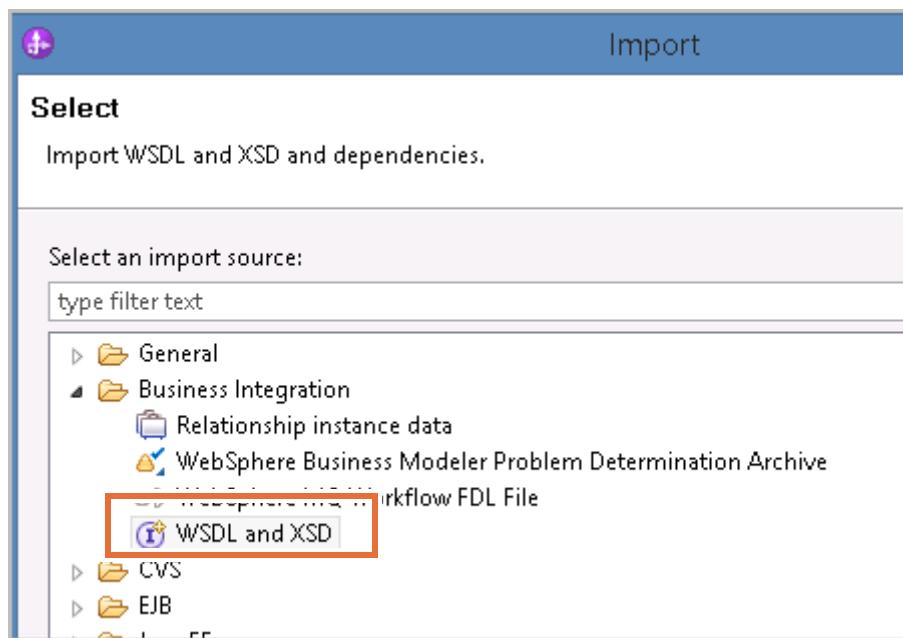
<https://www.<webaddress>.com/CreditReportService/> instead of localhost:port.

- e. Close CreditReportService.wsdl.
- f. Close Windows Explorer.
- 2. Create an IBM Integration Designer project that is called **FoundationModule**. This module is the starting point for the new end-to-end application to develop.
 - a. Click **File > New > Module** from the menu options.

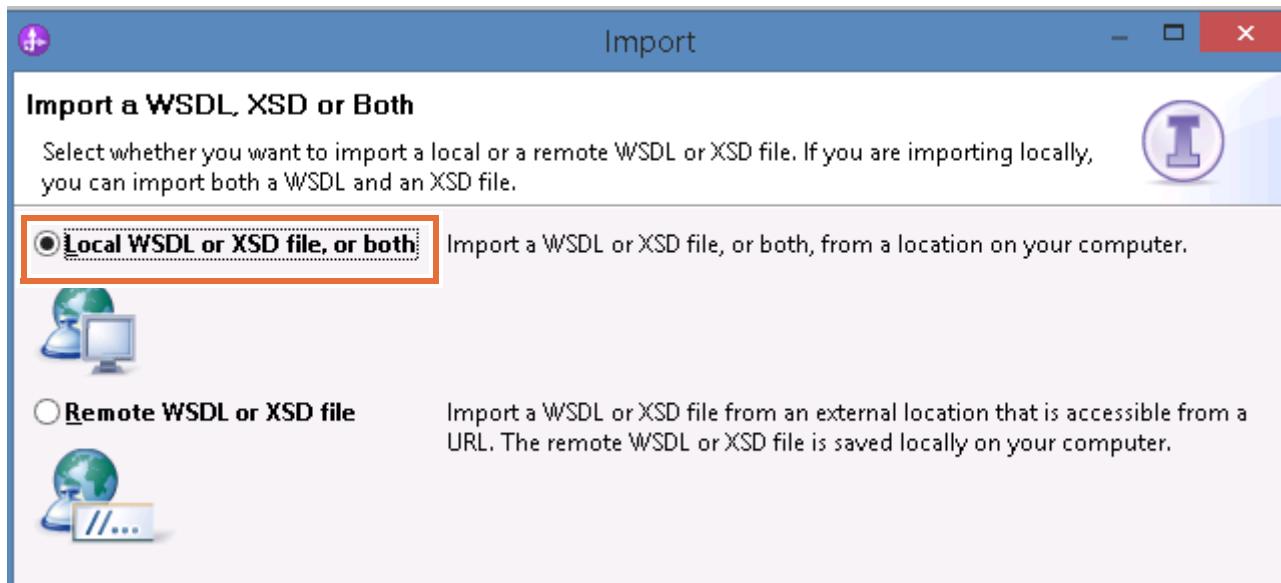
- __ b. At the “Create a Module” dialog box, type **FoundationModule** in the **Module name** field.



- __ c. Accept the remaining default options and click **Finish**.
- __ d. Wait until no messages appear in the IBM Integration Designer status bar, such as **Building workspace**.
- __ 3. Import `CreditReportService.wsdl` into **FoundationModule**.
- __ a. In the Business Integration view, right-click **FoundationModule** and click **Import**.
- __ b. Expand **Business Integration** and select **WSDL and XSD**.

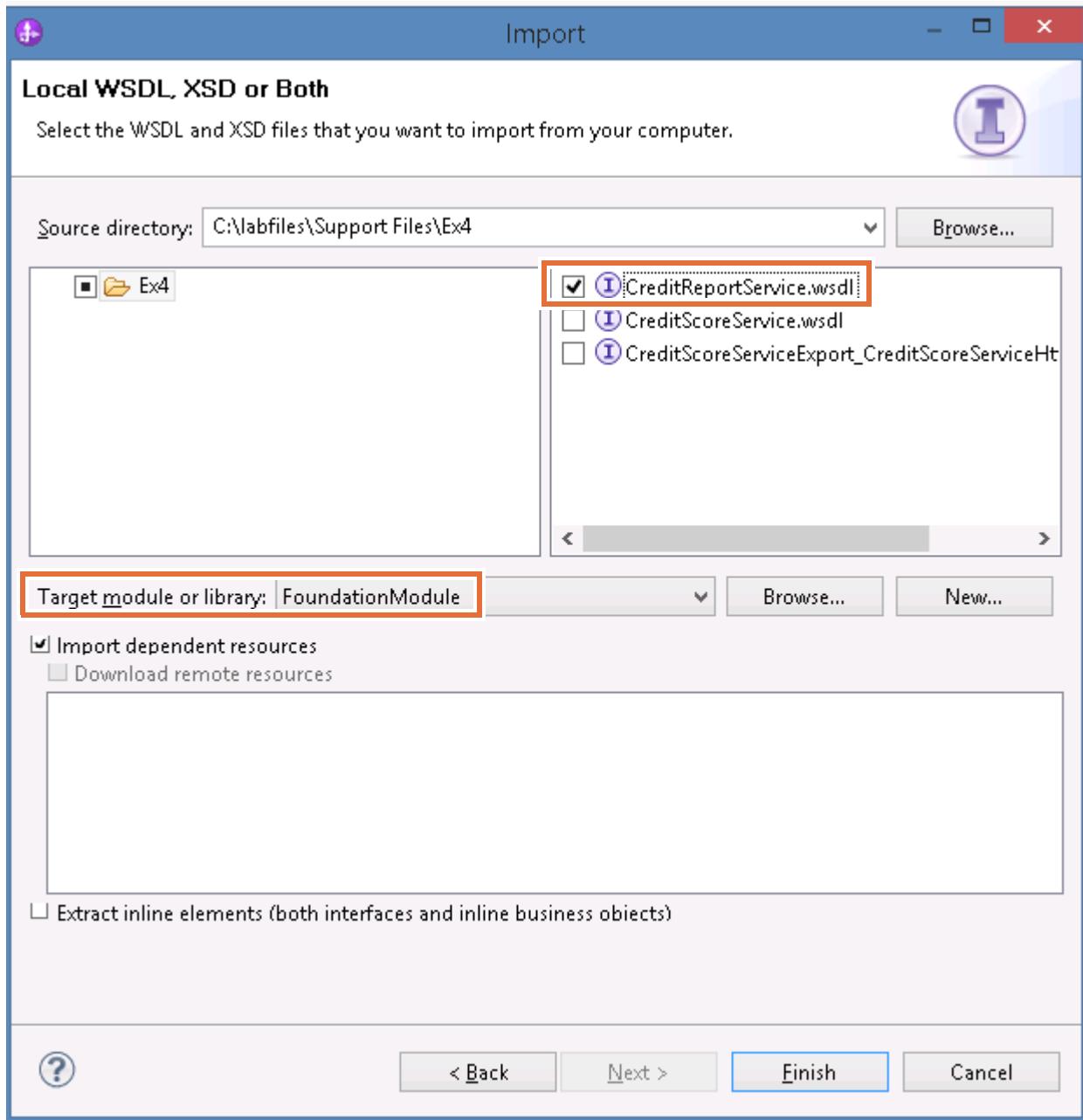


- ___ c. Click **Next**.
- ___ d. At the **Import a WSDL, XSD or Both** dialog box, select **Local WSDL or XSD file, or both**.



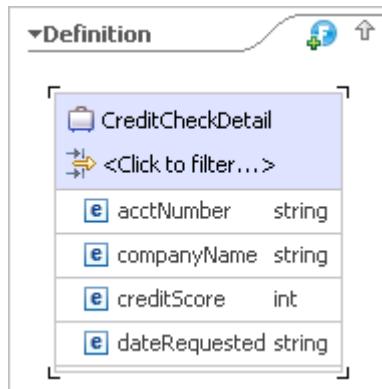
- ___ e. Click **Next**.
- ___ f. Click **Browse** next to **Source directory** and browse to: C:\labfiles\Support_Files
- ___ g. Select **Ex4** and click **OK**.
- ___ h. Select only the check box for **CreditReportService.wsdl**.

- __ i. Verify that **Target module or library** is set to: FoundationModule

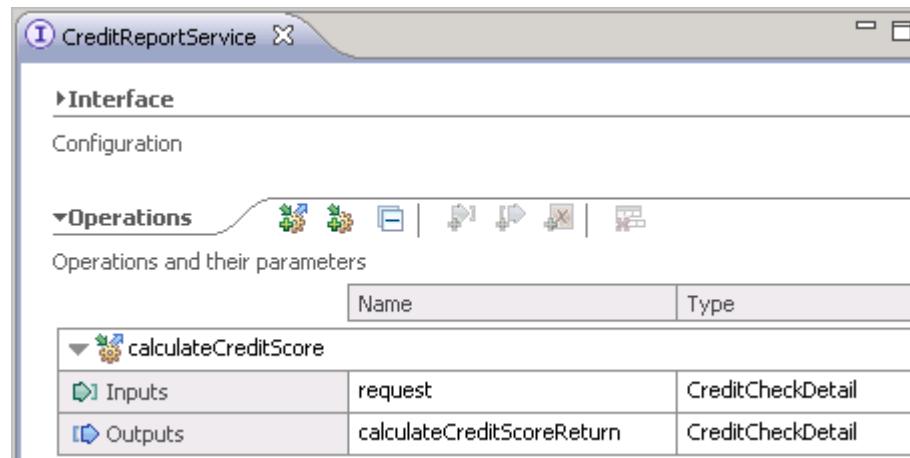


- __ 4. Accept the remaining default options and click **Finish**.

- 5. Examine the SCA artifacts in **FoundationModule** that were created as a result of the WSDL import. These artifacts were generated based on the information you examined earlier in the WSDL file.
- a. Expand **FoundationModule > Data** and double-click **CreditCheckDetail** to open the business object definition.



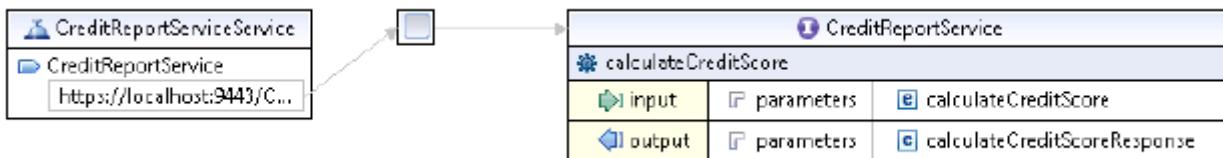
- b. Expand **Interfaces** and double-click **CreditReportService** to view the interface definition.



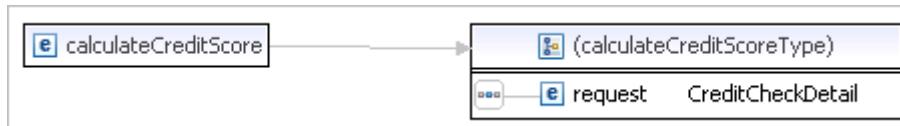
- c. Close the **CreditCheckDetail** tab and the **CreditReportService** tab.
- 6. Examine the WSDL file in the WSDL editor. The WSDL editor provides a graphical representation of the WSDL file that you examined previously in a text editor.
- a. Expand **FoundationModule > Interfaces**.
- b. Right-click **CreditReportService** and click **Open With > WSDL Editor** from the menu.

The WSDL editor provides you with a graphical view of the service address, the interface operation, and the input and output parameters. As you saw previously, the **calculateCreditScoreRequest** message is associated with data type **calculateCreditScore**.

- ___ c. Click the arrow to the right of **calculateCreditScore** input (it changes color when you hover over it).



- ___ d. A new tab opens, containing the inline schema of the WSDL file. The **calculateCreditScore** data type is a complex data type that includes **CreditCheckDetail**.



- ___ e. As time permits, explore other aspects of the interface in the WSDL editor. When you are done, close any open tabs.

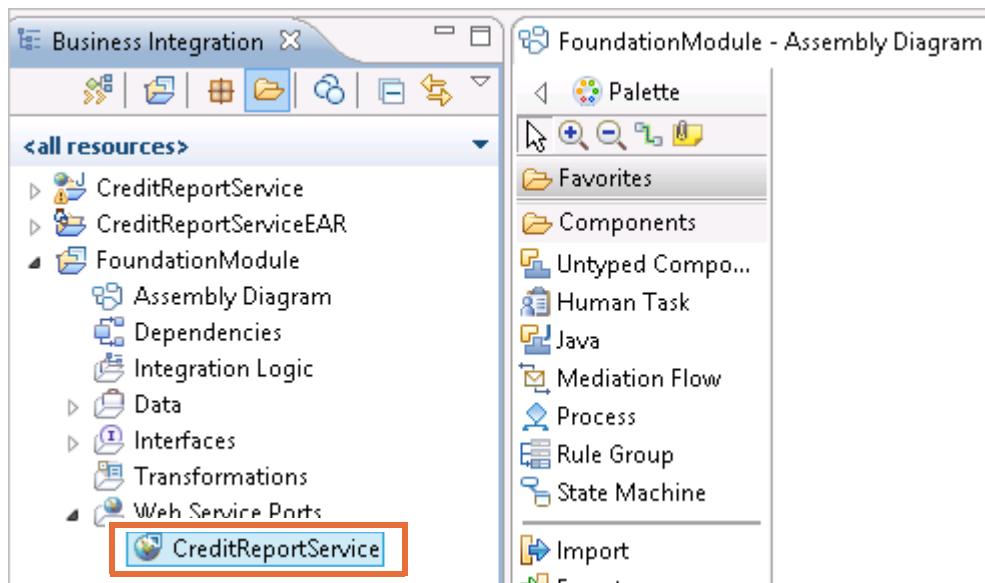
Creating an import component to invoke the CreditReportService

In the previous section, you imported the WSDL interface for your web service. In this portion of the exercise, you create the SCA import component that is used to invoke the CreditReportService web service in your application.

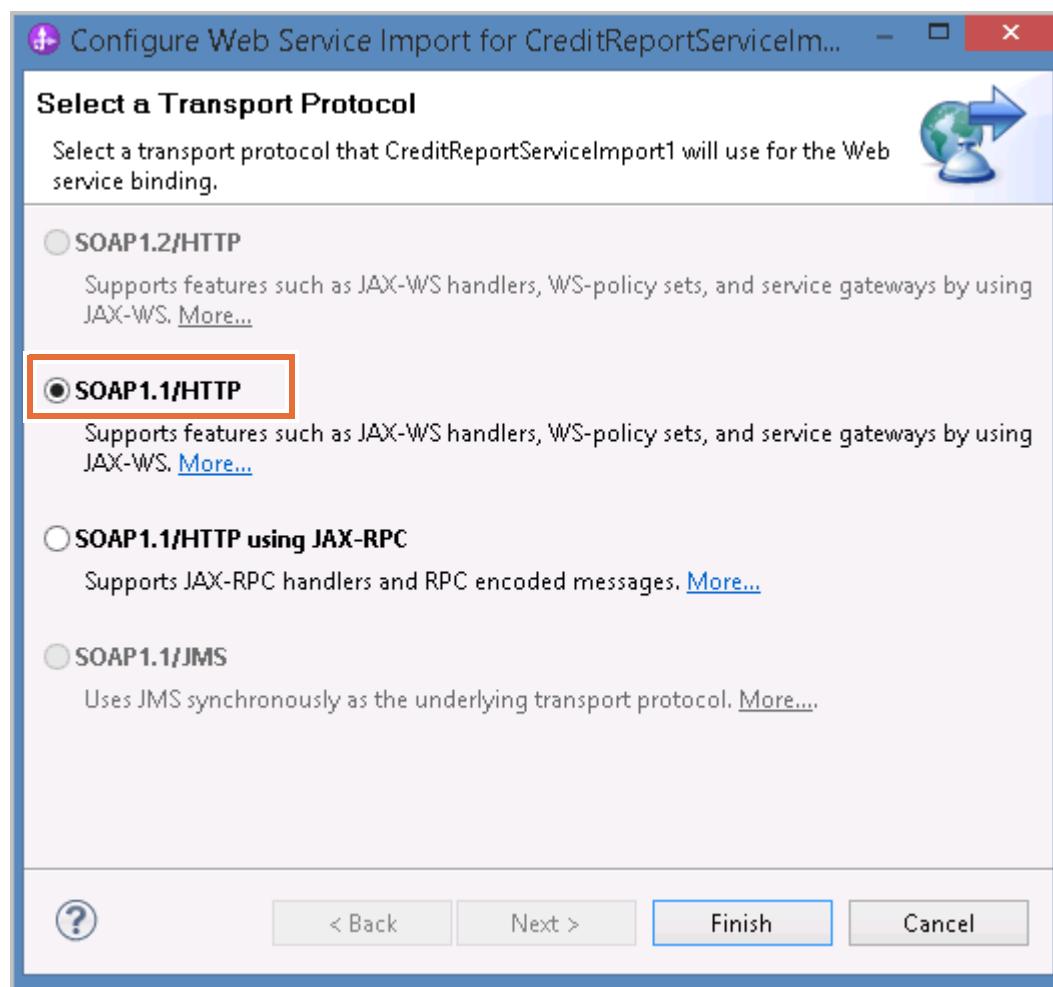
To create a CreditReportService import component:

- ___ 1. On the **FoundationModule** Assembly Diagram, create an import component to invoke the **CreditReportService** web service.
 - ___ a. If the assembly diagram is not open, expand **FoundationModule** and double-click **Assembly Diagram**.
 - ___ b. In the Business Integration view, expand **FoundationModule > Web Service Ports** and select **CreditReportService**.

- __ c. Drag **CreditReportService** onto the assembly diagram.

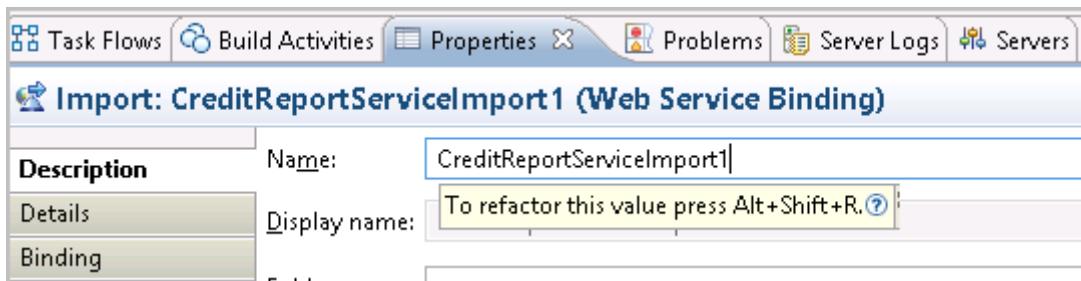


- __ d. Select the **SOAP1.1/HTTP** option.



- __ e. Click **Finish**.

- ___ f. Press Ctrl+S to save your changes.
- ___ 2. Change the default name of the import from `CreditReportServiceImport1` to: `CreditReportService`
 - ___ a. Select the **CreditReportServiceImport1** component.
 - ___ b. Switch to the **Properties** view.
 - ___ c. With the cursor in the **Name** field, press Alt+Shift+R to refactor the name.



- ___ d. In the Rename Artifact dialog box, type `CreditReportService` in the **New name** field.



- ___ e. Click **Refactor**. You can also analyze the effect of the changes before you click **Refactor**.



Hint

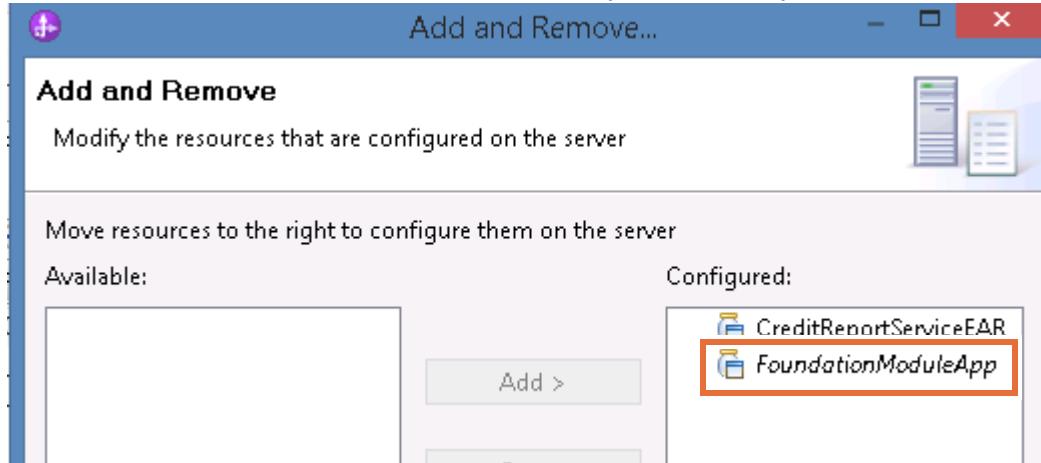
Although you can type the new name without using refactor, it is always a good idea to get in the habit of using **Refactor** to rename your components. These components often are referenced in other modules or libraries. Using refactor changes the name in all locations where the component is used.

Part 3: Use the integrated test client to test a web service

In this portion of the exercise, you test the web service import component. Using the IBM Integration Designer test client, you call the web service that is running in the local IBM Process Server test environment, which simulates a web service that is running in an external location. Many of the features in IBM Integration Designer support web services. A complete, full-featured web service development environment is available in Rational Application Developer.

To test the credit check service import:

- ___ 1. Deploy **FoundationModule** to the server and test the **CreditReportService** import component.
 - ___ a. If your server is not already started, in the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start**. The server was started when you deployed the credit check service EAR file earlier.
 - ___ b. When the server starts, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.
 - ___ c. Double-click **FoundationModuleApp** to add it to the **Configured projects** list, and click **Finish**. CreditReportServiceEAR was deployed previously.



- ___ d. Wait until the module is published and started. Publishing is complete when no messages appear in the IBM Integration Designer status bar such as **Publishing FoundationModuleApp**. The application is started when you see the message **Application started: FoundationModuleApp** in the **Server Logs** view.

Type	Time	Thread ID	Contents
<input type="checkbox"/> Log message	Apr 18, 2016 14:14:46.558 EDT	00000057	WSVR0200I: Starting application: CreditReportServiceEAR
<input type="checkbox"/> Log message	Apr 18, 2016 14:14:47.073 EDT	00000057	WSVR0221I: Application started: CreditReportServiceEAR
<input type="checkbox"/> Log message	Apr 18, 2016 14:14:51.511 EDT	00000057	WSVR0200I: Starting application: FoundationModuleApp
<input type="checkbox"/> Log message	Apr 18, 2016 14:14:52.401 EDT	00000057	WSVR0221I: Application started: FoundationModuleApp

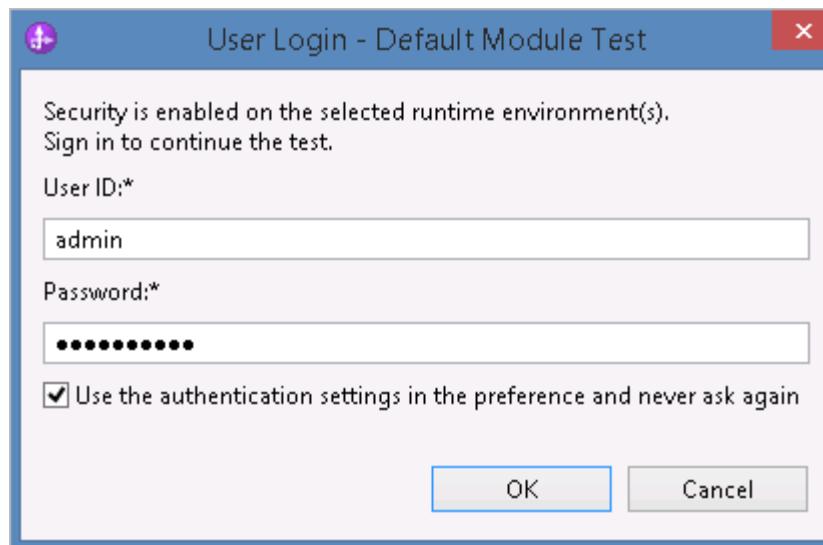
- ___ e. If the FoundationModuleApp application has a status of **Stopped**, then right-click the module and click **Restart** in the menu. If prompted to republish the module, republish it.
- ___ f. In the assembly diagram, right-click **CreditReportService** and click **Test Component** from the menu to open the integrated test client.

- __ g. In the **Initial request parameters** table, enter the following test data:

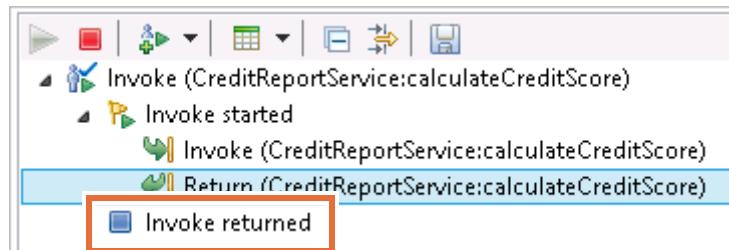
- acctNumber: 100
- companyName: IBM
- creditScore: 0
- dateRequested: 04/15/2016

Name	Type	
request	CreditCheckDetail	[ab]
acctNumber *	string	[ab] 100
companyName *	string	[ab] IBM
creditScore *	int	[ab] 0
dateRequested *	string	[ab] 04/15/2016

- __ h. Click **Continue** on the Events toolbar.
- __ i. When the Select a Deployment Location dialog box is displayed, select **IBM Process Server v8.5.7 at localhost**, select **Use this location as the default and do not ask again**, and click **Finish**.
- __ j. When the User Login dialog box is displayed, select the **Use the authentication settings in the preference and never ask again** check box, and click **OK**.



- k. Wait for the test to complete. When the test is finished, a stop node labeled “Invoke returned” is listed in the Events window.



- l. Select the **Return** event unless it is already selected and examine the **Return parameters** section. The result of the test is a randomly generated credit score in the range 0–11. Because the score is generated randomly, your result might differ from the following screen capture.

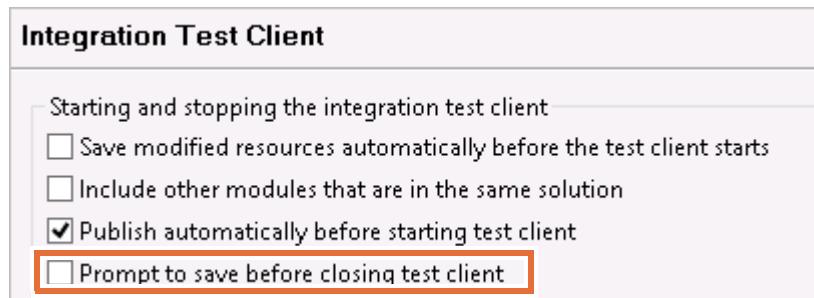
Name	Type	Value
calculateCreditScore	CreditCheckDetail	[ab]
acctNumber	string	[ab] 100
companyNan	string	[ab] IBM
creditScore *	int	[ab] 10
dateRequested	string	[ab] 04/15/2016

- m. Close the test client tab. When you are prompted to save the test trace, click **No**.



Note

You can disable the dialog box that prompts you to save the test trace. To disable the dialog box, click **Window > Preferences > Business Integration > Test > Integration Test Client**. Clear the **Prompt to save before closing test client** check box and click **OK**.



- 2. Remove the projects and (optionally) stop the server.
- a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.

- ___ b. Click **Remove All** and click **Finish**. Wait until no messages appear in the IBM Integration Designer status bar.

**Note**

If your computer has ample memory or if you are working on the exercises in the remote environment (for instructor-led online or self-paced courses), you might want the server to continue running to save time. If you choose to leave it running, you can ignore any steps in subsequent labs that require you to stop the server.

- ___ 3. Close IBM Integration Designer.

Part 4: Use a web service export to expose an existing IBM Process Server application

In this portion of the exercise, you use a different approach to implementing web service connectivity. In this approach, you use an export with a Web Service Binding type to expose an existing IBM Process Server application. In future exercises, this application replaces the service that you used in the previous section.

Unlike the previous service, this web service returns a specific credit score value for each of the test cases that are built into your application. For more information about how the application was created, see the appendixes.

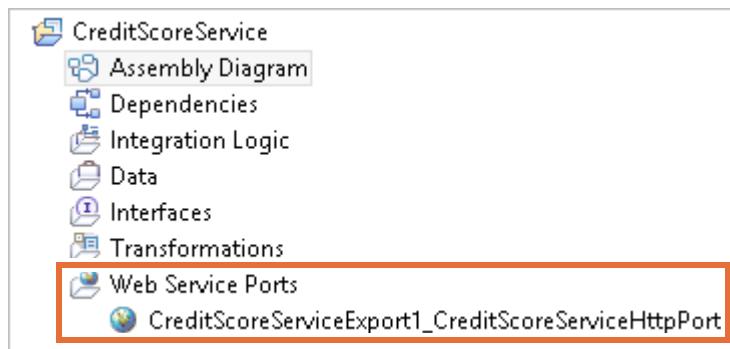
To expose an application by using a web service export:

- ___ 1. Open the Exercise 4b workspace.
 - ___ a. On your desktop, open the **Exercise Shortcuts** folder.
 - ___ b. Double-click the **Exercise 4b** shortcut.
 - ___ c. If you get a message that the server is already set to publish, then click **OK**. You get this message when the server is already running.
 - ___ d. Close the **Getting Started** tab.
- ___ 2. Use an export component with a Web Service Binding to expose the credit score rule group.
 - ___ a. In the Business Integration view, expand **CreditScoreService** and double-click **Assembly Diagram**.
 - ___ b. Right-click the **CreditScoreRG** component and click **Generate Export > Web Service Binding** from the menu.

- ___ c. In the “Select a Transport Protocol” dialog box, select **SOAP1.2/HTTP**. Because you are generating a new Web Service Binding, you can use the newer SOAP V1.2 protocol.

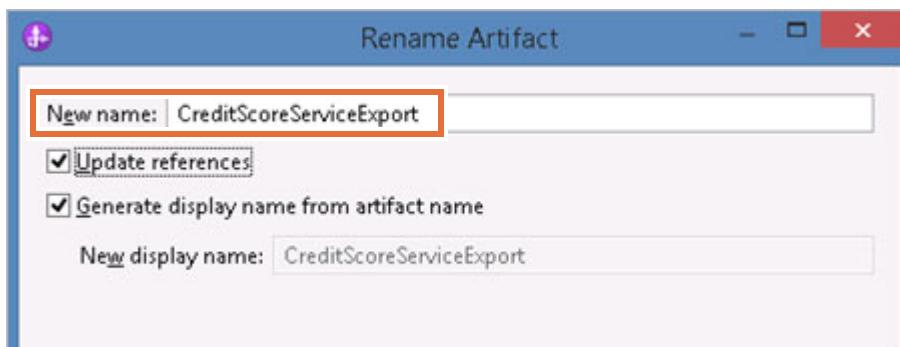


- ___ d. Click **Finish**. When the export is generated, a **Web Service Ports** section is added to the project.



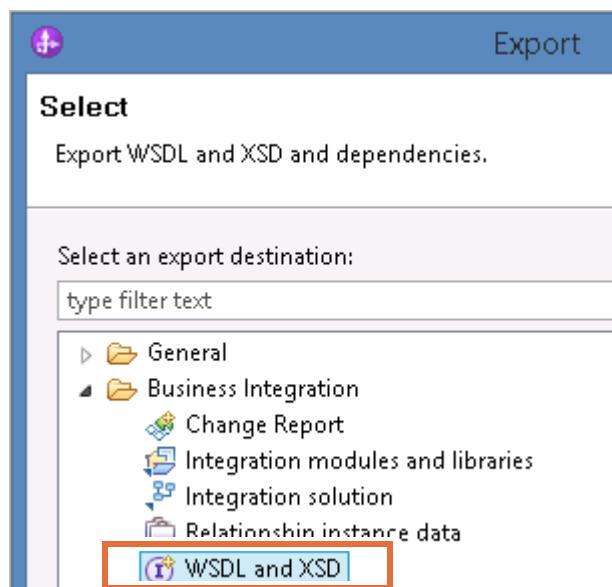
- ___ e. Save the changes to the assembly diagram.

- ___ 3. View the properties of the web service export.
- ___ a. On the assembly diagram, select **CreditScoreServiceExport1** and switch to the **Properties** view.
- ___ b. With the cursor in the **Name** field, press Alt+Shift+R.
- ___ c. In the Rename Artifact dialog box, change the value in the **New name** field to: **CreditScoreServiceExport**

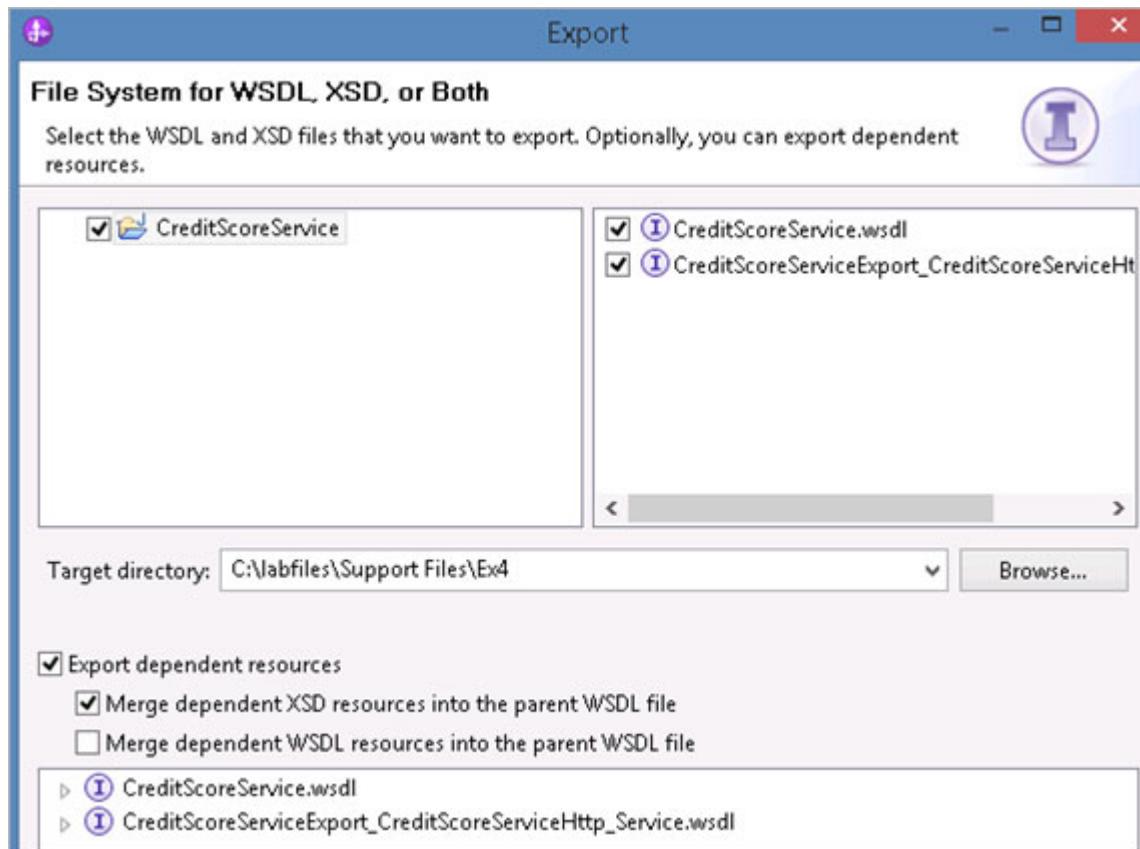


- ___ d. Accept the remaining default options and click **Refactor**.

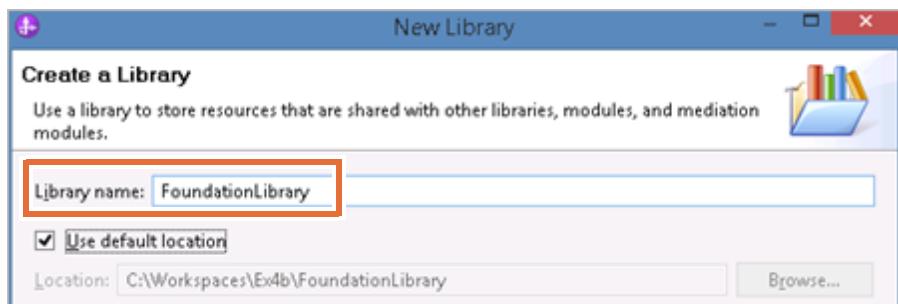
- 4. Export the credit score service WSDL file and web service port definition. Import them into a library that is named `FoundationLibrary`.
 - a. In the Business Integration view, right-click **CreditScoreService** and click **Export** from the menu.
 - b. In the Export dialog box, expand **Business Integration**, and select **WSDL and XSD**.
 - c. Click **Next**.



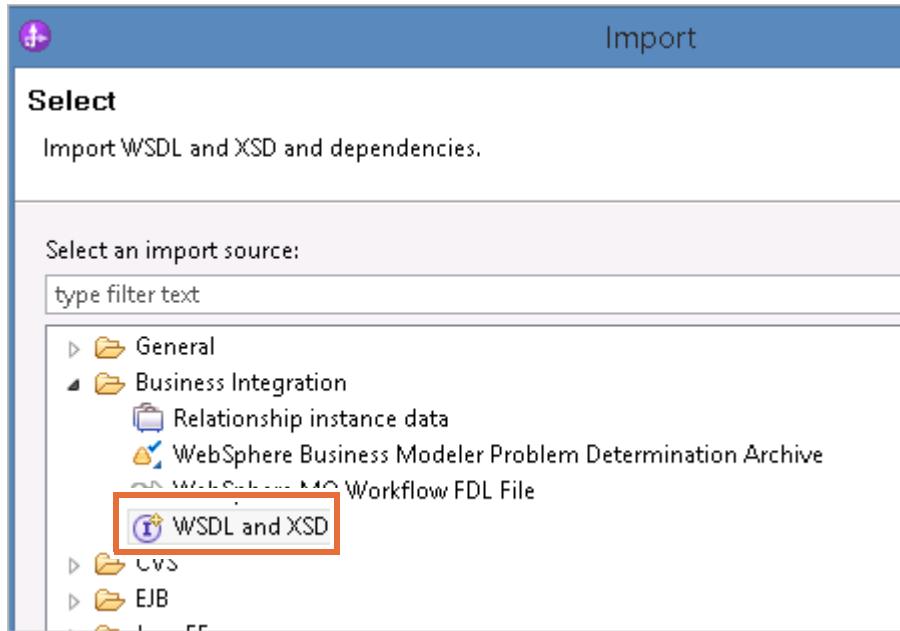
- ___ d. At the “File System for WSDL, XSD, or Both” window, do the following steps:
- Select **CreditScoreService.wsdl**, **CreditCheckRequest.xsd**, and **CreditScoreServiceExport_CreditScoreServiceHttp_Service.wsdl**.
 - For **Target directory**, click **Browse**. Browse to **C:\labfiles\Support Files\Ex4** and click **OK**.
 - Select **Merge dependent XSD resources into the parent WSDL file** (this action removes **CreditCheckRequest.xsd** from the window).



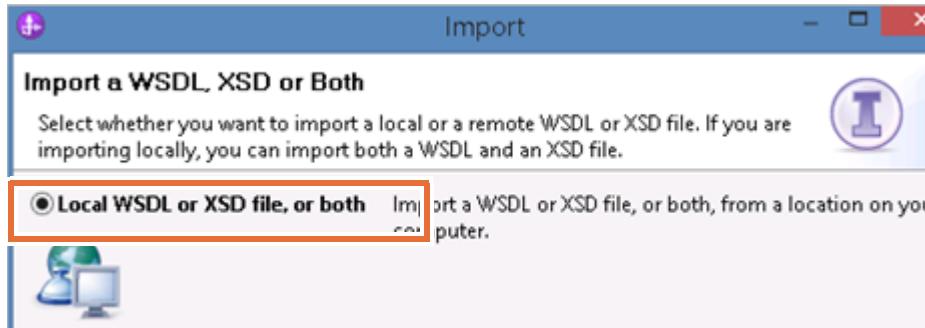
- ___ e. Accept the remaining default options and click **Finish**.
- ___ f. If you are prompted to overwrite any existing files, click **Yes**.
- ___ 5. Create a library that is named **FoundationLibrary** to store the WSDL interface and port definition.
- ___ a. Click **File > New > Library** from the menu options.
- ___ b. Type **FoundationLibrary** in the **Library name** field.



- ___ c. Accept the remaining default options and click **Finish**.
- ___ 6. Import the credit score service WSDL interface and port definition into FoundationLibrary.
- ___ a. In the Business Integration view, right-click **FoundationLibrary** and click **Import** from the menu.
- ___ b. Expand **Business Integration** and select **WSDL and XSD**.



- ___ c. Click **Next**.
- ___ d. In the “Import a WSDL, XSD or Both” dialog box, select **Local WSDL or XSD file, or both**.

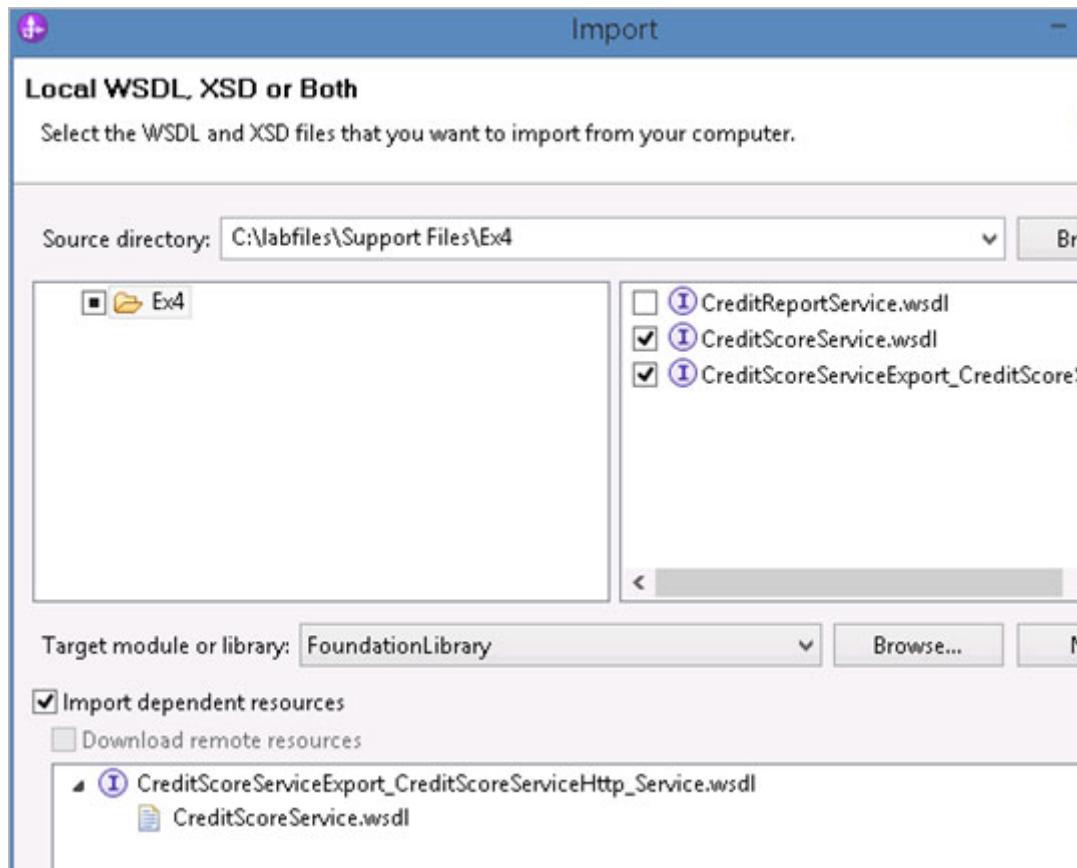


You can also use the import wizard to import files from an external location through a URL.

- ___ e. Click **Next**.

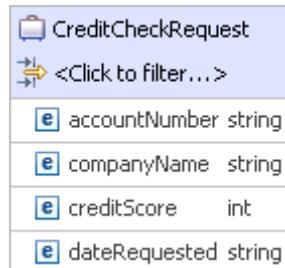
__ f. In the “Import a WSDL, XSD or Both” pane, take the following actions:

- For **Source directory**, click **Browse**, browse to C:\labfiles\Support Files\Ex4, and click **OK**.
- Select the **CreditScoreService.wsdl** and **CreditScoreServiceExport_CreditScoreServiceHttp_Service.wsdl** check boxes.
- Verify that **Target module or library** is set to **FoundationLibrary**.
- Verify that the **Import dependent resources** check box is selected.



__ g. Accept the remaining default options and click **Finish**.

- 7. Examine the SCA artifacts in FoundationLibrary that are created as a result of the WSDL import.
- a. Expand **FoundationLibrary > Data** and double-click **CreditCheckRequest** to open the business object definition.



- b. Expand **Interfaces** and double-click **CreditScoreService** to open the interface definition.

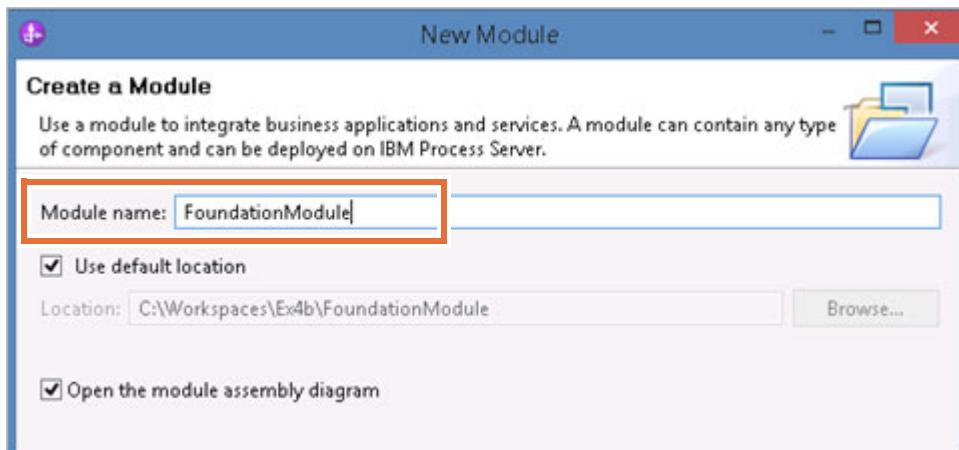
Interface

Configuration

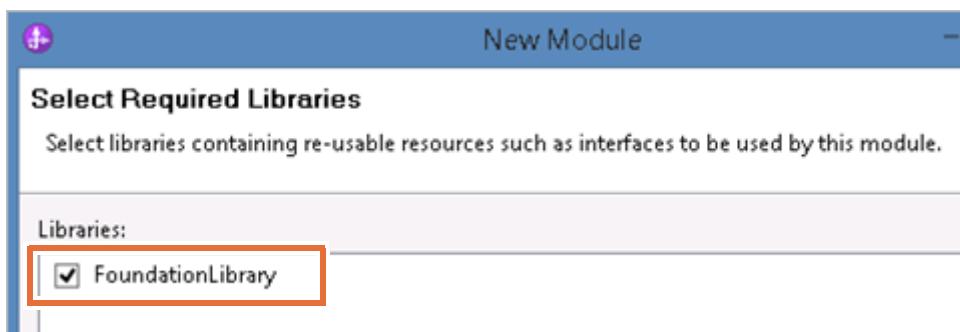
Operations

Name	Type
calculateCreditScore	
Inputs	request
Outputs	calculateCreditScoreReturn

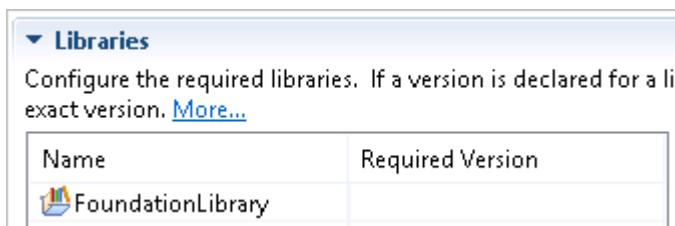
- c. Close both the **CreditCheckRequest** tab and the **CreditScoreService** tab.
- 8. Create an IBM Integration Designer project that is named **FoundationModule** that includes FoundationLibrary in its dependencies. This module is the starting point for the end-to-end application to develop throughout the remaining exercises.
- a. Click **File > New > Module** from the menu options.
- b. In the “Create a Module” dialog box, type **FoundationModule** in the **Module name** field.



- ___ c. Click **Next**.
- ___ d. In the Select Required Libraries dialog box, verify that **FoundationLibrary** is selected.

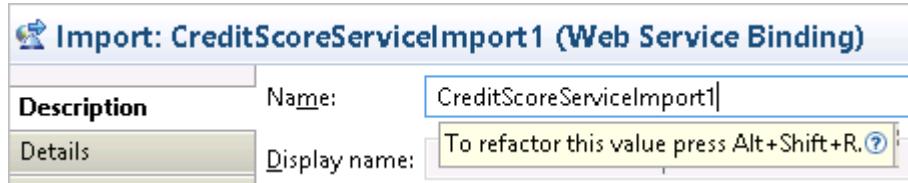


- ___ e. Click **Finish**.
 - ___ f. Wait until no messages appear in the IBM Integration Designer status bar, such as Building workspace.
- 9. View the FoundationModule dependencies.
- ___ a. Expand **FoundationModule** and double-click **Dependencies**.
 - ___ b. Expand the **Libraries** section.
 - ___ c. Verify that the **FoundationLibrary** is in the list.

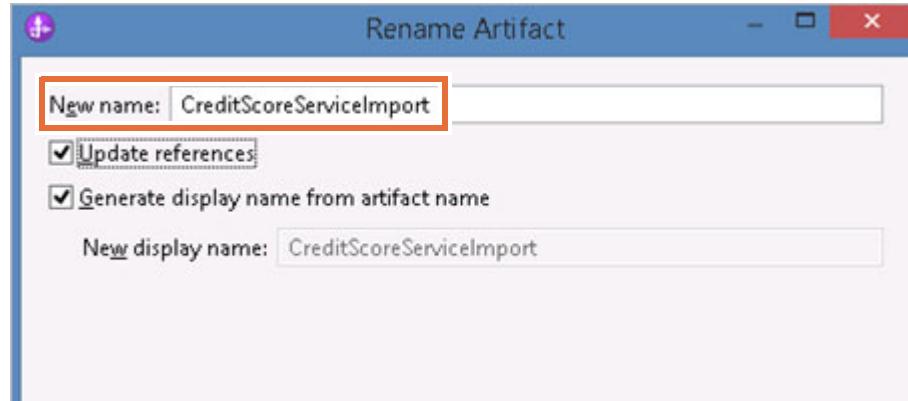


- ___ d. Close the **Dependencies** tab.
- 10. Add an import component that is named `CreditScoreServiceImport` to the FoundationModule assembly diagram that calls the credit score web service.
- ___ a. If the FoundationModule assembly diagram is not open, expand **FoundationModule** and double-click **Assembly Diagram**.
 - ___ b. In the Business Integration view, expand **FoundationLibrary > Web Service Ports**.
 - ___ c. Drag **CreditScoreServiceExport_CreditScoreServiceHttpPort** from **FoundationLibrary** onto the **FoundationModule** assembly diagram.
 - ___ d. Save the changes to the assembly diagram.
- 11. Refactor the name of the import from `CreditScoreServiceImport1` to `CreditScoreServiceImport`.
- ___ a. With `CreditScoreServiceImport1` selected, switch to the **Description** tab in the **Properties** view.

- ___ b. With the cursor in the **Name** field, press Alt+Shift+R.



- ___ c. In the Rename Artifact dialog box, change the value in the **New name** field to: CreditScoreServiceImport



- ___ d. Accept the remaining default options and click **Refactor**.

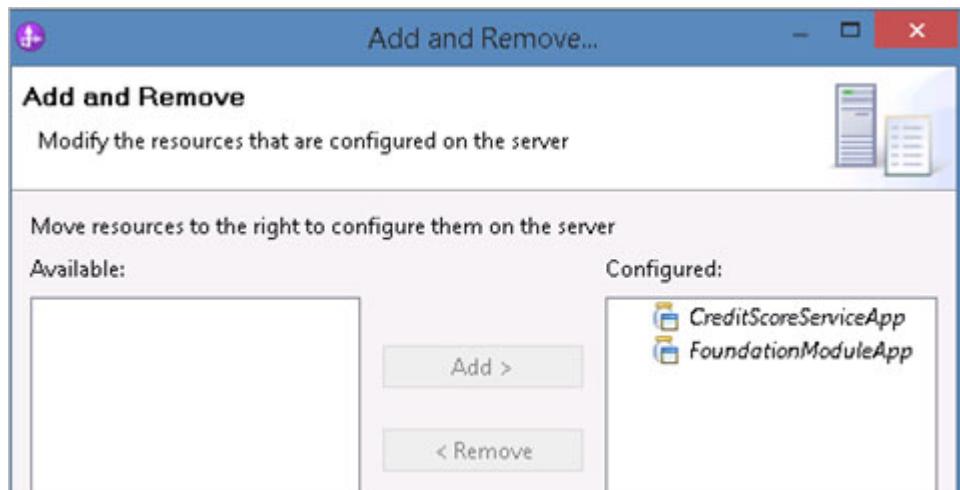
Testing the web service import

In this portion of the exercise, you test the web service import component. Using the IBM Integration Designer test client, you call the web service that is running in the local IBM Process Server test environment.

To test the credit score service import:

- ___ 1. Deploy FoundationModule and CreditScoreService to the server and test the CreditScoreServiceImport component.
 - ___ a. If your server is not already started, in the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start**. Wait for the startup process to complete before continuing.
The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
 - ___ b. When the server is started, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.

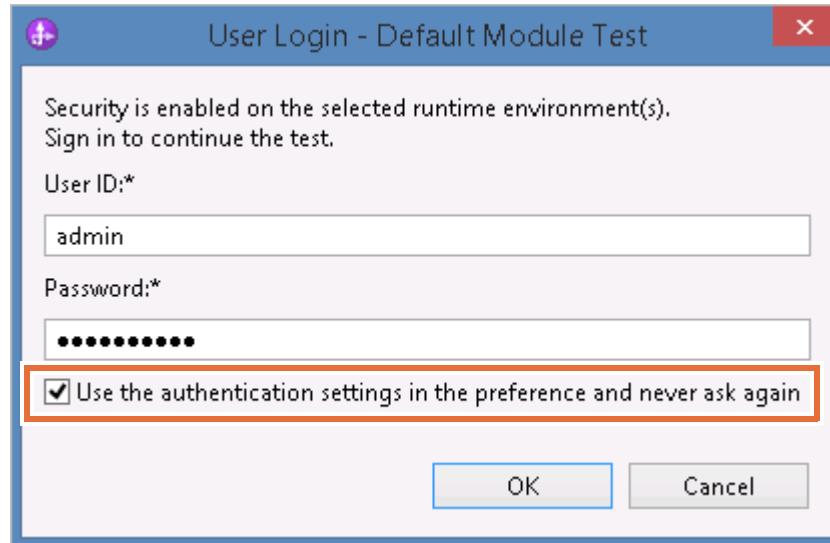
- c. Click **Add All** to add **FoundationModuleApp** and **CreditScoreServiceApp** to the Configured projects list.



- d. Click **Finish**.
- e. Wait until the module is published and started. Publishing is complete when no messages appear in the IBM Integration Designer status bar such as `Publishing FoundationModuleApp`. The applications are started when you see the messages `Application started: FoundationModuleApp` and `Application started: CreditScoreServiceApp` in the Server Logs view.
- f. If the **FoundationModuleApp** application or the **CreditScoreServiceApp** application has a **Stopped** status, then right-click the module and click **Restart**. If prompted to republish the module, republish it. Continue to the next step when the status is changed to **Started**.
- g. In the assembly diagram, right-click **CreditScoreServiceImport** and click **Test Component** from the menu.
- h. In the **Initial request parameters** table, enter the following test data:
- acctNumber: 100
 - companyName: IBM
 - creditScore: 0
 - dateRequested: 04/15/2016

Name	Type	
request	CreditCheckDetail	[ab]
acctNumber*	string	[ab] 100
companyName *	string	[ab] IBM
creditScore *	int	[ab] 0
dateRequested *	string	[ab] 04/15/2016

- __ i. Click **Continue** on the Events toolbar.
- __ j. When the “Select a Deployment Location” dialog box is displayed, select **IBM Process Server v8.5.7 at localhost**, select **Use this location as the default and do not ask again**, and click **Finish**.
- __ k. When the User Login dialog box is displayed, select the **Use the authentication settings in the preference and never ask again** check box, and click **OK**.



- __ l. When the test completes, a stop node labeled “Invoke returned” is displayed in the Events window.
- __ m. Select the **Return** event and examine the **Return parameters** section. The result of the test is a credit score of 11.

The Events window displays the following test steps:

- Invoke (CreditScoreServiceImport:calculateCreditScore)
 - Invoke started
 - Invoke (CreditScoreServiceImport:calculateCreditScore)
 - Return (CreditScoreServiceImport:calculateCreditScore)
 - Invoke returned

The Return event is selected, and its parameters are shown in the table below:

Name	Type	Value
calcul...	CreditCheck...	[REDACTED]
acco	string	[REDACTED] 001
com	string	[REDACTED] IBM
cred	int	[REDACTED] 11
date	string	[REDACTED] 04/15/2016

- __ n. If time permits, you can test the additional test cases. If you enter the company name AbcCo or TestCo, the credit score that is returned is 1. If you enter the company name ACME, the credit score that is returned is 6.

 **Note**

To rerun a test, click the **Invoke** icon on the toolbar.



- ___ o. Close the test client tab. When you are prompted to save the test trace, click **No**.
- ___ 2. Remove the projects and (optionally) stop the server.
 - ___ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
 - ___ b. Click **Remove All** and click **Finish**. Wait until no messages appear in the IBM Integration Designer status bar.
 - ___ c. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Stop** to stop the server.
- ___ 3. Close IBM Integration Designer.

End of exercise

Exercise review and wrap-up

In this exercise, you imported an external WSDL file into IBM Integration Designer. You tested the connectivity with the web service and implemented SCA components in the IBM Integration Designer test environment. Finally, you used a web service export to expose an existing IBM Process Server application.

Exercise 5. Creating business objects and shared interfaces

What this exercise is about

In this exercise, you create business objects and interfaces that the components use in your end-to-end solution. These artifacts are created in a library for use by multiple modules.

What you should be able to do

After completing this exercise, you should be able to:

- Compare business objects between IBM Process Designer and IBM Integration Designer
- Implement business objects and define their elements
- Implement interfaces that use business objects as inputs and outputs

Introduction

In this exercise, you implement business objects with multiple attributes in the library you created previously. You also define abstract interfaces in the library that have request and response operations and use the defined business objects as inputs and outputs.

Requirements

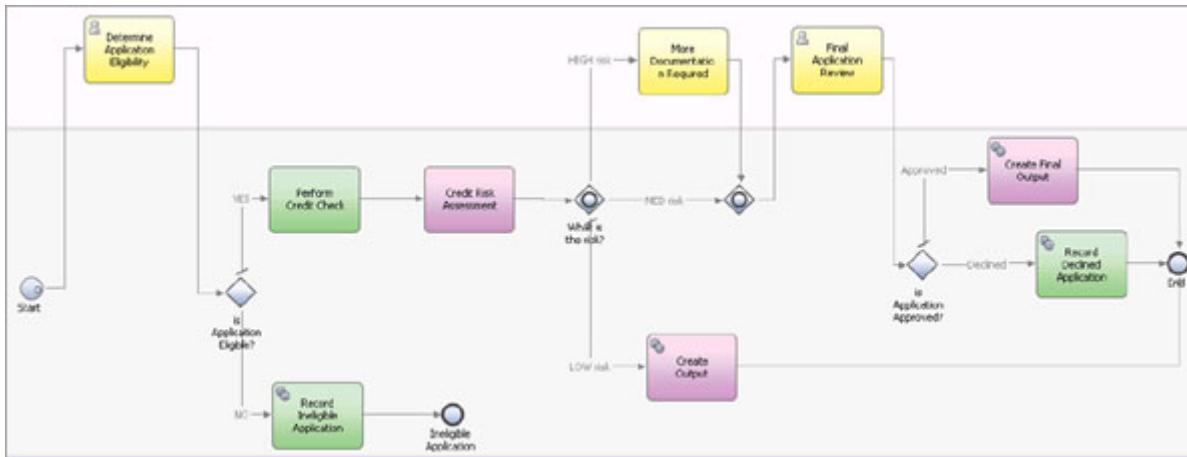
Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Exercise instructions

In this exercise, you use the business object editor in IBM Integration Designer to define a business object. You also use the interface editor to implement the RecordIneligibleApplication and MapToIneligible interface definitions. Because multiple modules can use business object definitions and interface definitions, you implement these definitions in a library, FoundationLibrary.

The account verification scenario, which was explored in previous exercises, uses the business object. The business objects in this exercise are used at different levels of the business process.

Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



- CustomerApplication.** This business object is the most widely used of the process application. A new application is submitted, and it is the goal of the process to handle that customer application. This business object is created for you.
- IneligibleApplication.** This business object is used if the application is not eligible (as determined by the **DetermineApplicationEligibility** activity, which sets the *eligibleApplication* field of the **CustomerApplication** business object). You create this business object.
- CreditCheckRequest.** In a previous exercise, you created a service component, which called a web service. The purpose of that web service was to check the customer's credit. That web service requires a specific business object. You might recall that **CreditCheckRequest** was imported with the web service definition in a previous exercise.
- Message.** The response of the business process application is to return this Message business object. This business object is created for you.

Part 1: Compare business objects between IBM Process Designer and IBM Integration Designer

In the account verification business process diagram model, which was built and demonstrated in IBM Process Designer, business objects were constructed for demonstration purposes. However, these business objects are needed as part of your overall solution as well.

In this part of the exercise, you examine the business objects that were created in IBM Process Designer and compare them to your requirements for your business process in IBM Integration Designer. You must first start the IBM Process Center repository.

- 1. Start the Process Center deployment manager. If the Process Center deployment manager, the node agent, and the cluster are already running, then skip to step 4 to start the Process Designer.
 - a. On your Windows desktop in the lab environment, select the shortcut that is titled **Start Process Center deployment manager**. It might be necessary to select the shortcut to completely view its description. Double-click the shortcut or press Enter to start the server.

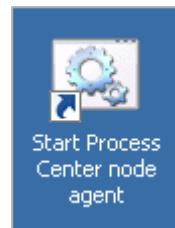


A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts.

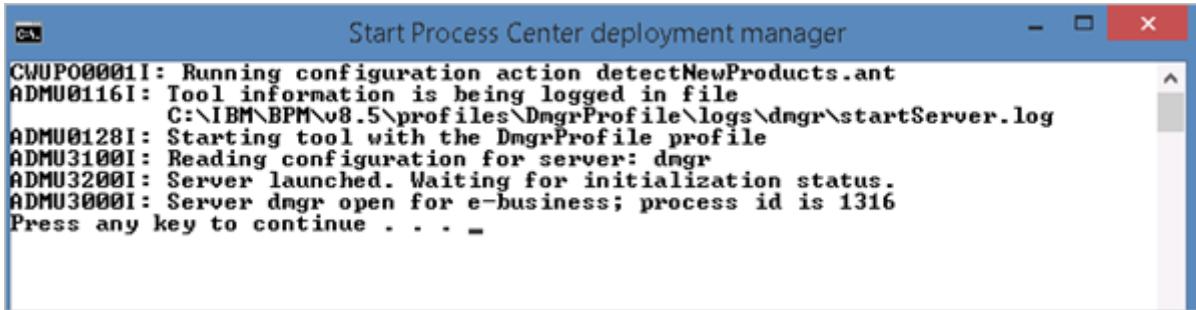
- b. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.

```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
          C:\IBM\BPM\v8.5\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 1316
Press any key to continue . . . =
```

- 2. Start the Process Center node agent.
 - a. On your Windows desktop, select the shortcut that is titled **Start Process Center node agent**. Double-click the shortcut or press Enter to start the server.



- ___ b. A DOS command window is displayed. Press any key to continue when prompted.



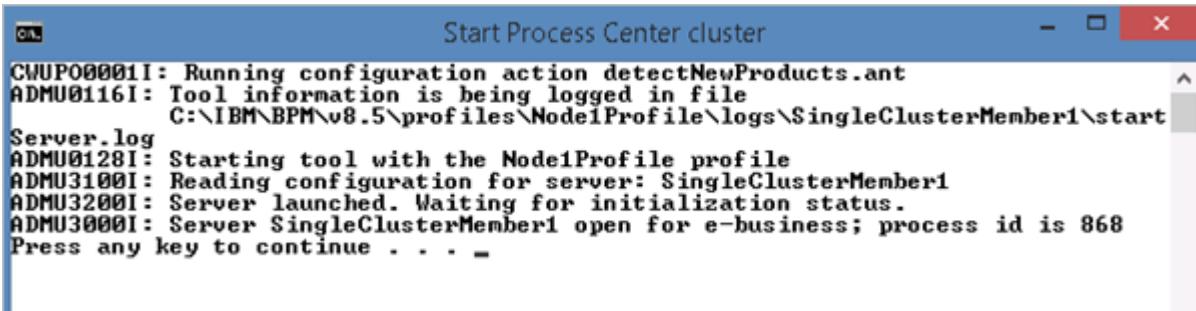
```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
  C:\IBM\BPM\v8.5\profiles\dmgrProfile\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the DmgrProfile profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 1316
Press any key to continue . . .
```

- ___ 3. Start the single cluster.

- ___ a. On your Windows desktop, select the shortcut that is titled **Start Process Center Cluster**. Double-click the shortcut or press Enter to start the cluster member.



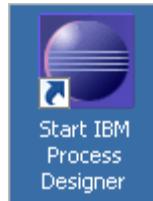
- ___ b. A DOS command window is displayed. Press any key to continue when prompted.



```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
  C:\IBM\BPM\v8.5\profiles\Node1Profile\logs\SingleClusterMember1\startServer.log
ADMU0128I: Starting tool with the Node1Profile profile
ADMU3100I: Reading configuration for server: SingleClusterMember1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server SingleClusterMember1 open for e-business; process id is 868
Press any key to continue . . .
```

- ___ 4. Start IBM Process Designer.

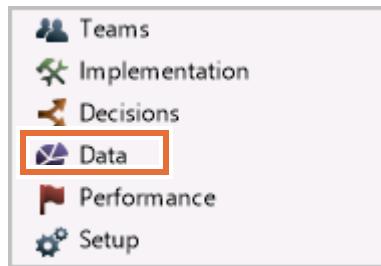
- ___ a. To start IBM Process Designer, double-click the **Start IBM Process Designer** icon on the desktop.



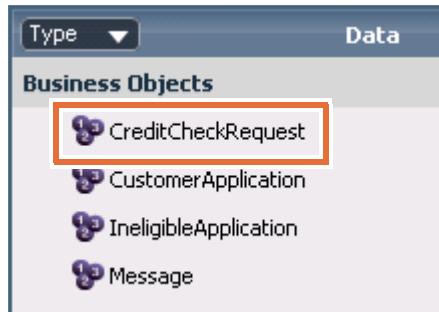
- ___ b. Enter `pcdeadmin` in the **User name** field and `web1sphere` in the **Password** field, and click **Log In**. After a few moments, IBM Process Designer starts.



- ___ 5. When the **Security Alert** window is displayed, click **Yes** to proceed.
- ___ 6. Click **Yes** one more time to proceed.
- ___ 7. If prompted, close the **Getting Started with IBM Process Designer 8.5.7.0** window by clicking **X**.
- ___ 8. When IBM Process Designer starts, the IBM Process Center perspective is displayed, displaying the process applications in the IBM Process Center repository. Verify that you are in the **Process Apps** tab. If you are not, then it might be necessary to click the Process Center icon at the top.
- ___ 9. Next to the **Account Verification Skeleton (AVS)** process application, click the **Open in Designer** link.
- ___ 10. Examine the business objects in the **Account Verification Skeleton** business process application.
 - ___ a. From the menu in the left, click **Data**.



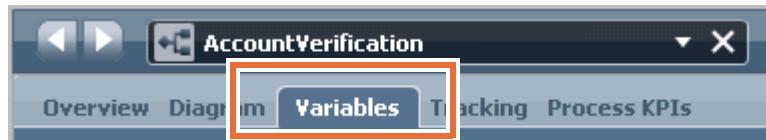
- ___ b. A list is displayed with a set of business objects. Double-click **CreditCheckRequest**.



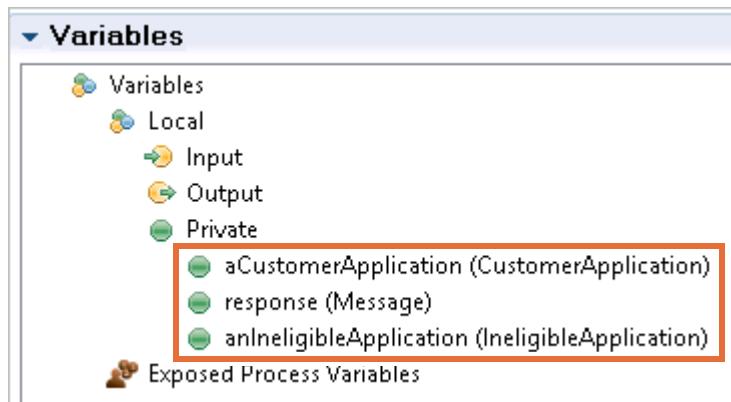
- ___ c. Compare the parameters of this business process to the WSDL file that you imported in a previous lab. The names of the types are slightly different, but they are similar data types.

The screenshot displays two side-by-side views of the 'CreditCheckRequest' parameters. On the left, within the 'Parameters' section of the Business Object list, are four parameters: accountNumber (String), companyName (String), creditScore (Integer), and dateRequested (String). On the right, within the WSDL definition, are four corresponding elements: accountNumber string, companyName string, creditScore int, and dateRequested string. The WSDL view includes a search bar labeled '<Click to filter...>' and a legend indicating element types (blue square for element).

- ___ d. In IBM Process Designer, close the **CreditCheckRequest** business object by clicking X.
___ e. Take some time to explore the other business objects and examine their parameters.
11. Examine how business objects are used in the sample process application.
- ___ a. In the IBM Process Designer window, click **Processes** and double-click **AccountVerification** to open the process in the editor.
___ b. In the editor window, select the **Variables** tab. These variables refer to the entire process application.



- ___ c. Three variables are named in this process application. A customer application (**CustomerApplication**) serves as the input, and a response (**Message**) or an ineligible application (**IneligibleApplication**) is returned.



- ___ d. Feel free to explore the variables. When you are done, click the Process Center icon at the upper right to switch to the Process Center view.
 - ___ e. Click **File > Exit** to close IBM Process Designer.
- ___ 12. For performance reasons, you might want to stop the Process Center environment because it is not used until a later exercise. If performance is not an issue, skip this step and keep the Process Center running.
- ___ a. Double-click the **Stop Process Center cluster** icon. Wait until it stops successfully. Press any key to close the command window.
 - ___ b. Double-click the **Stop Process Center node agent** icon. Wait until it stops successfully. Press any key to close the command window.
 - ___ c. Double-click the **Stop Process Center deployment manager** icon. Wait until it stops successfully. Press any key to close the command window.

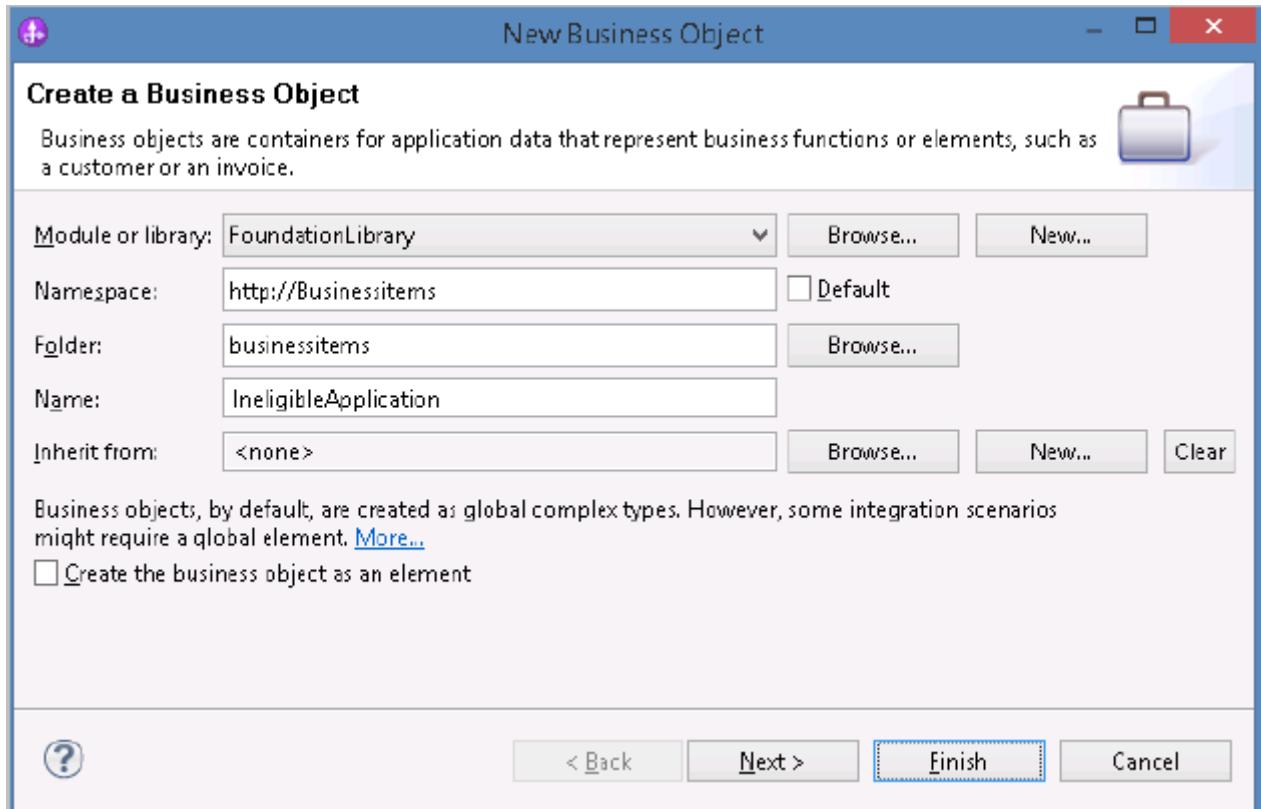
Part 2: Implement business objects and define their elements

In this portion of the exercise, you create the `IneligibleApplication` business object that is required for the account verification business process application.

To create the business object:

- ___ 1. Open the Exercise 5 workspace.
- ___ a. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
- ___ b. Double-click the shortcut that is labeled **Exercise 5**. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of the Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
- ___ c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
- ___ d. Close the **Getting Started** tab.

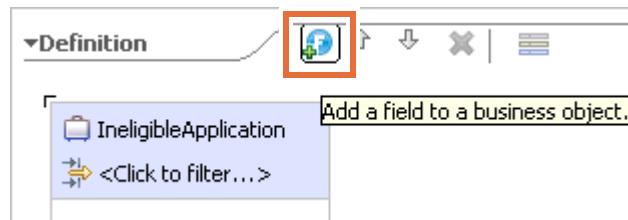
2. Use the `http://Businessitems` namespace to create a business object, `IneligibleApplication`, in the `businessitems` folder.
- Expand **FoundationLibrary**, right-click **Data**, and click **New > Business Object** from the menu.
 - At the “Create a Business Object” dialog box, take the following actions:
 - Verify that the **Module or library** field is set to `FoundationLibrary`.
 - Clear the **Default** check box, and change the **Namespace** field to: `http://Businessitems` (note the case).
 - Type `businessitems` in the **Folder** field (note the case), or click **Browse** to select it.
 - Type `IneligibleApplication` in the **Name** field.



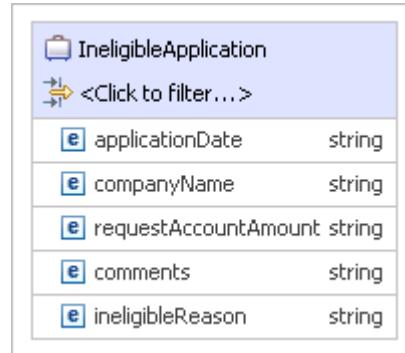
- c. Accept the remaining default options and click **Finish**. The **IneligibleApplication** business object opens in the business object editor.

3. Add the following fields to the **IneligibleApplication** business object:

- applicationDate (string)
 - companyName (string)
 - requestAccountAmount (int)
 - comments (string)
 - ineligibleReason (string)
- a. In the business object editor, with **IneligibleApplication** selected, click the **Add a field to a business object** icon.



- b. Change the default field name `field1` to: `applicationDate`
- c. With **IneligibleApplication** selected, click the **Add a field to a business object** icon.
- d. Change the default field name `field1` to: `companyName`
- e. With **IneligibleApplication** selected, click the **Add a field to a business object** icon.
- f. Change the default field name `field1` to: `requestAccountAmount`
- g. With **IneligibleApplication** selected, click the **Add a field to a business object** icon.
- h. Change the default field name `field1` to: `comments`
- i. With **IneligibleApplication** selected, click the **Add a field to a business object** icon.
- j. Change the default field name `field1` to: `ineligibleReason`
- k. The completed business object resembles the following figure:



However, as indicated in a previous step, the `requestAccountAmount` type is expected to be an integer, not a string.

- l. Right-click the **requestAccountAmount** field and click **Show In > Properties View** from the menu.
- m. In the Properties view, click **Browse** beside the **Type** field.

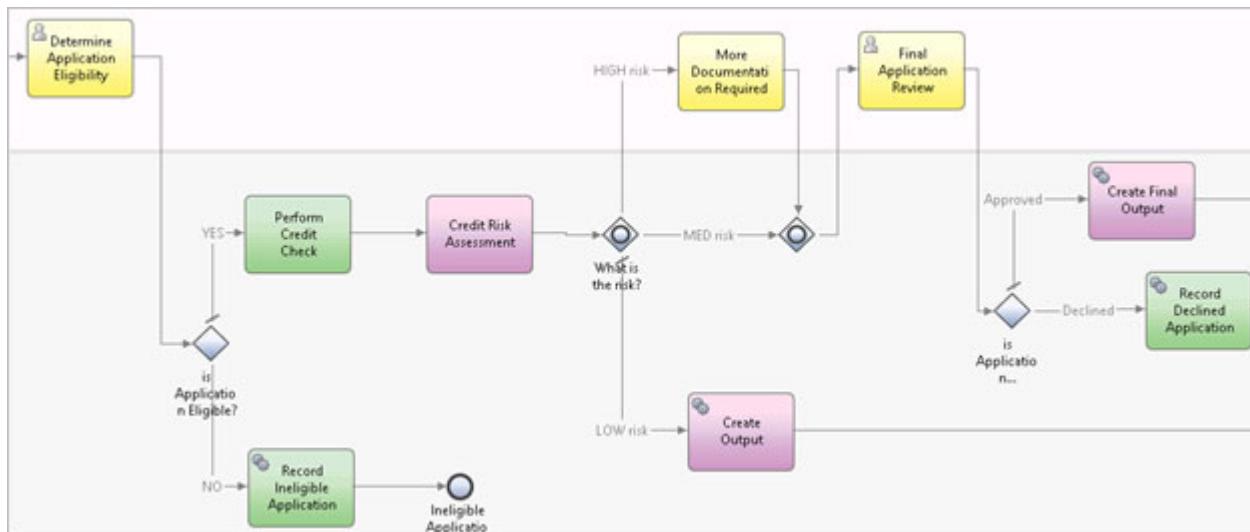
- ___ n. Select **int** from the list and click **OK**.

Element - requestAccountAmount		
Description	Name:	requestAccountAmount
Documentation	Type:	int
Application Info	Default value:	

- ___ o. Press **Ctrl + S** to save your changes and **Ctrl + W** to close the editor.

Part 3: Implement interfaces that use business objects as inputs and outputs

In this portion of the exercise, you create shared interfaces that your modules use. Because the components you create in subsequent exercises use the same business object inputs and outputs, you can reuse the interfaces by defining them in a shared library. To save time, many of the shared interfaces were created for you in the workspace.



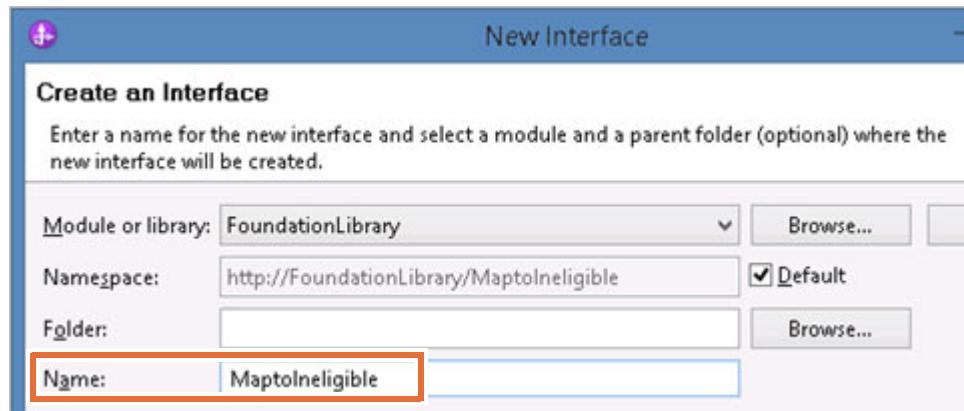
In this exercise, you create two interfaces:

- **MaptoIneligible:** This interface takes the customer's application as its input and produces an ineligible application as the output. The implementation of an activity does the transformation.
- **RecordIneligibleApplication:** This interface takes an ineligible application and returns a message. The implementation of an activity records the ineligible application.

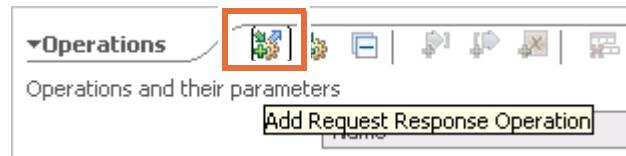
To create the shared interfaces:

- ___ 1. Create an interface **MaptoIneligible** with a request and response operation that is named **InputCriterion** that uses the **CustomerApplication** business object as the input (named **Input**), and **IneligibleApplication** as the output (named **Output**).
- ___ a. Expand **FoundationLibrary**, right-click **Interfaces**, and click **New > Interface** from the menu.

- ___ b. In the “Create an Interface” dialog box, type `MaptoIneligible` in the **Name** field.



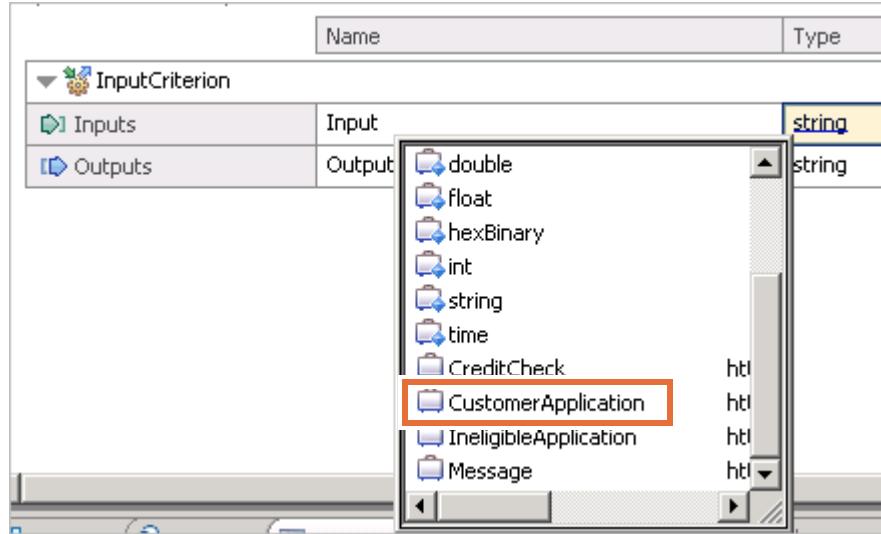
- ___ c. Accept the remaining default options and click **Finish** to open the **Maptolineligible** interface in the interface editor.
 ___ d. In the interface editor, click the **Add Request Response Operation** icon.



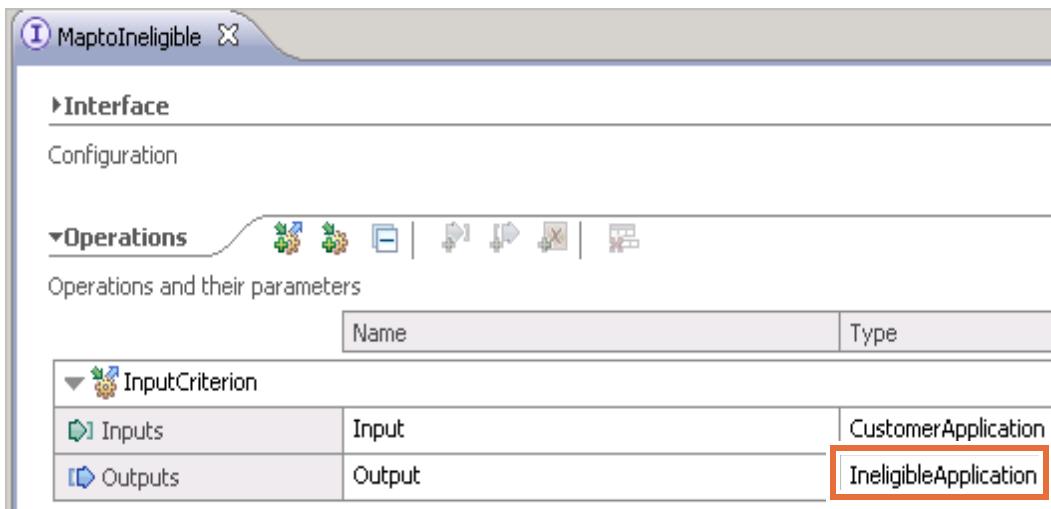
- ___ e. Change the default operation name from `operation1` to: `InputCriterion`
 ___ f. Change the default input name from `input1` to `Input` (note the case).
 ___ g. Change the output name from `output1` to `Output` (note the case).

Name	Type
InputCriterion	
<code>Inputs</code>	<code>Input</code>
<code>Outputs</code>	<code>Output</code>

- __ h. Change the default input type by clicking the **string** link and selecting the **CustomerApplication** business object from the menu.

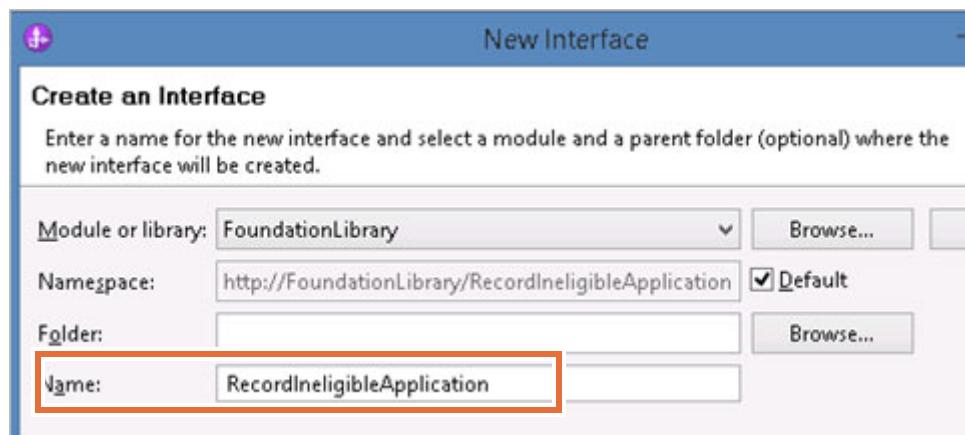


- __ i. Click the link for the **Output** type and select the **IneligibleApplication** business object from the menu.

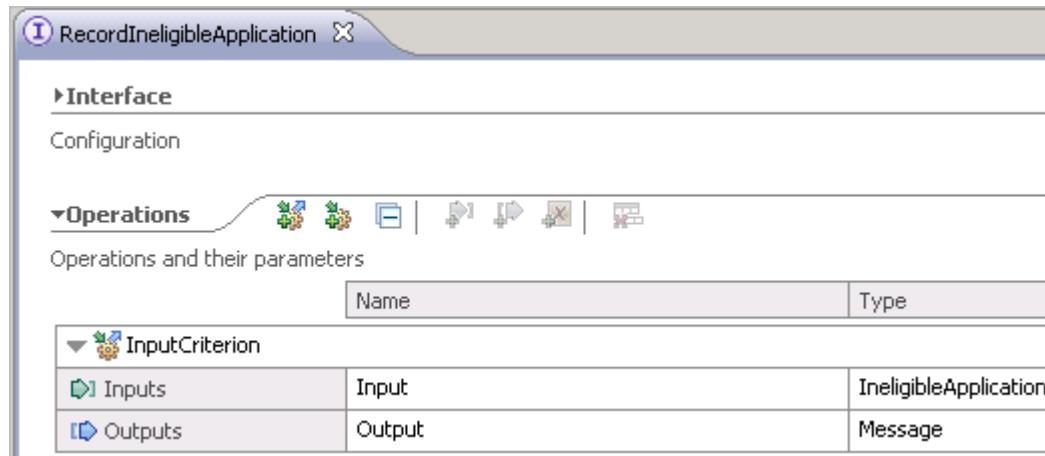


- __ j. Press Ctrl + S to save your changes.
 __ k. Press Ctrl + W to close the editor.
- __ 2. Create another interface `RecordIneligibleApplication` with a request and response operation that is named `InputCriterion` that uses the **IneligibleApplication** business object as the input (named `Input`) and the **Message** business object as the output (named `output`).
 __ a. Expand **FoundationLibrary**, right-click **Interfaces**, and click **New > Interface** from the menu.

- ___ b. In the “Create an Interface” dialog box, type RecordIneligibleApplication in the **Name** field.



- ___ c. Accept the remaining default options and click **Finish**.
- ___ d. Click the **Add Request Response Operation** icon and change the operation name from `operation1` to: `InputCriterion`
- ___ e. Change the default input name from `input1` to `Input` (note the case).
- ___ f. Change the output name from `output1` to `Output` (note the case).
- ___ g. Change the default input type by clicking the **string** link and selecting the **IneligibleApplication** business object from the menu.
- ___ h. Click the link for the **Output** type and select the **Message** business object from the menu. The completed interface resembles the following figure:



- ___ i. Press **Ctrl + S** to save your changes.
- ___ j. Press **Ctrl + W** to close the editor.
- ___ 3. If time permits, examine the remaining business objects and interfaces that are created for you. In the remaining exercises, you build components that are going to use these artifacts. For more information about the artifacts, see the exercise appendixes.
- ___ 4. Close IBM Integration Designer.

End of exercise

Exercise review and wrap-up

In this exercise, you created business objects and interfaces that the components use in your end-to-end solution.

Exercise 6. Creating a business process, part I

What this exercise is about

In this exercise, you use IBM Integration Designer to begin creating a complex business process. You create a long-running business process, process variables, interface partners, and reference partners.

What you should be able to do

After completing this exercise, you should be able to:

- Create a business process
- Implement WS-BPEL interface partners and reference partners
- Create process variables
- Compare business processes between IBM Integration Designer and IBM Process Designer

Introduction

The Web Services Business Process Execution Language (WS-BPEL) provides syntax for specifying the behavior of a business process in a platform-independent manner. It is used to coordinate a series of service invocations to fulfill a business task. Although the language provides conditional and control logic, most of the work is done through the invoked services.

Combining WS-BPEL with the SCA programming model allows for the coordination of SCA services into much larger units of work. Individual SCA services can be brought together and can benefit from the advanced capabilities of event handling, fault handling, and compensation.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Exercise instructions

In a previous exercise, you examined a business process application skeleton that is named Account Verification in IBM Process Designer. In this portion of the exercise, you re-create this process in IBM Integration Designer as a long-running process that is named AccountVerification.

AccountVerification automates the process of opening customer accounts that are based on SOA principles. As a result, the process of opening customer accounts is more flexible (easily accommodating future changes), and existing services can be reused instead of rewritten.



Questions

Why create the business process in IBM Integration Designer?

The business process application that you examined in IBM Process Designer was incomplete. Although the activities were in place, and in a previous exercise, you wired them together, each individual activity did not accomplish anything: they had no implementations.

In this exercise and the subsequent exercises, you build this process in IBM Integration Designer; but more importantly, you focus on building the implementations for the individual activities.

Part 1: Create a business process

In this portion of the exercise, you create a long-running process that is named AccountVerification in FoundationModule. This process forms the core of the end-to-end application that you are developing in the exercises. The process is designed to receive a customer application, do a credit check, and send the application forward for approval, supplemental documentation, or rejection. For more information about how the solution functions, see the exercise appendixes.



Note

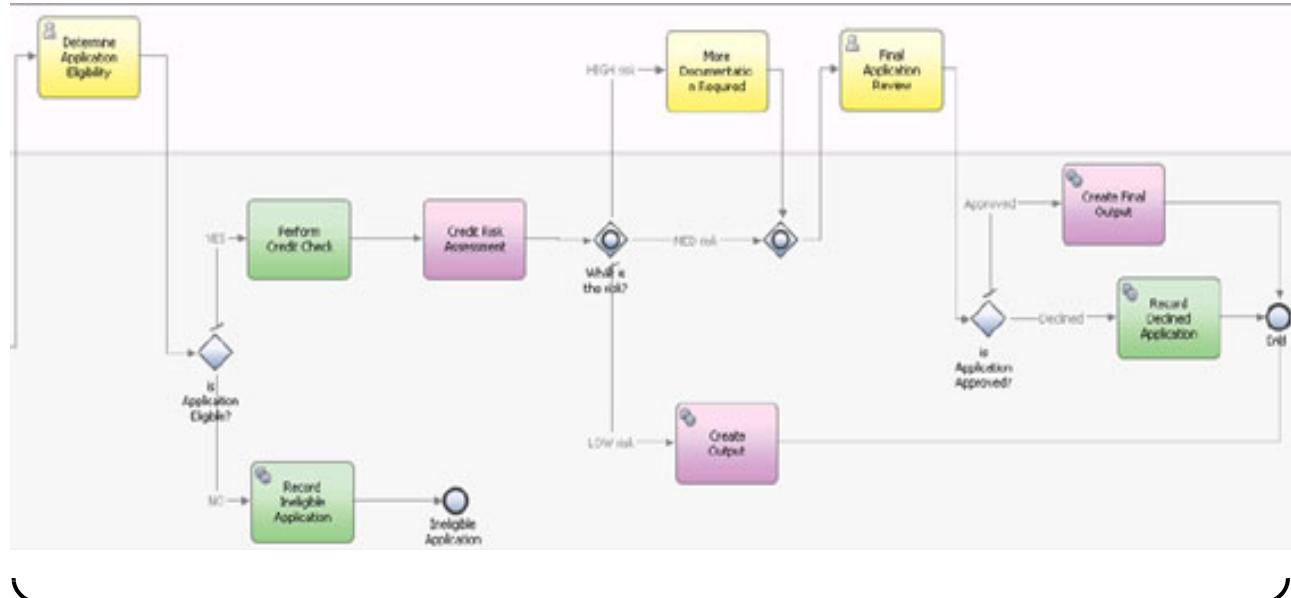
Recall the skeleton of the process model that you completed in a previous exercise.

Review the process narrative that is used to create that process model: it serves as the narrative for your new business process.

- When the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and end the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is done against the customer's credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.
- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.

- If the customer is determined to be high risk, the customer must submit more documentation, and then the customer must seek final approval for the application.
- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

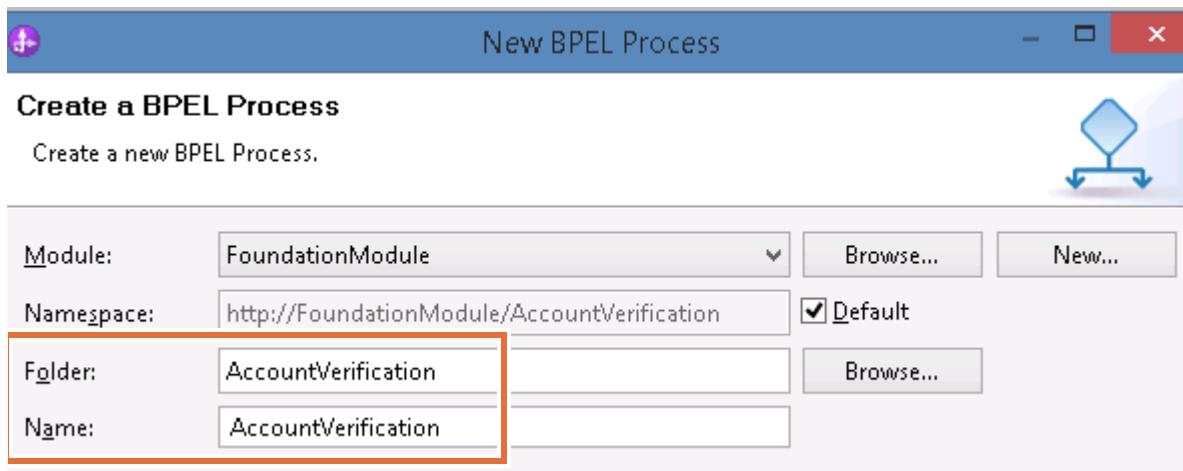
Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



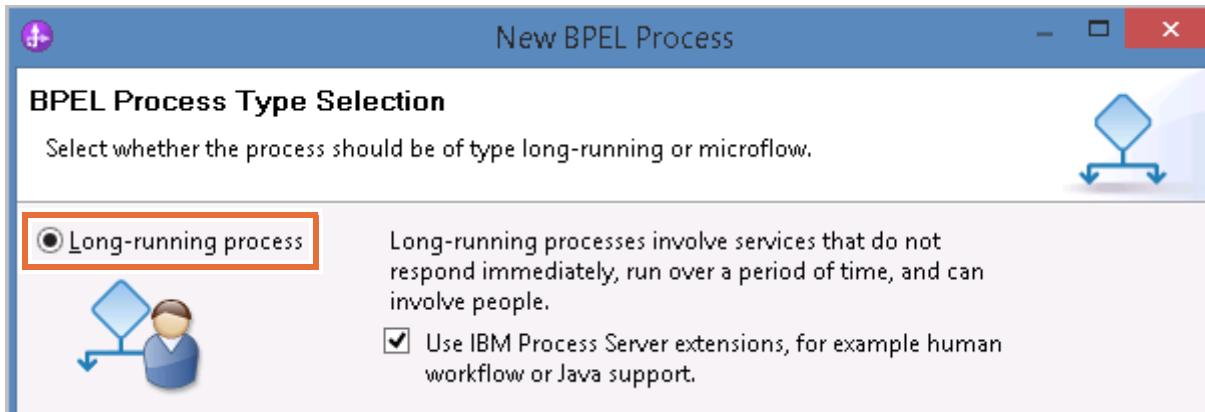
To create a long-running business process:

1. Open the Exercise 6 workspace.
 - a. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
 - b. Double-click the shortcut that is labeled **Exercise 6**. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of the Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
 - c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
 - d. Close the **Getting Started** tab.
2. Create a business process in **FoundationModule** named **AccountVerification** that uses the **AccountVerification** interface.
 - a. Expand **FoundationModule**, right-click **Integration Logic**, and click **New > BPEL Process** from the menu.

- __ b. In the New BPEL Process dialog box, at the “Create a BPEL Process” dialog box, enter the following information:
- Type AccountVerification in the **Folder** field.
 - Type AccountVerification in the **Name** field.



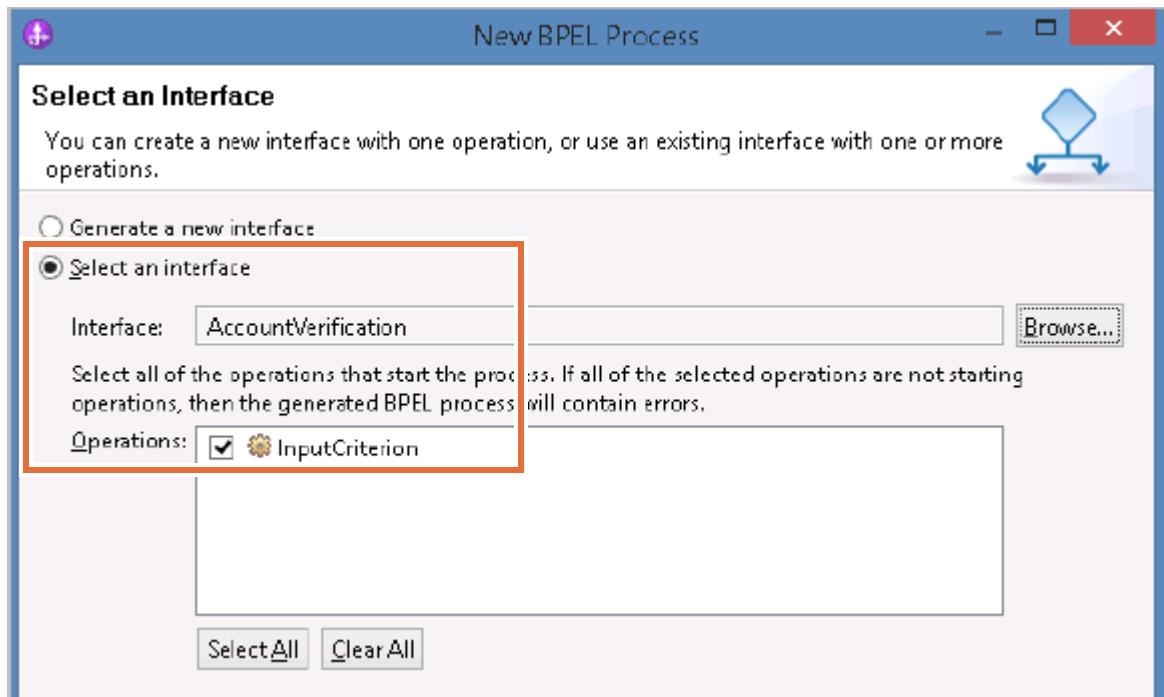
- __ c. Click **Next**.
- __ d. At the BPEL Process Type Selection dialog box, verify that **Long-running process** is selected, and verify that **Use IBM Process Server extensions** is selected.



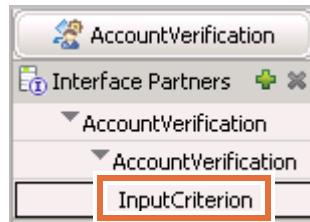
For information about the capabilities that are lost by clearing the **Use IBM Process Server extensions** option, see the IBM Business Process Manager product documentation.

- __ e. Click **Next**.

- ___ f. At the “Select an Interface” dialog box, take the following actions:
- Click **Select an interface**.
 - Click **Browse**, select the **AccountVerification** interface from the list, and click **OK**.



- ___ g. Accept the remaining default options and click **Finish** to open the process editor.
- ___ 3. Examine the **AccountVerification** interface partner. This interface becomes a WS-BPEL interface partner.
- ___ a. In the **Interface Partners** portion of the tray, expand **AccountVerification > AccountVerification** and select the **InputCriterion** operation.

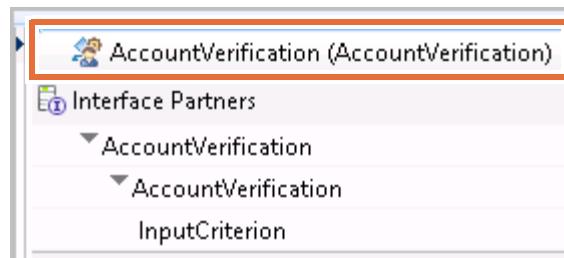


- ___ b. Switch to the **Properties** view.

- ___ c. Note the properties of the interface and the business object inputs and outputs. The interface accepts a CustomerApplication business object as the input and returns a Message business object as the output.

Description		
Interface	AccountVerification	
Operation	InputCriterion	
Input	Input	CustomerApplication
Output	Output	Message

- ___ 4. Examine the properties of the long-running process. You learn more about these properties later in this course.
- ___ a. In the tray on the far right of the screen, click **AccountVerification (AccountVerification)**.

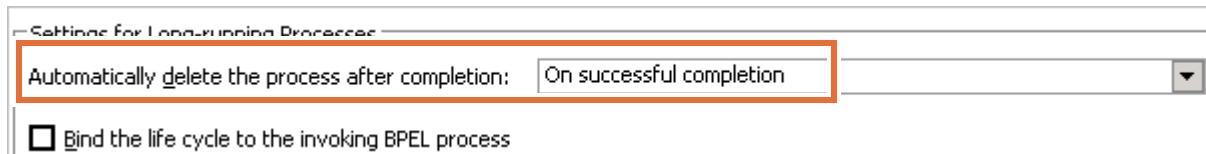


- ___ b. Switch to the **Details** tab in the **Properties** view.
- ___ c. Examine the detailed properties in the **General Settings** section.
- ___ d. Note the value in the **Select the date when the process becomes valid** field. Process instances that are created from this date forward use this process template. Processes currently “in flight” use the process template available when they were started unless you upgrade them to a new template.

Description	General Settings
Details	BPEL process type: Long-running process Uses IBM Process Server extensions
Administration	<input checked="" type="checkbox"/> Select the date when the process becomes valid: <input type="text" value="April 20, 2016"/> <input type="button" value="Select Date"/>
Java Imports	
Defaults	

- ___ e. Examine the properties in the **Settings for Long-running Processes** section.

- ___ f. Note the value in the **Automatically delete the process after completion** field. By default, when a process is successfully completed, process-related data for the completed instance is deleted from the Business Process Choreographer database.

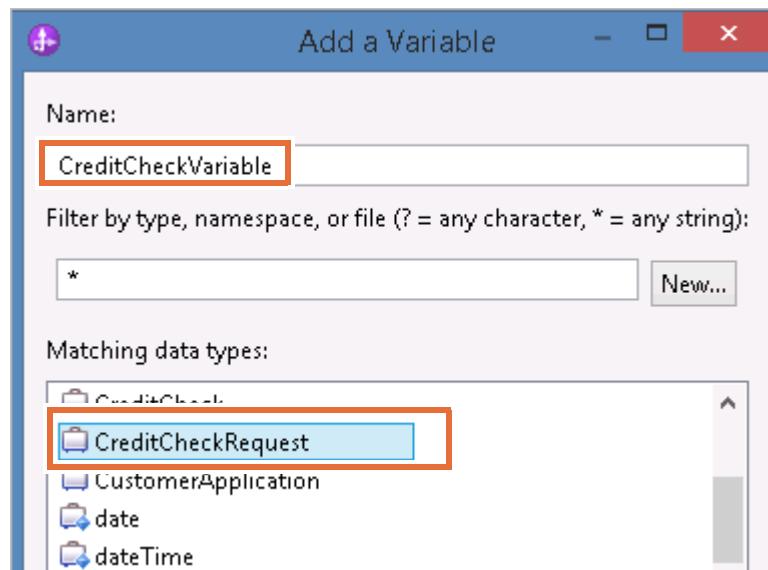


Part 2: Create process variables

In this portion of the exercise, you add several global variables to the business process. The variables store business objects for manipulation by the process activities. A variable belongs to the scope in which it is declared. If it is declared in the global process scope, then it is a global variable and is visible to the process as a whole. If it is declared in a nested scope, it is called a scoped or local variable and is visible only by objects within the scope in which it is created.

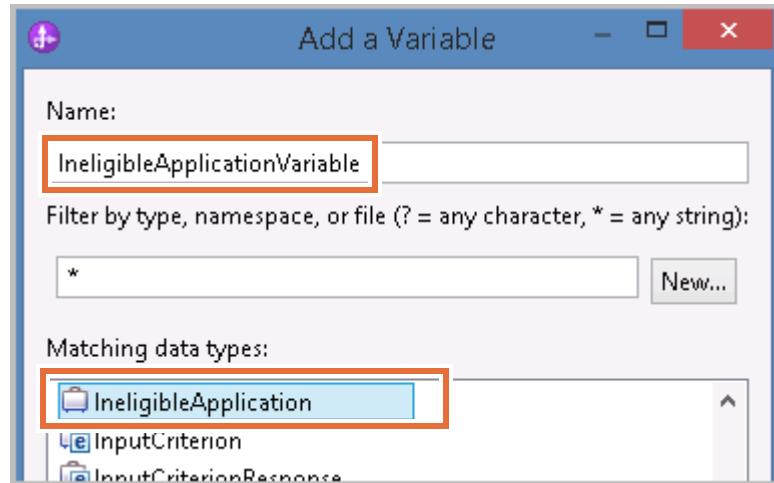
To declare process variables:

- ___ 1. Add a global variable that is named `CreditCheckVariable` that stores a **CreditCheckRequest** business object.
 - ___ a. In the **Variables** section of the tray, click the plus sign (+) to add a variable.
 - ___ b. In the “Add a Variable” dialog box, change the **Name** to: `CreditCheckVariable`
 - ___ c. In the “Matching data types” section, select the **CreditCheckRequest** business object.

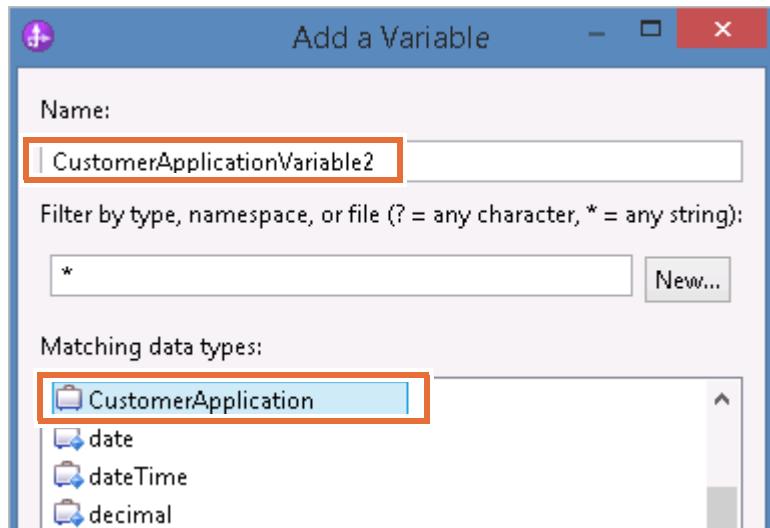


- ___ d. Click **OK**.
- ___ 2. Add a global variable that is named `IneligibleApplicationVariable` that stores an **IneligibleApplication** business object.
 - ___ a. In the **Variables** section of the tray, click the plus sign (+) to add a variable.

- __ b. In the “Add a variable” dialog box, change the **Name** to:
IneligibleApplicationVariable
- __ c. In the “Matching data types” window, select the **IneligibleApplication** business object.

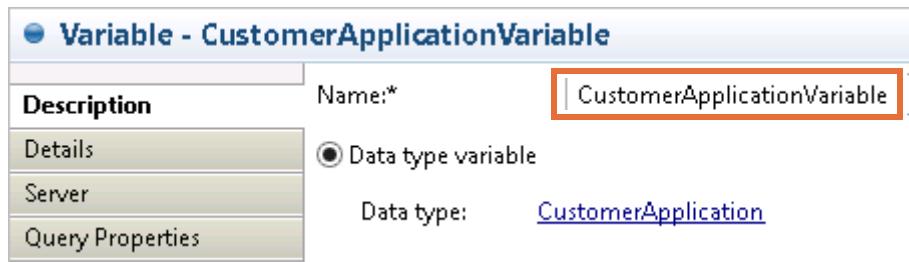


- __ d. Click **OK**.
- __ 3. Add a global variable that is named `CustomerApplicationVariable2` that stores a **CustomerApplication** business object.
 - __ a. In the **Variables** section, click the plus sign (+) to add a variable.
 - __ b. In the “Add a variable” dialog box, change the **Name** to:
`CustomerApplicationVariable2`
 - __ c. In the “Matching data types” section, select the **CustomerApplication** business object.

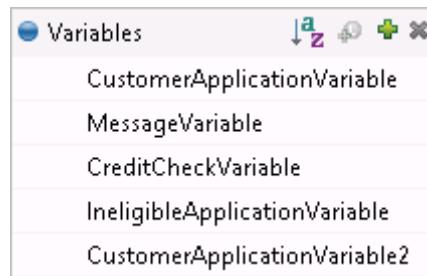


- __ d. Click **OK**.
- __ 4. Save your changes.

5. Change the name of the `Input` variable to `CustomerApplicationVariable` and change the name of the `Output` variable to `MessageVariable`. These values correspond to the type of business objects that form the input and output for the process interface.
- a. In the **Variables** section of the tray, select the `Input` variable and switch to the **Description** tab in the **Properties** view.
 - b. Type `CustomerApplicationVariable` in the **Name** field and press Enter. The `Input` variable is renamed to `CustomerApplicationVariable`. It is not necessary to refactor it.



- c. In the **Variables** section of the tray, select the `Output` variable and switch to the **Description** tab in the **Properties** view.
- d. Type `MessageVariable` in the **Name** field and press Enter. The `Output` variable is renamed to `MessageVariable`. It is not necessary to refactor it.
- e. Save your changes. Your process variables look like the following screen capture:



Questions

Why are there so many variables?

In the AccountVerification skeleton, which you examined in IBM Process Designer, only three variables were used: `CustomerApplication`, a response (`Message`), and `IneligibleApplication`.

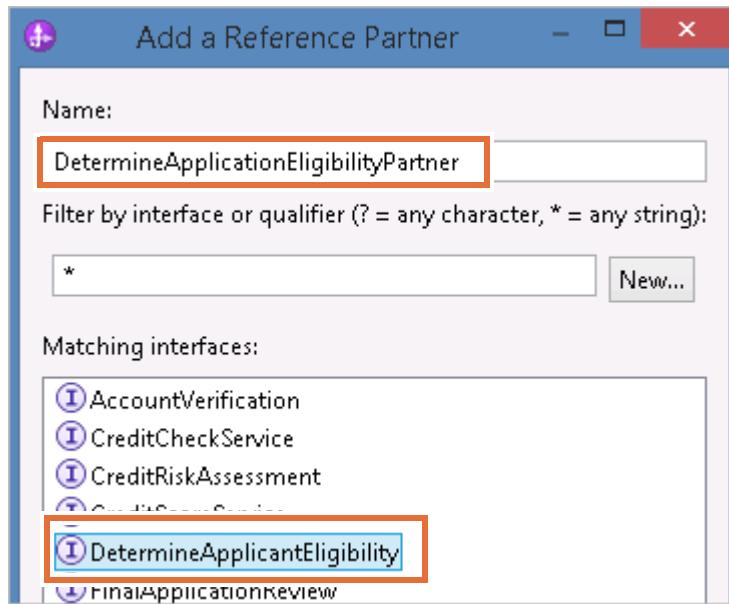
However, in that example, it was presumed that all the implementations would be contained in the same process application. The same can be true for this new business process, but that would result in a highly compartmentalized solution. The strength of IBM Integration Designer is to use the Service Component Architecture, so other implementations might be placed in separate modules. Because implementations are spread throughout other modules, it maximizes reusability, but necessitates intercommunication, and passing variables, between modules.

Part 3: Implement WS-BPEL interface partners and reference partners

In this portion of the exercise, you add WS-BPEL reference partners to your process. Reference partners represent the service interfaces your WS-BPEL activities call during process execution.

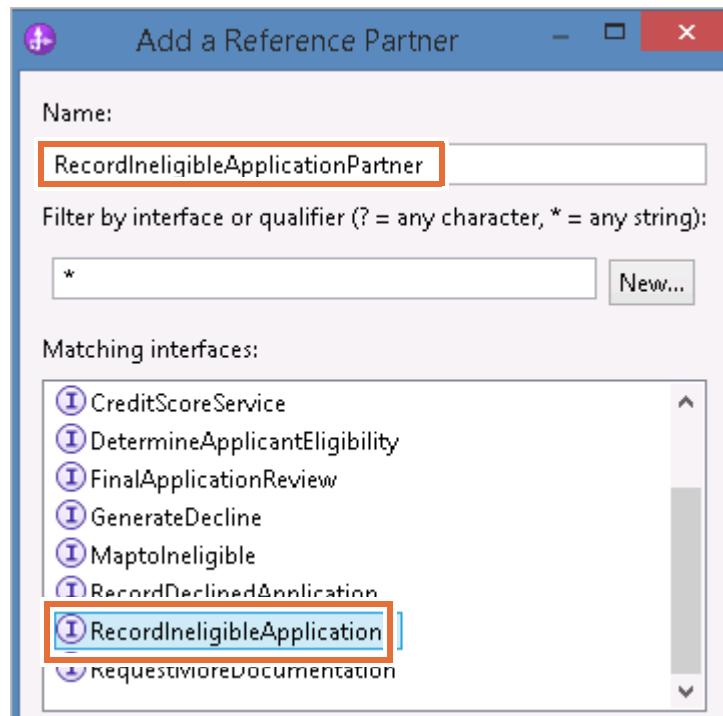
To create reference partners:

- 1. Add the `DetermineApplicationEligibilityPartner` reference partner with the **DetermineApplicantEligibility** interface.
 - a. In the **Reference Partners** section of the tray, click the plus sign (+) icon to add a reference partner.
 - b. In the “Add a Reference Partner” dialog box, change the **Name** to: `DetermineApplicationEligibilityPartner`
 - c. In the **Matching interfaces** section, select the **DetermineApplicantEligibility** interface from the list.



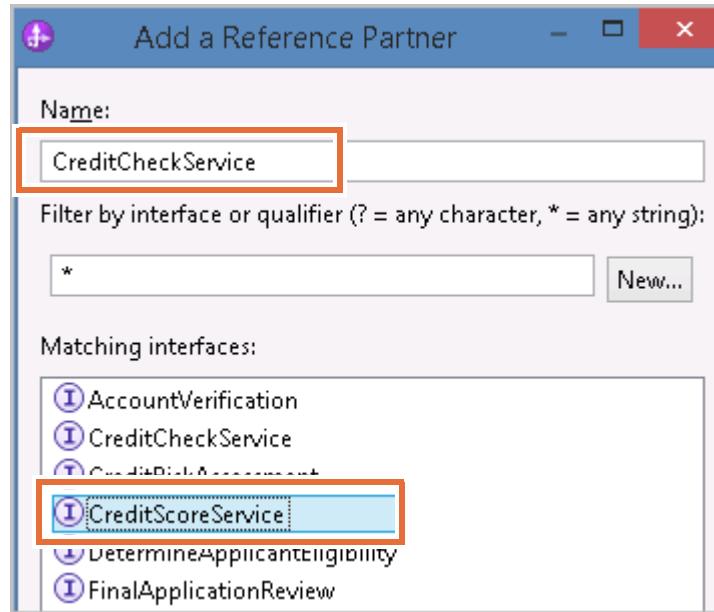
- d. Click **OK**.
- 2. Add the `RecordIneligibleApplicationPartner` reference partner with the **RecordIneligibleApplication** interface.
 - a. In the **Reference Partners** section of the tray, click the plus sign (+) icon to add a reference partner.
 - b. In the “Add a Reference Partner” dialog box, change the **Name** to: `RecordIneligibleApplicationPartner`

- c. In the **Matching interfaces** section, select the **RecordIneligibleApplication** interface from the list.

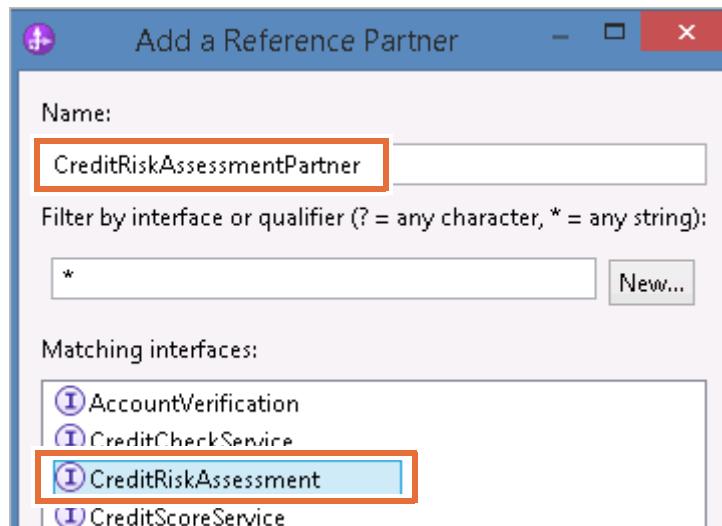


- d. Click **OK**.
- 3. Add the `CreditCheckServicePartner` reference partner with the **CreditScoreService** interface.
- a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
- b. In the “Add a Reference Partner” dialog box, change the **Name** to:
`CreditCheckServicePartner`

- ___ c. In the **Matching interfaces** section, select the **CreditScoreService** interface from the list.

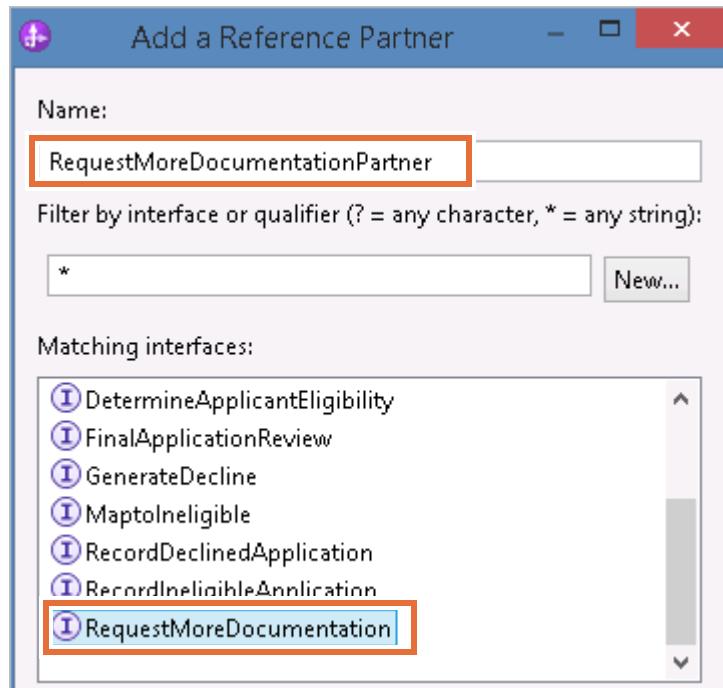


- ___ d. Click **OK**.
- ___ 4. Add the `CreditRiskAssessmentPartner` reference partner with the **CreditRiskAssessment** interface.
- ___ a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
 - ___ b. In the "Add a Reference Partner" dialog box, change the **Name** to: `CreditRiskAssessmentPartner`
 - ___ c. In the **Matching interfaces** section, select the **CreditRiskAssessment** interface from the list.



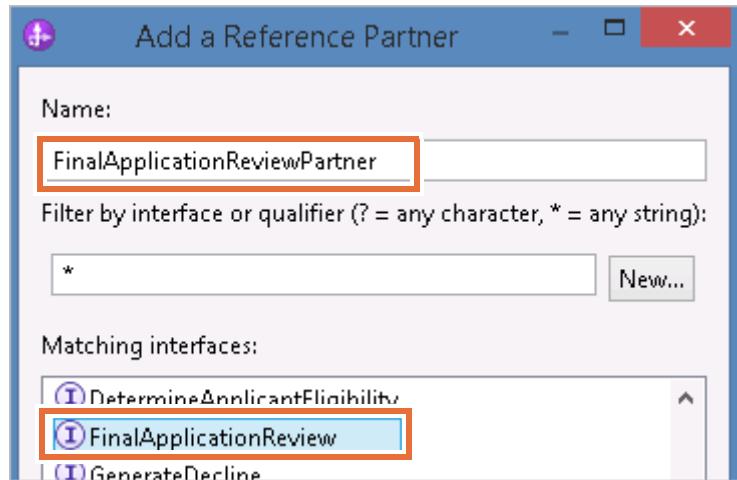
- ___ d. Click **OK**.

- 5. Add the `RequestMoreDocumentationPartner` reference partner with the **RequestMoreDocumentation** interface.
- a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
 - b. In the “Add a Reference Partner” dialog box, change the **Name** to: `RequestMoreDocumentationPartner`
 - c. In the **Matching interfaces** section, select the **RequestMoreDocumentation** interface from the list.



- d. Click **OK**.
- 6. Add the `FinalApplicationReviewPartner` reference partner with the **FinalApplicationReview** interface.
- a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
 - b. In the “Add a Reference Partner” dialog box, change the **Name** to: `FinalApplicationReviewPartner`

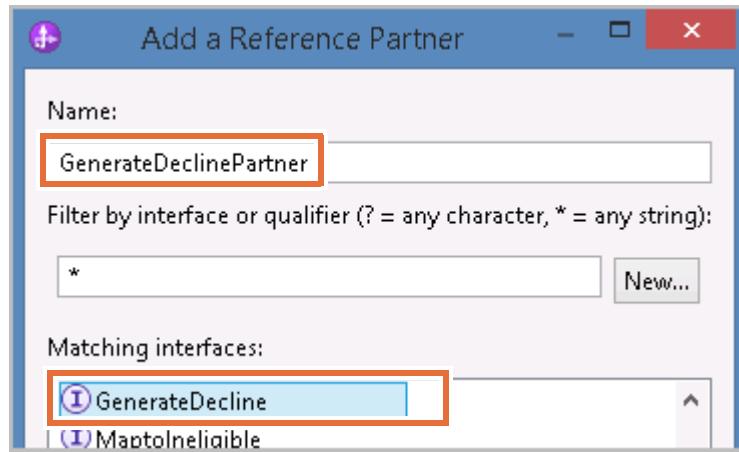
- ___ c. In the **Matching interfaces** section, select the **FinalApplicationReview** interface from the list.



- ___ d. Click **OK**.

- ___ 7. Add the `GenerateDeclinePartner` reference partner with the **GenerateDecline** interface.

- ___ a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
- ___ b. In the “Add a Reference Partner” dialog box, change the **Name** to:
GenerateDeclinePartner
- ___ c. In the **Matching interfaces** section, select the **GenerateDecline** interface from the list.

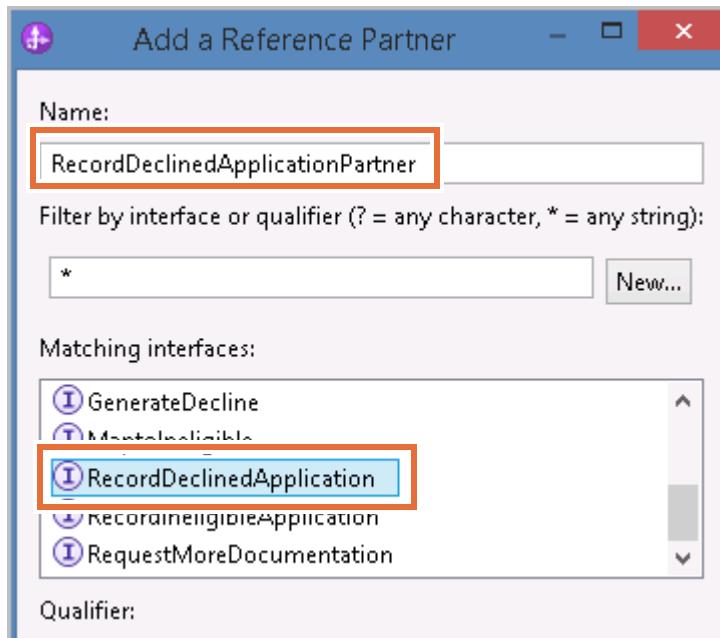


- ___ d. Click **OK**.

- ___ 8. Add the `RecordDeclinedApplicationPartner` reference partner with the **RecordDeclinedApplication** interface.

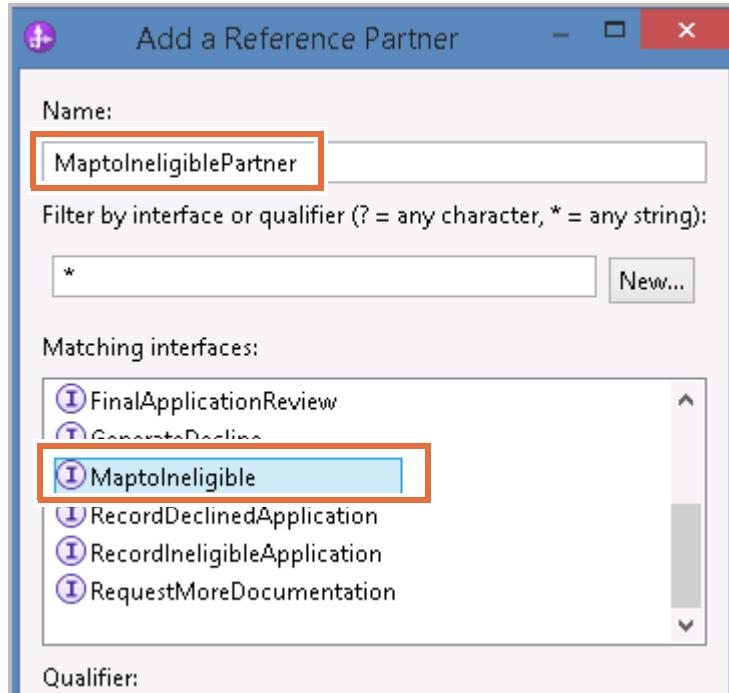
- ___ a. In the Reference Partners section, click the plus sign (+) icon to add a reference partner.
- ___ b. In the “Add a Reference Partner” dialog box, change the **Name** to:
RecordDeclinedApplicationPartner

- c. In the **Matching interfaces** section, select the **RecordDeclinedApplication** interface from the list.



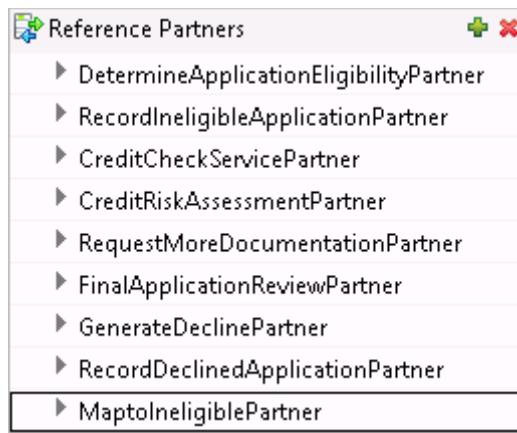
- d. Click **OK**.
- 9. Add the **MaptoIneligiblePartner** reference partner with the **MaptoIneligible** interface.
- a. In the **Reference Partners** section, click the plus sign (+) icon to add a reference partner.
- b. In the "Add a Reference Partner" dialog box, change the **Name** to:
MaptoIneligiblePartner

- __ c. In the **Matching interfaces** section, select the **Maptoleligible** interface from the list.



- __ d. Click **OK**.

- __ 10. Save your changes. Verify that your **Reference Partners** section resembles the following figure:



Questions

What is the purpose of these reference partners?

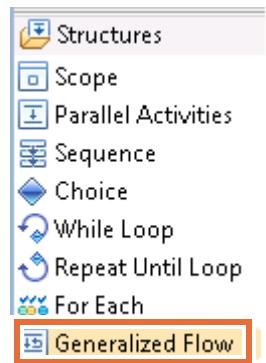
Remember that in a previous statement, it is the expressed intention of this solution to maximize reusability by spreading the implementations of business components through several modules. To facilitate this reusability, you must build “communication conduits” between the modules. These conduits, import, and export components are realized in the implementations as abstract concepts: reference and interface partners.

Part 4: Define a business process

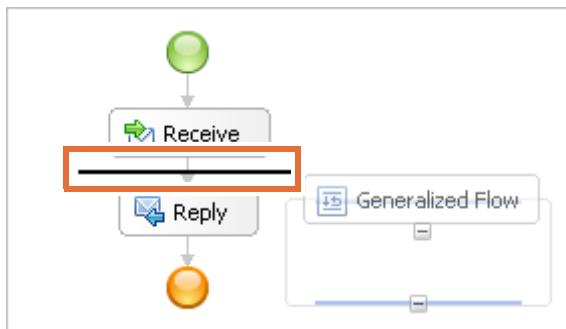
In this portion of the exercise, you create the structured activity (a generalized flow) that acts as the container for the simple and complex BPEL activities you implement in the next exercise.

To define the process:

- 1. Add a **Generalized Flow** activity to the process and place the **Receive** and **Reply** activities in it.
 - a. Expand **Structures** in the **Palette** and click **Generalized Flow**.

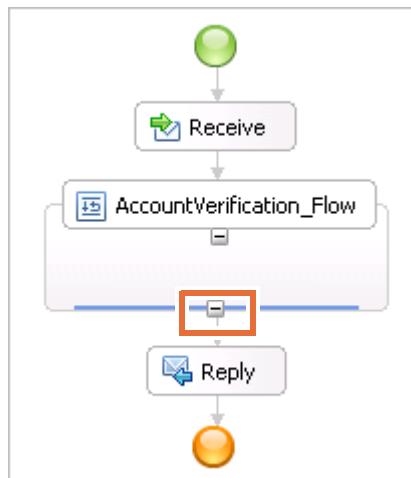


- b. Click the space under the **Receive** activity in the editor. A solid black line indicates the insertion point.



- c. Ensure that the generalized flow activity is selected, and switch to the **Description** tab in the **Properties** view.
- d. Change the **Display Name** from `GeneralizedFlow` to: `AccountVerification_Flow`. For this activity (and all other activities), you alter the Display Name and not the Name. The **Name** field does not support spaces, and the two fields do not have to be the same.

- __ e. Expand the size of **AccountVerification_Flow** by selecting it and dragging the handle at the bottom of the activity.

**Note**

Structured activities (containing other activities) can be expanded or collapsed either by clicking the plus and minus icons at the top of the activity or by double-clicking the activity.

- __ f. Right-click the process editor and click **Align Parallel Activities Contents Automatically** (to clear it). You can manually control the placement of the activities in the diagram by clearing automatic alignment.
- __ g. Save your changes. Ignore any errors in the **Problems** view.
- __ 2. Change the **Display Name** of the **Receive** activity to: Account Verification Receive
- __ a. In the editor, right-click the **Receive** activity, and click **Show In > Properties View** from the menu.
- __ b. On the **Description** tab in the **Properties** view, change the **Display Name** to Account Verification Receive and press Enter.

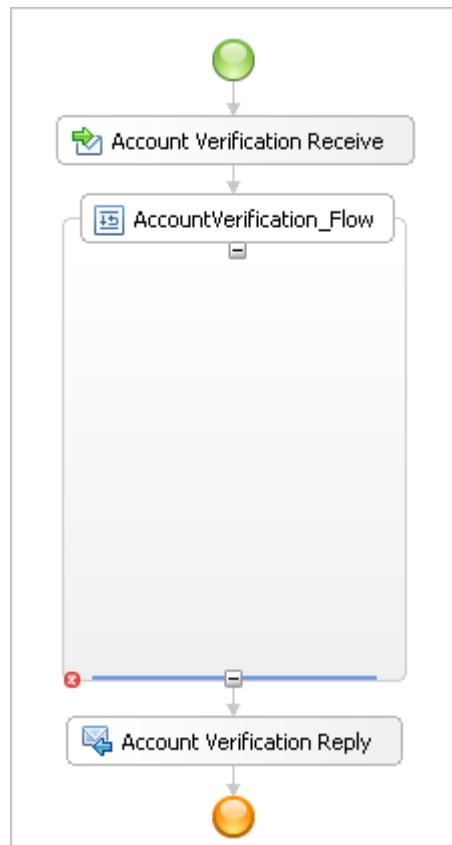
Receive - Account Verification Receive (Receive)	
Description	Name: Receive
Details	Display name: Account Verification Receive

- __ 3. Change the **Display Name** of the **Reply** activity to: Account Verification Reply
- __ a. In the editor, right-click the **Reply** activity, and click **Show In > Properties View** from the menu.

- ___ b. On the **Description** tab in the **Properties** view, change the **Display Name** to Account Verification Reply and press Enter.



- ___ c. Verify that your process looks like the following image:

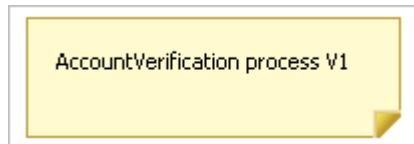


- ___ 4. Save your changes. Continue to ignore any errors in the **Problems** view.
 ___ 5. Add a sticky note to the process with the following text that identifies it as the first version of the process: AccountVerification process v1
 ___ a. In the palette, click the **Note** icon.

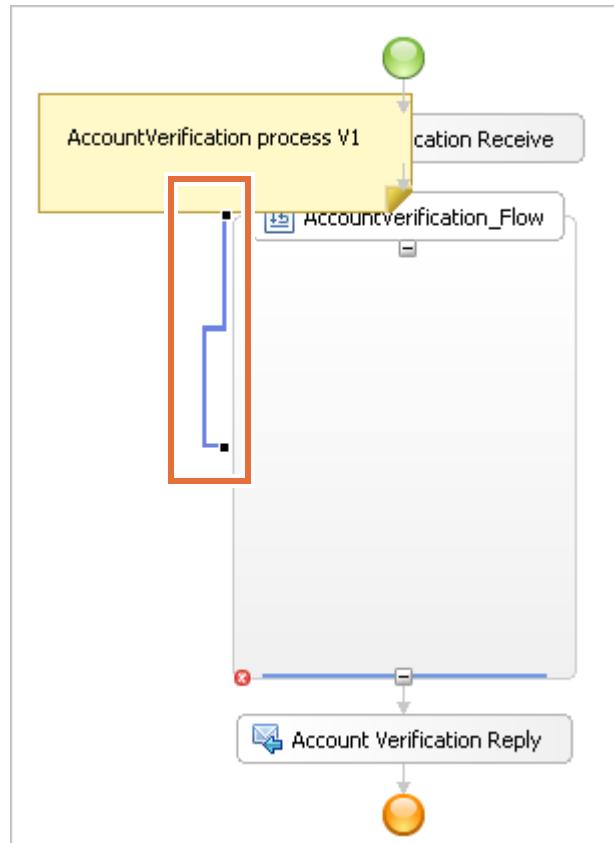


- ___ b. Click an empty space in the upper-left corner of the process diagram to add the sticky note.

- __ c. Type the following text in the note: AccountVerification process v1



- __ 6. Associate the sticky note with AccountVerification_Flow.
- __ a. Click a blank space on the canvas to escape edit mode in the sticky note.
 - __ b. Right-click the note and click **Add Association** from the menu.
 - __ c. Click the **AccountVerification_Flow** activity to wire the note to the structured activity.



To aid readability, you can right-click the canvas and toggle the **Hide Notes/Show Notes** option in the menu.

- __ 7. Add a snippet to AccountVerification_Flow.
- __ a. In the palette, expand **Basic Actions** and select **Snippet**.
 - __ b. Click **AccountVerification_Flow** to add the snippet to the diagram.

- __ c. In the **Properties** view on the **Description** tab, change the **Display name** to: Print Hello World
- __ d. Switch to the **Details** tab in the **Properties** view.
- __ e. Click **Java**. If you are prompted with a dialog box, click **Yes**.



Information

When adding code to a snippet element, you can add the code by using either the visual editor or the text editor, but not both.

- __ f. Copy the "Print Hello World" snippet code from C:\labfiles\Support Files\Ex6\ AccountVerification_Flow_snippet.txt.

```
*****  
// "Print Hello World" snippet  
*****  
  
System.out.println("[Java] AccountVerification_Flow - begins");  
System.out.println("[Java] Hello World !!");  
System.out.println("[Java] AccountVerification_Flow - ends");
```

- __ g. Paste the snippet in the expression window. Close the text file when you are done pasting the text. Alternatively, you can manually enter the following text:

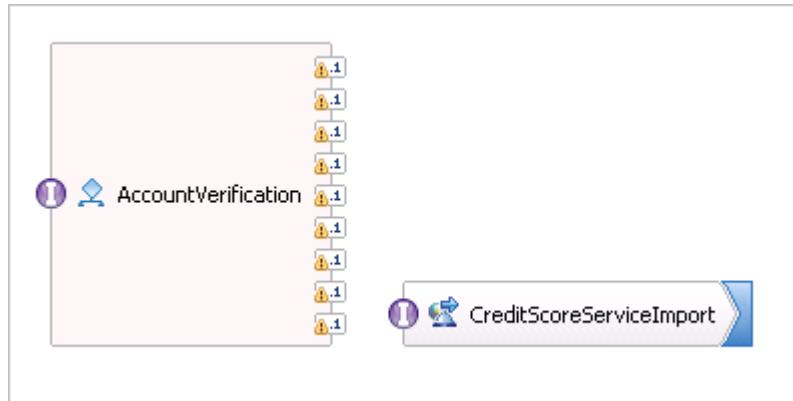
```
System.out.println("[Java] AccountVerification_Flow - begins");  
System.out.println("[Java] Hello World !!");  
System.out.println("[Java] AccountVerification_Flow - ends");
```

- __ h. To organize the contents for readability, right-click any blank space inside the **AccountVerification_Flow** activity and click **Arrange Parallel Activities Contents Automatically**.



- __ i. Save your changes.

- ___ 8. Add the **AccountVerification** service component to the **FoundationModule** assembly diagram.
- ___ a. In the Business Integration view, expand **FoundationModule**.
 - ___ b. Double-click **Assembly Diagram** to open the assembly editor.
 - ___ c. Expand **FoundationModule > Integration Logic > BPEL Processes > AccountVerification**.
 - ___ d. Drag the **AccountVerification** BPEL process onto the assembly diagram.

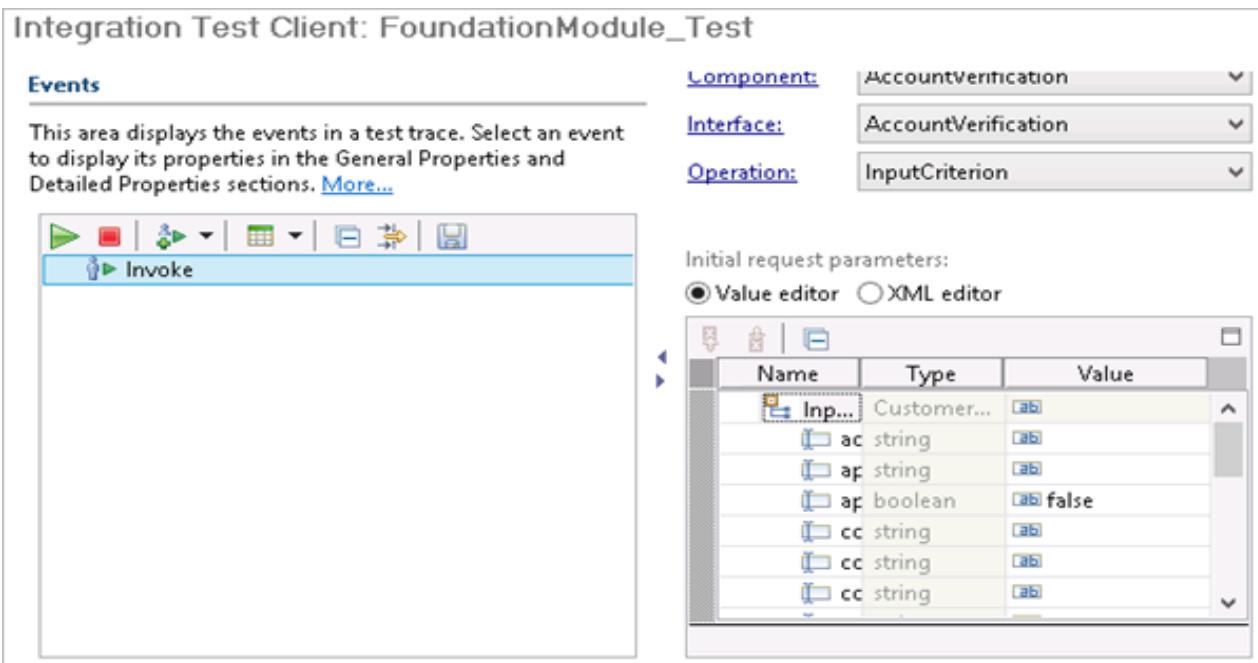


You do not need to wire the services on the assembly editor now. You import the referenced partners and wire them with the AccountVerification process in future exercises.

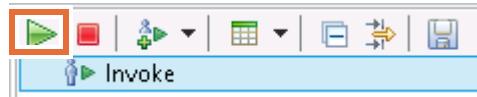
- ___ e. Save your changes.
- ___ 9. Test the **AccountVerification** BPEL process.
- ___ a. If necessary, in the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost**, and click **Start**. Wait for the startup process to complete before continuing.

The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.

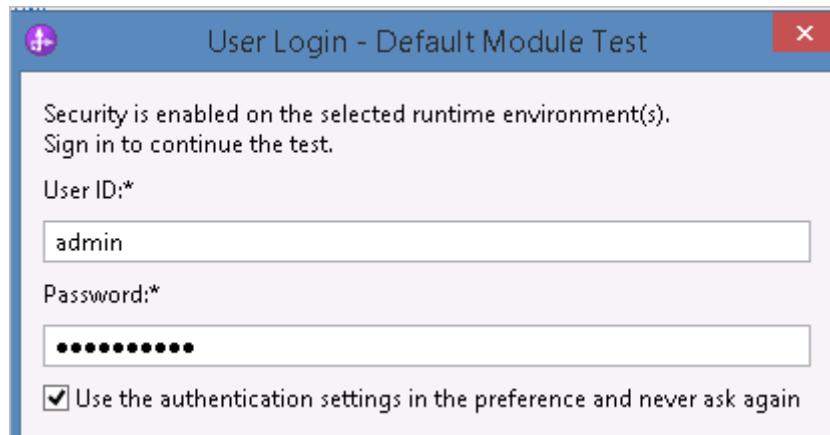
- ___ b. In the assembly diagram, right-click the **AccountVerification** BPEL process and click **Test Component** from the menu to open the integration test client.



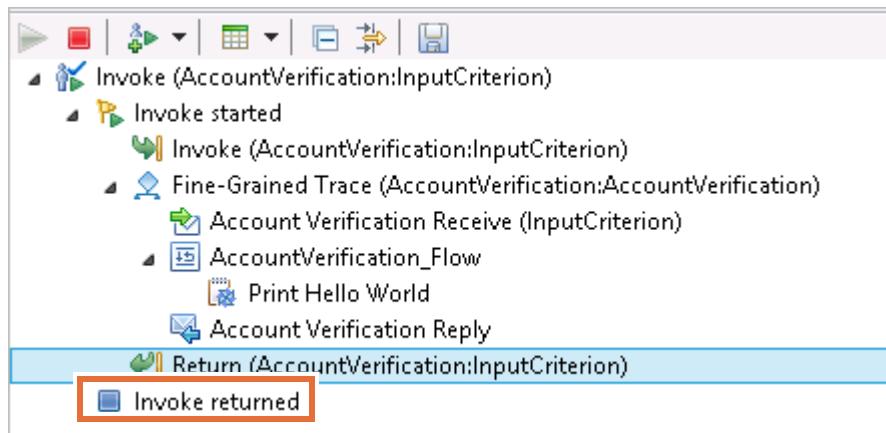
- ___ c. You do not need to input any test data now. Here you are testing the snippet inside the BPEL process that prints the messages in the console. Click **Continue** on the Events toolbar.



- ___ d. When you receive the “Select a Deployment Location” dialog box, select **IBM Process Server v8.5.7 at localhost**, select **Use this location as the default and do not ask again**, and click **Finish**.
- ___ e. When the User Login dialog box is displayed, select the **Use the authentication settings in the preference and never ask again** check box, and click **OK**.



- ___ f. Wait for the test to complete. When the test is finished, you see a stop node that is labeled "Invoke returned" in the Events window.



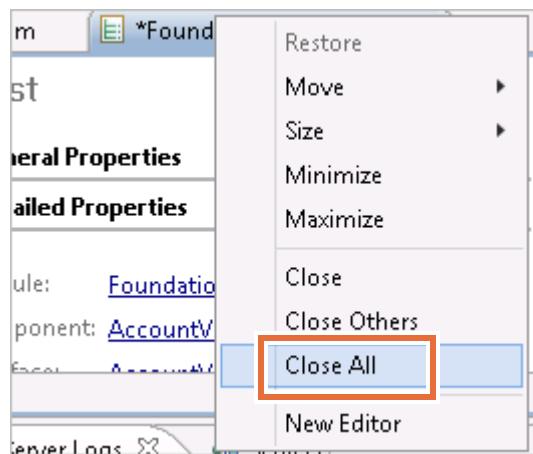
- ___ g. You also see messages in the **Server Logs** view that the **AccountVerification** BPEL process returns. Go to the **Server Logs** view and locate the following messages.

[Java] AccountVerification_Flow - begins
[Java] Hello World !!
[Java] AccountVerification_Flow - ends

Type	Time	Thread ID	Contents
Start invoke	Apr 21, 2016 11:07:45.091...	0000503d	Start of the asynchronous invocation of operation A...
Start corr	Apr 21, 2016 11:07:45.231...	00000056	Start of the component processing of operation Acc...
In BPI	Apr 21, 2016 11:07:45.575...	00000056	0748a3dd-4d40-4a64-54245fc79404 <TATF_R69...
Start I	Apr 21, 2016 11:07:46.419...	000002a2	Start of processing for BPEL process AccountVerific...
Lc	Apr 21, 2016 11:07:46.466...	000002a2	[Java] AccountVerification_Flow - begins
Lc	Apr 21, 2016 11:07:46.466...	000002a2	[Java] Hello World !!
Lc	Apr 21, 2016 11:07:46.466...	000002a2	[Java] AccountVerification_Flow - ends

- ___ 10. Remove the snippet from the **AccountVerification** BPEL process.
- Expand **FoundationModule** > **Integration Logic** > **BPEL Processes** > **AccountVerification**.
 - Double-click **AccountVerification**.
 - In the process editor, right-click the **Print Hello World** snippet and click **Delete** from the menu.
 - Save your changes.
- ___ 11. Remove the **AccountVerification** BPEL process from the **FoundationModule** assembly editor.
- In the Business Integration view, expand **FoundationModule**.
 - Double-click **Assembly Diagram** to open the assembly editor.
 - Right-click **AccountVerification** service and click **Delete** from the menu.

- ___ d. In the Confirm Delete dialog box, do **not** select **Also delete the AccountVerification.bpel implementation**. Click **Yes**.
- ___ 12. Remove the projects and (optionally) stop the server.
 - ___ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
 - ___ b. Click **Remove All** and click **Finish**. Wait until no messages appear in the IBM Integration Designer status bar.
 - ___ c. (Optional) In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Stop** to stop the server.
- ___ 13. Click **File > Save All** to save your changes. Continue to ignore any errors in the **Problems** view; they are resolved in later exercises.
- ___ 14. Close all open editors in IBM Integration Designer. You can right-click any tab in an open editor and click **Close All**.

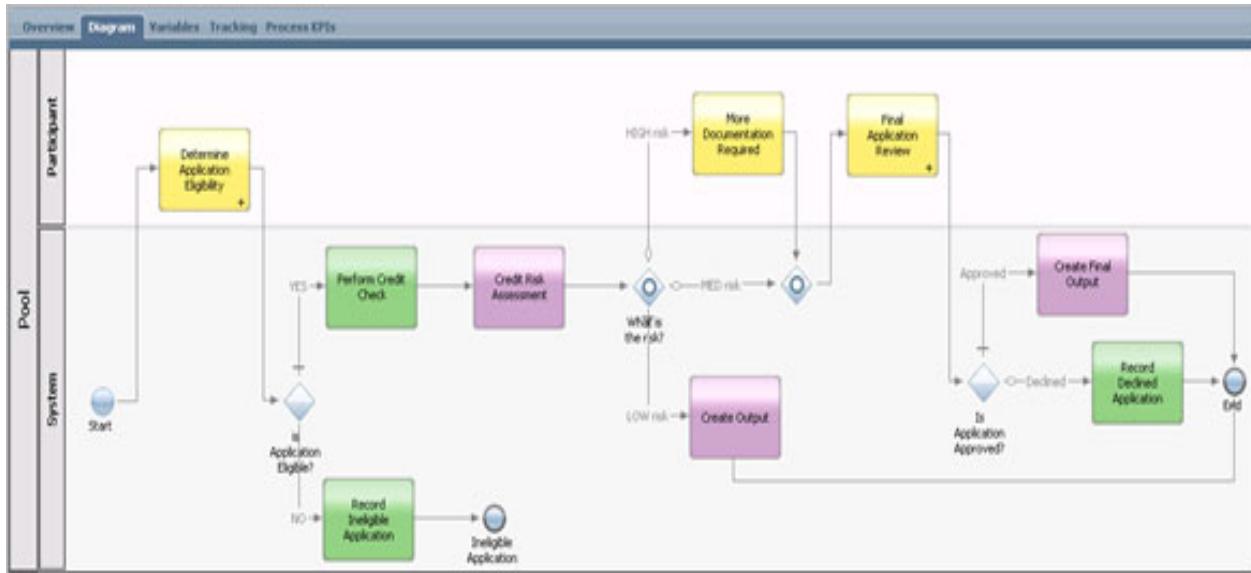


- ___ 15. Close IBM Integration Designer.

Part 5: Compare business processes between IBM Integration Designer and IBM Process Designer

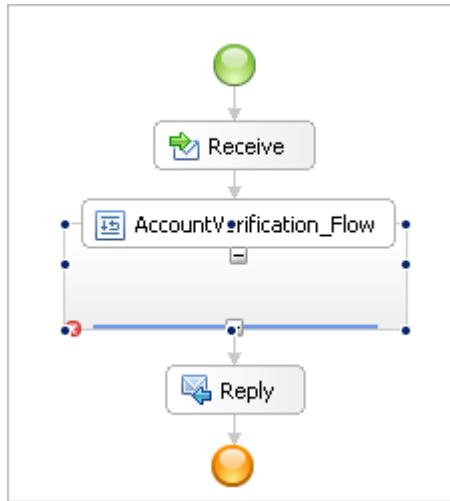
The business process that you created thus far is incomplete. Thus far, you created the basic business process and declared variables, interface partners, and reference partners in preparation for implementing the individual activities of the business process. You add these activities and their implementations in the exercises that follow.

Consider the following process application as your “to be” model.



Do not be concerned about reading the small text in the previous diagram. The purpose of the solution diagram is to view the connection wiring and the flow.

Thus far, you built start and end points only, and the container in which the other activities are going to be built:



You build the remaining activities and their implementations in the exercises that follow.

End of exercise

Exercise review and wrap-up

In this exercise, you used IBM Integration Designer to create a business process. You also created process variables, interface partners, and reference partners.

Exercise 7. Creating a business process, part II

What this exercise is about

In this exercise, you use IBM Integration Designer to implement a complex WS-BPEL business process. You use a combination of simple and structured activities to choreograph the service invocations that are designed to implement the account verification process.

What you should be able to do

After completing this exercise, you should be able to:

- Implement basic BPEL activities in a business process
- Implement structured activities in a business process
- Compare the BPEL to the BPD in IBM Process Designer

Introduction

The Web Services Business Process Execution Language (WS-BPEL) provides syntax for specifying the behavior of a business process in a platform-independent manner. It is used to coordinate a series of service invocations to fulfill a business task. Although the language provides conditional and control logic, most of the work is done through the invoked services.

Combining WS-BPEL with the SCA programming model allows for the coordination of SCA services into much larger units of work. Individual SCA services can be brought together and can benefit from the advanced capabilities of WS-BPEL.

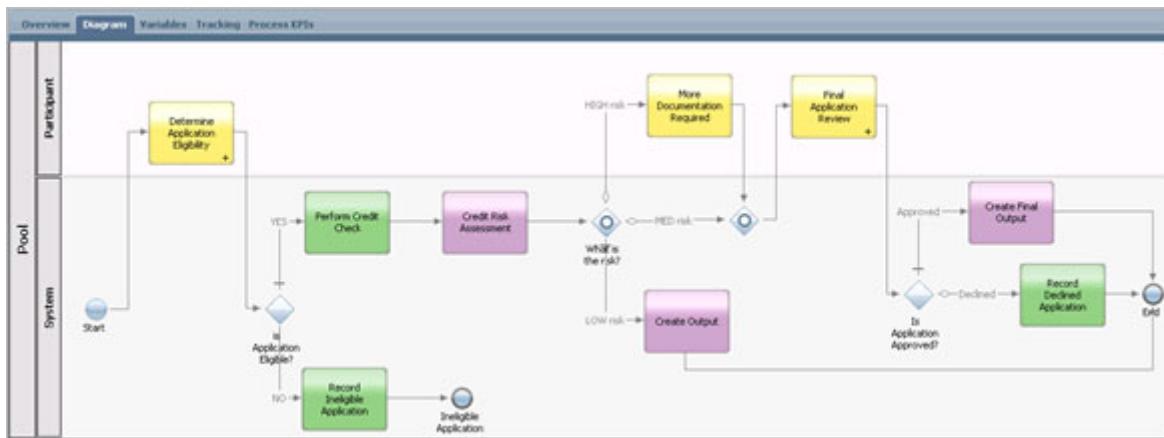
Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Exercise instructions

In this exercise, you use the WS-BPEL graphical editor to define the core business logic for the “account verification” process. You implement the simple and structured activities that choreograph the service invocations that the process requires. For a complete overview of how the business process functions, see the exercise appendixes.

Compare with the model from IBM Process Designer. Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



Part 1: Implement basic and structured BPEL activities in a business process

To implement the AccountVerification business process:

1. Open the Exercise 7 workspace.
 - a. On your desktop, open the **Exercise Shortcuts** folder.
 - b. Double-click the shortcut that is labeled **Exercise 7**. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
 - c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
 - d. Close the **Getting Started** tab.



Note

Because the business process you began developing in Exercise 6 is incomplete, you see errors in the Problems view after you open the workspace. These errors are resolved when you complete this exercise.

- ___ 2. Open the AccountVerification business process.
 - ___ a. In the Business Integration view, expand **FoundationModule > Integration Logic > BPEL Processes > AccountVerification**.
 - ___ b. Double-click **AccountVerification** to open the process editor.
 - ___ 3. Add an invoke activity with the **Display Name**: Determine Application Eligibility
- The **Determine Application Eligibility** activity invokes the **InputCriterion** operation of the **DetermineApplicantEligibility** interface of **DetermineApplicationEligibilityPartner**.
- The **Determine Application Eligibility** activity examines the customer application to determine whether the customer is eligible for an account. Eligibility is predetermined for the test data by using a Java component that you implement later. In practice, this activity would invoke a service such as a human task.
- ___ a. In the palette, expand **Basic Actions** and click **Invoke**.
 - ___ b. Click inside **AccountVerification_Flow** to place the activity on the diagram.
 - ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Determine Application Eligibility

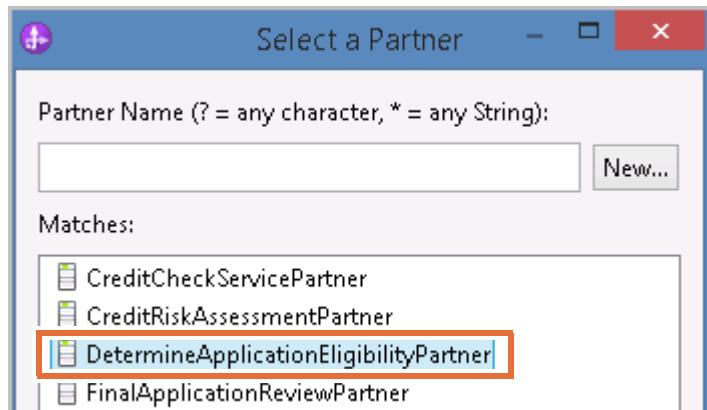
You do not need to change the value in the **Name** field. The **Name** field does not support spaces and is not required to match the value in the **Display Name** field.



Note

All of the activity and link display names are listed in `C:\labfiles\Support Files\Ex7\ActivityNames.txt`. You can use this file to copy and paste the text instead of typing it.

- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. Beside the **Partner** field, click **Browse** and select **DetermineApplicationEligibilityPartner** from the “Select a Partner” dialog box.



- ___ f. Click **OK**.

- __ g. Verify that the **Interface** field is set to `DetermineApplicantEligibility`, the **Operation** field is set to `InputCriterion`, and the **Use data type variables mapping** option is selected.

Description	Partner: DetermineApplicationEligibilityPartner
Details	Interface: DetermineApplicantEligibility
Server	Operation: InputCriterion
Administration	<input checked="" type="checkbox"/> Use data type variables mapping
Exit Condition	

- __ h. In the **Inputs Read from Variable** column, click the **none** link and select `CustomerApplicationVariable` from the list.



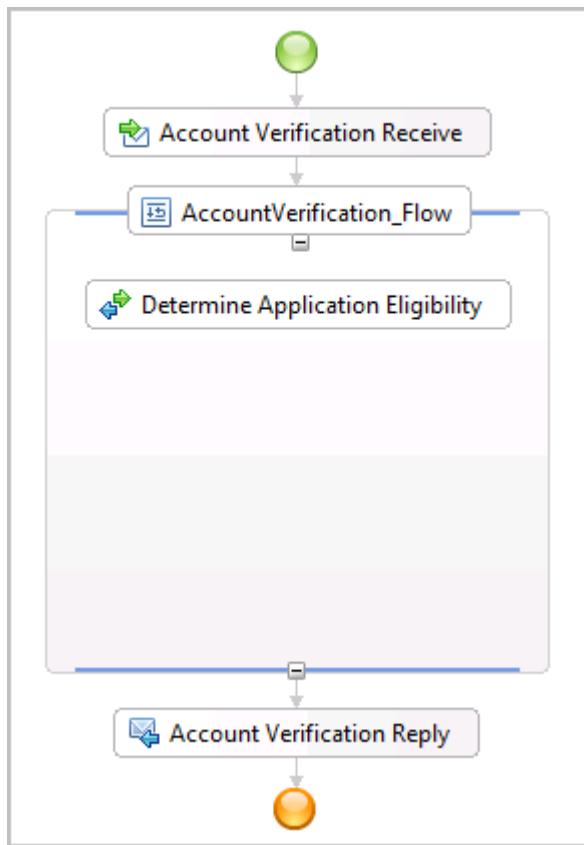
Note

Do *not* select `CustomerApplicationVariable2`.

- __ i. In the **Outputs Store into Variable** column, click the **none** link and select `CustomerApplicationVariable` from the list.

	Name	Type	Read from Variable
Inputs	credappin	CustomerApplication	CustomerApplicationVariable
	Name	Type	Store into Variable
Outputs	credappout	CustomerApplication	CustomerApplicationVariable

- 4. Save your changes. Notice that the errors in the Problems view go away. The process diagram resembles the one shown. (The sticky note was hidden in the following graphic by using the **Hide Notes** menu option.)



Note

You created the first activity **Determine Application Eligibility** from where the process flow starts. See the process model that is provided at the start of the lab. You implement this service invoke in the upcoming labs.

- 5. Add an invoke activity with the **Display Name**: `Map to Ineligible`

The `Map to Ineligible` invoke activity invokes the `InputCriterion` operation of the `MaptoIneligible` interface of `MaptoIneligiblePartner`.

If it is determined that the customer is not eligible for an account, the `Map to Ineligible` activity invokes a service to archive the application. In practice, this activity would likely be used to notify the customer as well.

- In the palette, expand **Basic Actions** and select **Invoke**.
- Click under **Determine Application Eligibility** to add the activity to the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** to: `Map to Ineligible`

- __ d. Switch to the **Details** tab in the **Properties** view.
- __ e. Beside the **Partner** field, click **Browse** and select **MaptoIneligiblePartner** in the “Select a Partner” dialog box.
- __ f. Click **OK**.
- __ g. Verify that the **Interface** field is set to **MaptoIneligible**, the **Operation** is set to **InputCriterion**, and the **Use data type variables mapping** check box is selected.

Description	Partner:*	MaptoIneligiblePartner
Details	Interface:*	MaptoIneligible
Server	Operation:*	InputCriterion
Administration	<input checked="" type="checkbox"/> Use data type variables mapping	
Exit Condition		

- __ h. In the **Inputs Read from Variable** column, click the **none** link and select **CustomerApplicationVariable** from the list.



Note

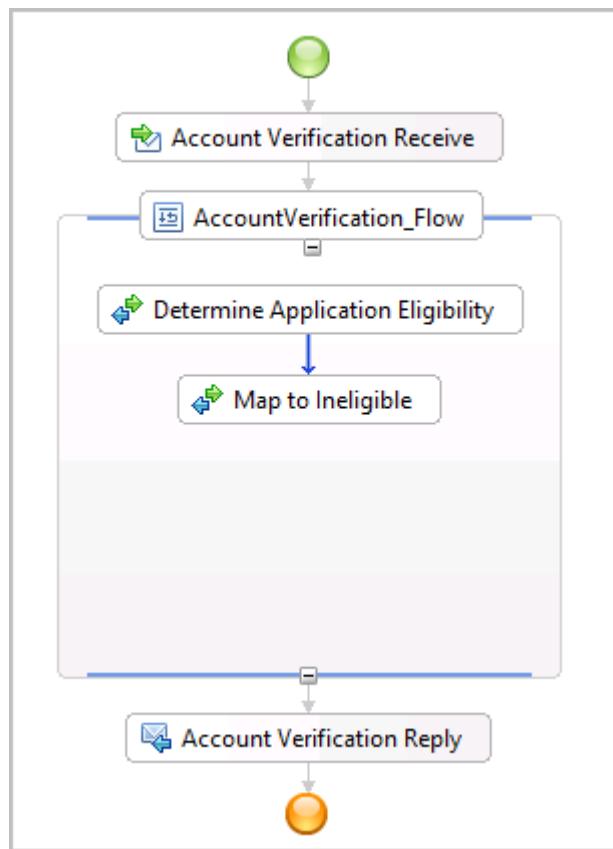
Do **not** select **CustomerApplicationVariable2**.

- __ i. In the **Outputs Store into Variable** column, click the **none** link and select **IneligibleApplicationVariable** from the list.

	Name	Type	Read from Variable	Store into Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable	IneligibleApplicationVariable
	Name	Type		
Outputs	Output	IneligibleApplication		

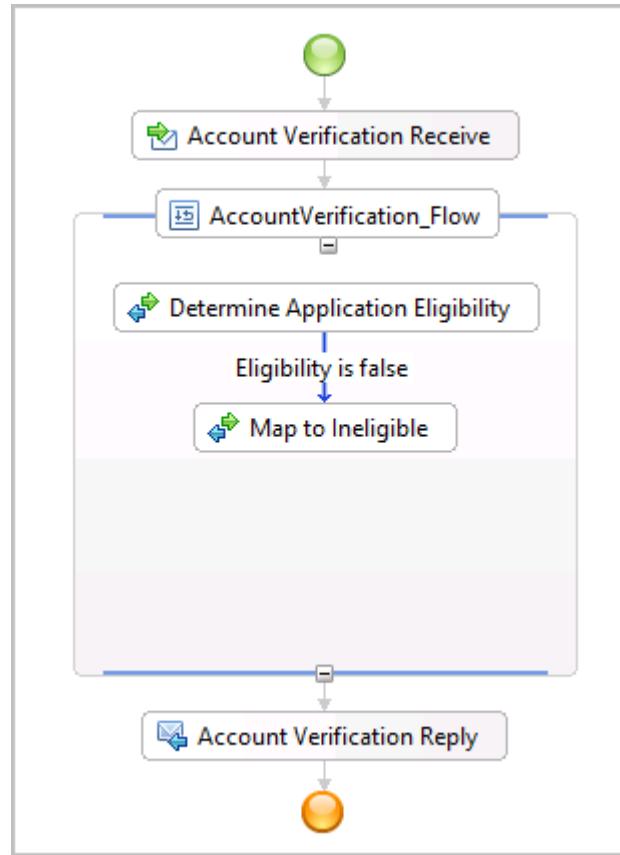
- __ 6. Link **Determine Application Eligibility** to **Map to Ineligible**.
 - __ a. Right-click **Determine Application Eligibility** and click **Add a link** from the menu.

- ___ b. Click **Map to Ineligible** to add the link. Your diagram looks like the following figure:



- ___ 7. Change the **Display Name** of the link between Determine Application Eligibility and Map to Ineligible to: `Eligibility is false`
- ___ a. Select the link between the **Determine Application Eligibility** and **Map to Ineligible**.
 - ___ b. Click the **Description** tab in the **Properties** view.
 - ___ c. Change the **Display Name** of this link to: `Eligibility is false`

- __ d. Right-click the process diagram and click **Show Labels on Links** from the menu. The diagram looks like the following figure:



Troubleshooting

It is possible that the labels do not show on the links after saving your changes. If that occurs, then try turning off the label and then turning it back on again. If they still do not show, then you can continue with the lab because it has no impact.



Note

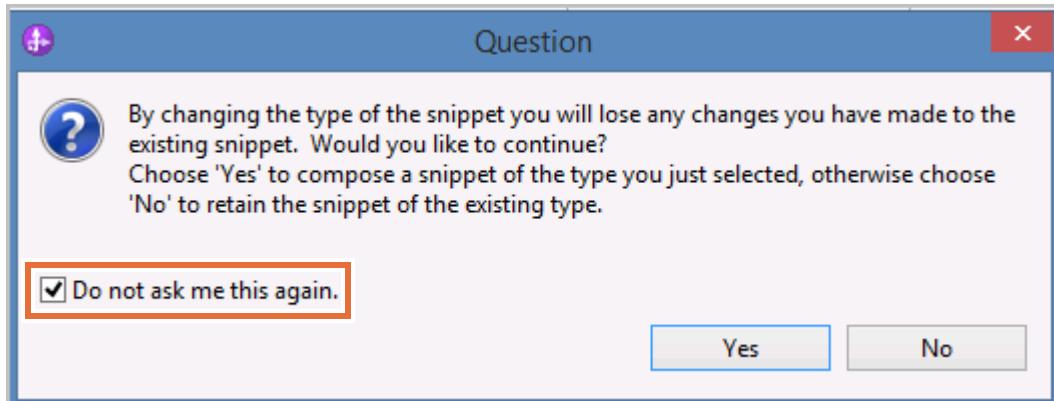
Depending on when you save your process, you might see a red error indicator at the lower left of your process. Do not be concerned, even if the error is there during the entire exercise. However, the error should not be there at the end of this exercise.

- __ 8. Set the link condition so the process flows from Determine Application Eligibility to Map to Ineligible when the eligibleApplication attribute value is `false`.
- __ a. Select the link between the **Determine Application Eligibility** and **Map to Ineligible**.

- ___ b. Switch to the **Details** tab in the **Properties** view.
- ___ c. In the **Expression Language** field, select **Same as Process (Java)**.

The screenshot shows the 'Link - Eligibility is false (Link)' properties window. The 'Expression Language' dropdown at the top is set to 'Same as Process (Java)'. Below it, the 'Details' tab is active, showing the 'Expression Type' section where the 'Visual' radio button is selected. A red box highlights the 'Same as Process (Java)' option in the dropdown.

- ___ d. For **Expression Type**, click **Java**.
- ___ e. In the Question dialog box, select **Do not ask me this again** and click **Yes** to verify that you want to switch to a Java (text) expression.



- ___ f. Open Windows Explorer and browse to `C:\labfiles\Support Files\Ex7`.
- ___ g. Open `AccountVerification_snippets.txt` in a text editor such as Notepad. This file contains all of the Java code that the different parts of your process use.
- ___ h. Copy the `Determine App Eligibility --> Map to Ineligible` code snippet. You do not need to copy the comment lines that begin with `//`.

```
-----  
//-- determine App Eligibility --> Map to Ineligible  
-----  
return !CustomerApplicationvariable.getBoolean("eligibleApplication");
```

- ___ i. Paste the snippet in the expression window over the existing text. Alternatively, you can manually replace the code with the following text:

```
return !CustomerApplicationVariable.getBoolean("eligibleApplication");
```

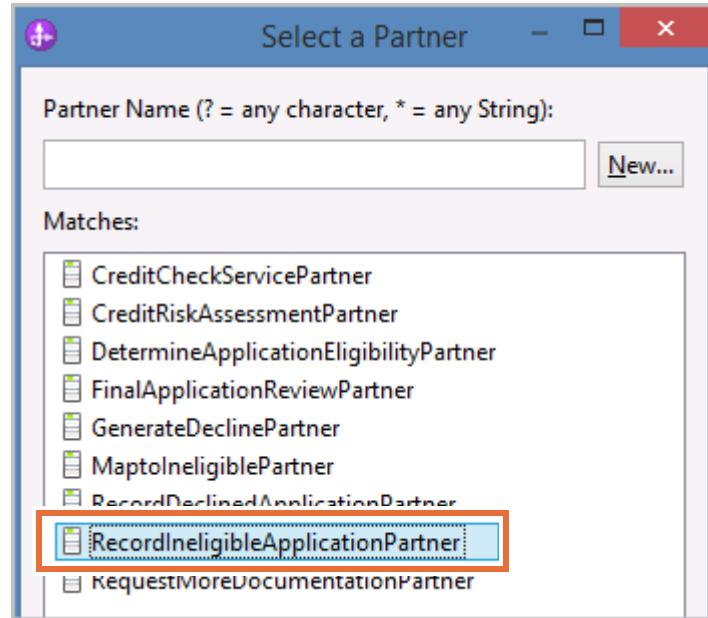
The screenshot shows the 'Link - Eligibility is false (Link)' properties window. The 'Expression Language' dropdown at the top is set to 'Same as Process (Java)'. Below it, the 'Details' tab is active, showing the 'Expression Type' section where the 'Java' radio button is selected. The expression text area contains the Java code: `return !CustomerApplicationVariable.getBoolean("eligibleApplication");`, which is highlighted with a red box.

- ___ 9. Add an invoke activity with the **Display Name**: Record Ineligible Application

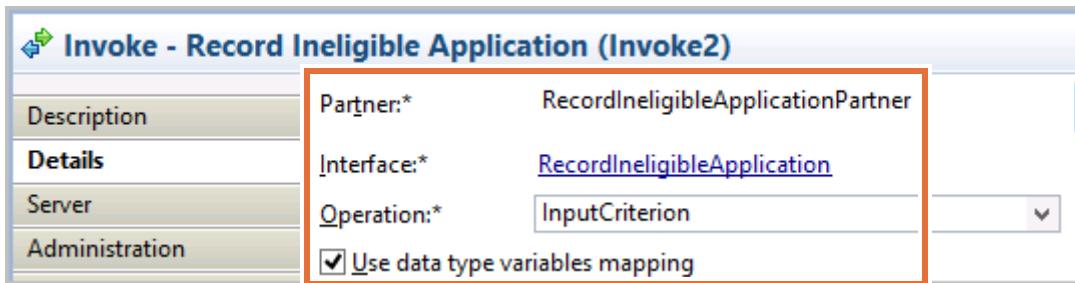
This activity invokes the `InputCriterion` operation of the `RecordIneligible` interface of `RecordIneligibleApplicationPartner`.

This activity archives any customer application that was determined to be ineligible. You implement the service that archives the application in a later exercise.

- __ a. In the palette, expand **Basic Actions**, and select **Invoke**.
- __ b. Click under **Map to Ineligible** to place the activity on the diagram.
- __ c. On the **Description** tab in the **Properties** view, change the **Display Name** to: Record Ineligible Application
- __ d. Switch to the **Details** tab in the **Properties** view.
- __ e. Beside the **Partner** field, click **Browse** and select `RecordIneligibleApplicationPartner` in the “Select a Partner” dialog box.



- __ f. Click **OK**.
- __ g. Verify that the **Interface** field is set to `RecordIneligibleApplication`, the **Operation** field is set to `InputCriterion`, and the **Use data type variables mapping** option is selected.

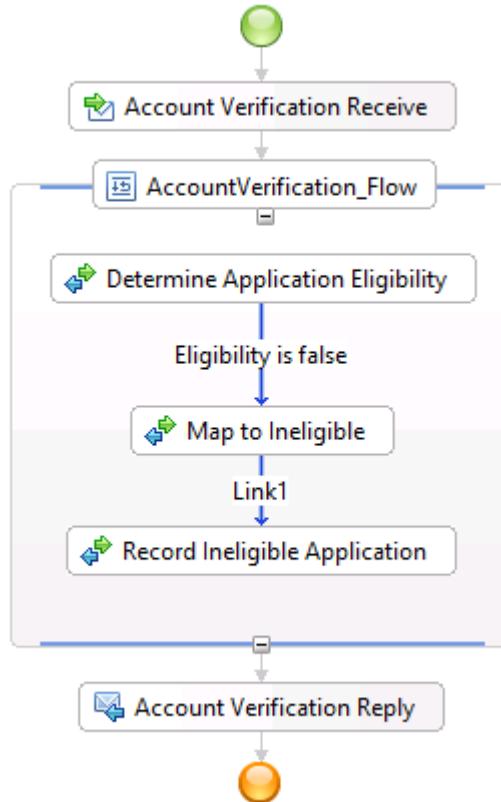


- __ h. In the **Inputs Read from Variable** column, click the **none** link and select `IneligibleApplicationVariable` from the list.

- ___ i. In the **Outputs Store into Variable** column, click the **none** link and select **MessageVariable** from the list.

	Name	Type	
Inputs	Input	IneligibleApplication	Read from Variable IneligibleApplicationVariable ➔
Outputs	Output	Message	Store into Variable ➔ MessageVariable

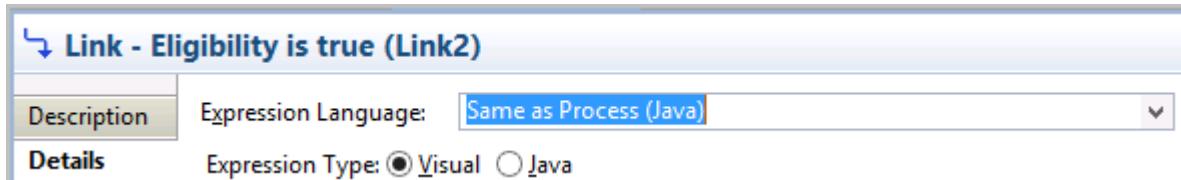
- ___ 10. Link the Map to Ineligible activity to the Record Ineligible Application activity.
- ___ a. Right-click **Map to Ineligible** and click **Add a link** from the menu.
 - ___ b. Click **Record Ineligible Application** to add the link.
 - ___ c. Accept the default link display name.
 - ___ d. To organize the contents for readability, right-click any blank space inside the **AccountVerification_Flow** activity and select **Align Parallel Activities Contents Automatically** (if not already checked).



- ___ 11. Underneath and to the right of **Determine Application Eligibility**, add an empty action with the **Display Name**: **Map to Credit Check**

When implemented, Map to Credit Check transforms the data from a customer application into a format that is suitable for the credit score service. You are using an empty action because this activity is implemented in a later exercise.

- __ a. In the palette, expand **Basic Actions** and select **Empty Action**.
- __ b. Click under and to the right of **Determine Application Eligibility** to place the activity on the diagram.
- __ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: **Map to Credit Check**
- __ 12. Link Determine Application Eligibility to Map to Credit Check and change the Display Name of the link to: **Eligibility is true**
 - __ a. Right-click **Determine Application Eligibility** and click **Add a link** from the menu.
 - __ b. Click **Map to Credit Check** to add the link.
 - __ c. Accept the default link display name.
 - __ d. Select the link between **Determine Application Eligibility** and **Map to Credit Check**.
 - __ e. In the **Properties** view on the **Description** tab, change the **Display Name** to: **Eligibility is true**
 - __ f. To organize the contents for readability, right-click any blank space inside the **AccountVerification_Flow** activity and select **Align Parallel Activities Contents Automatically** (if not already checked).
- __ 13. Set the link condition so that the process flows from Determine Application Eligibility to Map to Credit Check when the eligibleApplication attribute value is `true`.
 - __ a. With the link between **Determine Application Eligibility** and **Map to Credit Check** selected, switch to the **Details** tab in the **Properties** view.
 - __ b. In the **Expression Language** field, select **Same as Process (Java)**.



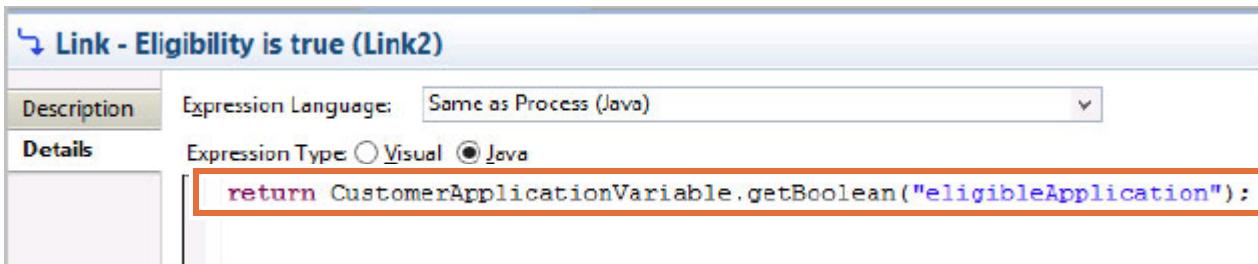
- __ c. For **Expression Type**, click **Java**.
- __ d. Copy the **Determine App Eligibility** --> **Map to Credit Check** code snippet from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt. You do not need to copy the comment lines that begin with `//`.

```
// Determine App Eligibility --> Map to Credit Check
```

```
return CustomerApplicationvariable.getBoolean("eligibleApplication");
```

- ___ e. Paste the snippet in the expression window over the existing text. Alternatively, you can manually replace the code with the following text:

```
return CustomerApplicationVariable.getBoolean("eligibleApplication");
```

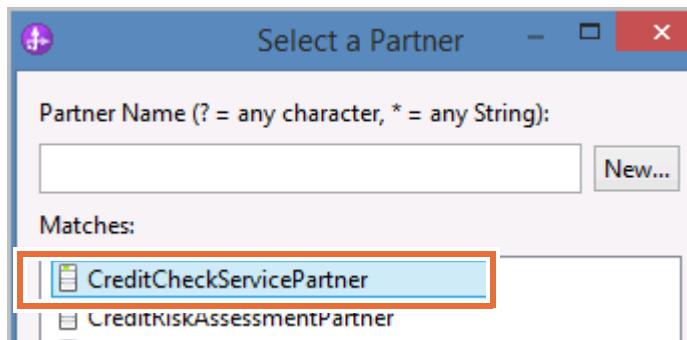


- ___ 14. Be sure to leave the snippets file open.
- ___ 15. Add an invoke activity under Map to Credit Check with the **Display Name**: Credit Check Service

The Credit Check Service activity invokes the calculateCreditScore operation of the CreditScoreService interface of CreditCheckServicePartner.

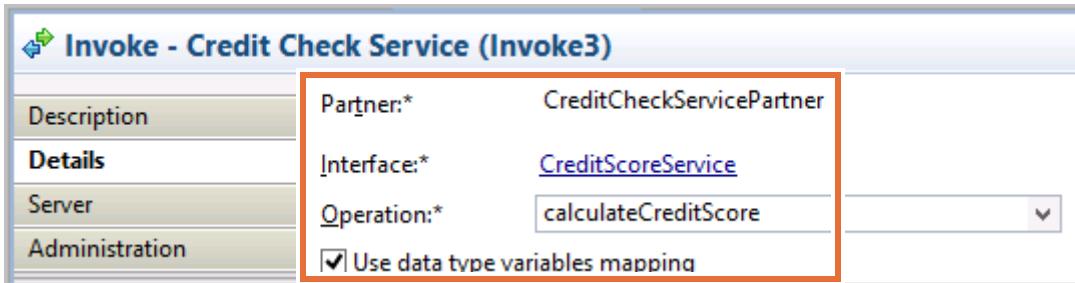
When the application data is transformed into a format that is suitable for the credit score service, the application is sent to the service for credit score calculation. In this application, business rules determine the credit scores. In practice, this service is likely to be an external service that a Business Partner offers.

- ___ a. In the palette, expand **Basic Actions** and select **Invoke**.
- ___ b. Click under **Map to Credit Check** to place the activity on the diagram.
- ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Credit Check Service
- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. Beside the **Partner** field, click **Browse** and select **CreditCheckServicePartner** from the "Select a Partner" dialog box.



- ___ f. Click **OK**.

- ___ g. Verify that **Interface** is set to `CreditScoreService`, **Operation** is set to `calculateCreditScore`, and the **Use data type variables mapping** check box is selected.



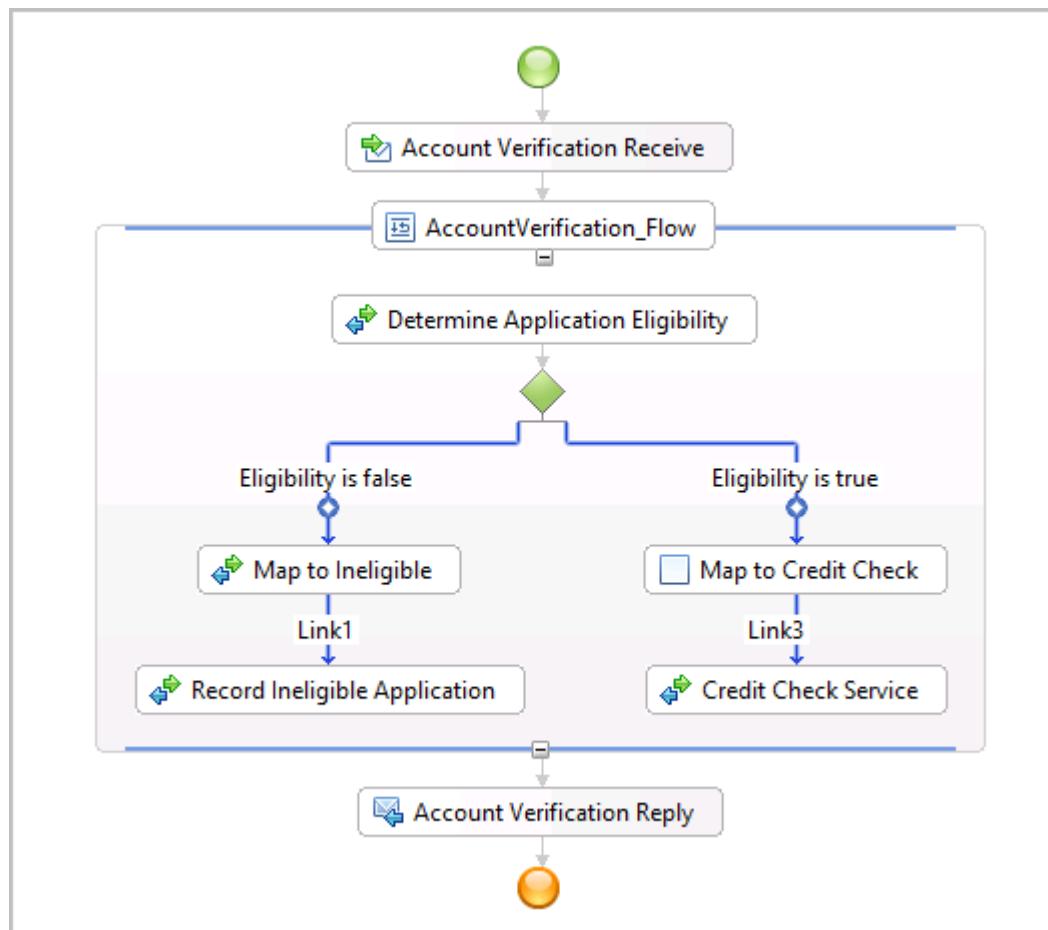
- ___ h. In the **Inputs Read from Variable** column, click the **none** link and select `CreditCheckVariable` from the list.
- ___ i. In the **Outputs Store into Variable** column, click the **none** link and select `CreditCheckVariable` from the list.

	Name	Type	Read from Variable
Inputs	request	CreditCheckRequest	CreditCheckVariable
	Name	Type	Store into Variable
Outputs	calculateCreditScoreReturn	CreditCheckRequest	CreditCheckVariable

___ 16. Link **Map to Credit Check** to **Credit Check Service**.

- ___ a. Right-click **Map to Credit Check** and click **Add a link** from the menu.
- ___ b. Click **Credit Check Service** to add the link.

- ___ c. Accept the default link display name. The diagram looks similar to the following figure:



Note

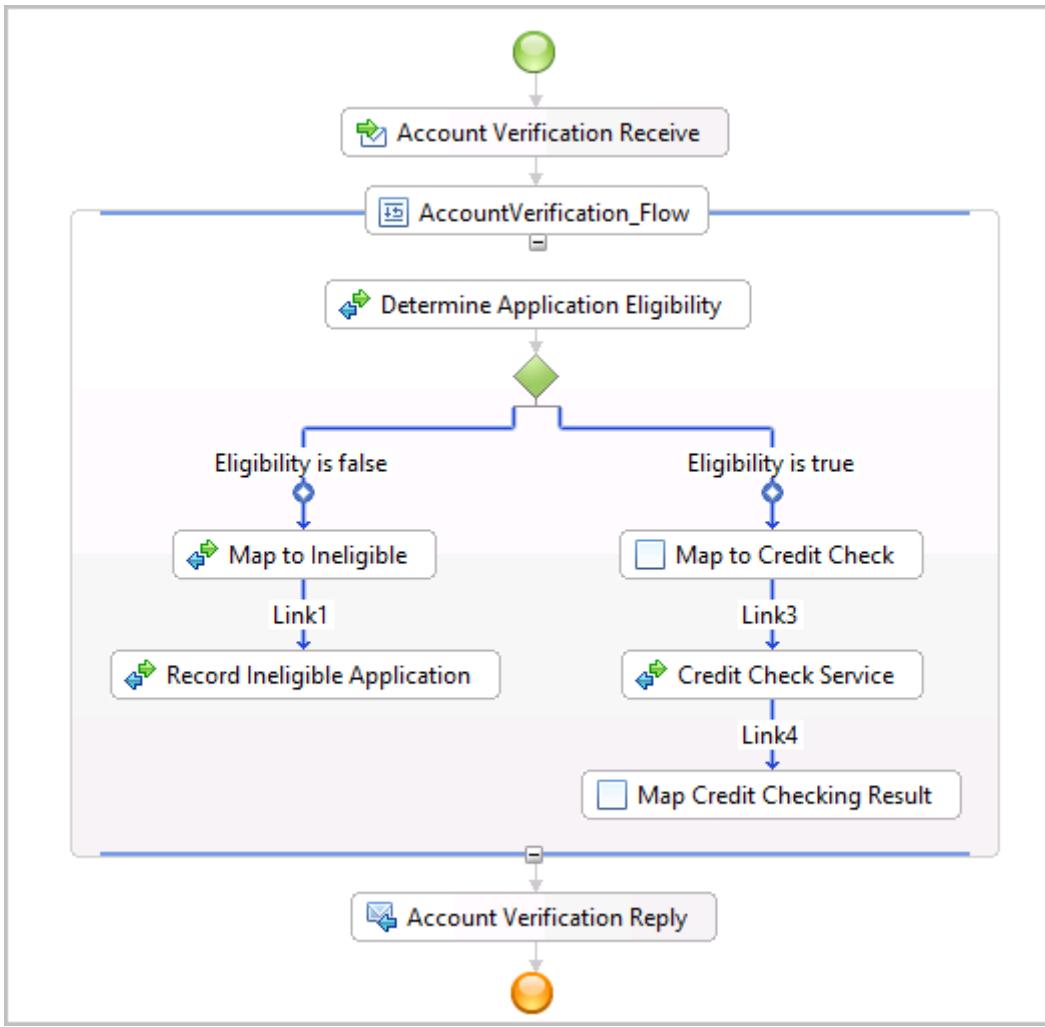
See the model at the beginning of the lab. The **Perform Credit Check** activity in the model corresponds to the **Credit Check Service** build. The invoked service is implemented in the upcoming labs.

- ___ 17. Add an empty action under Credit Check Service with the **Display Name**: Map Credit Checking Result

When the credit score is returned, the data is again transformed. In this case, it is transformed back into a customer application. You are using an empty action because this activity is implemented in a later exercise.

- In the palette, expand **Basic Actions** and select **Empty Action**.
- Click under **Credit Check Service** to place the activity on the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** to: **Map Credit Checking Result**

- ___ 18. Link Credit Check Service to Map Credit Checking Result.
- Right-click **Credit Check Service** and click **Add a link** from the menu.
 - Click **Map Credit Checking Result** to add the link.
 - Accept the default link display name. The diagram looks like the following figure:



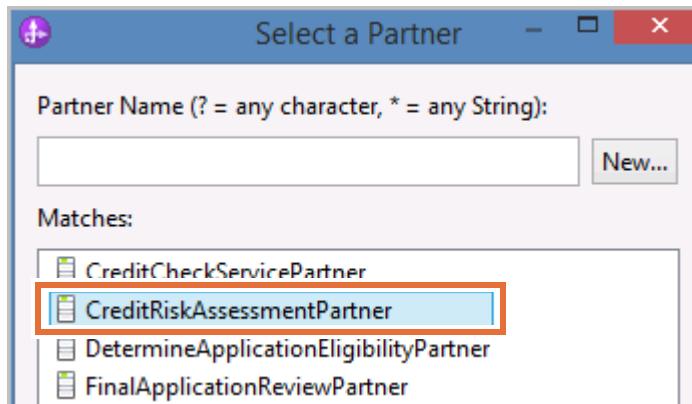
- ___ 19. Add an invoke activity under **Map Credit Checking Result** with Display Name: **Credit Risk Assessment**

Credit Risk Assessment invokes the InputCriterion operation of the CreditRiskAssessment interface of CreditRiskAssessmentPartner. The CreditRiskAssessment activity examines the credit score that is received and takes one of three actions.

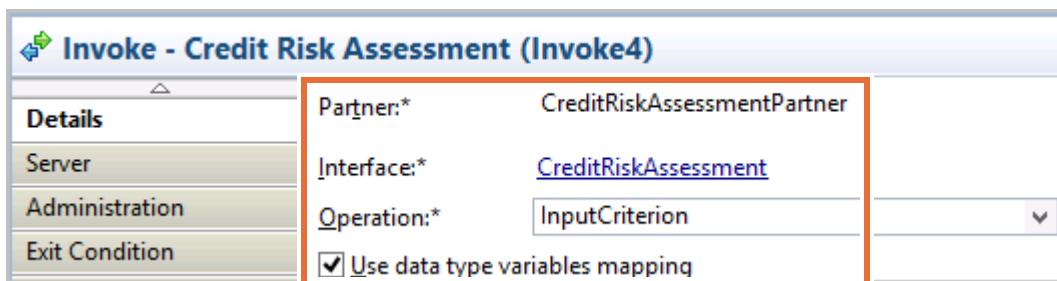
If the credit risk is high, more documentation is requested before final review. If the credit risk is medium, final employee review is requested before approval. If the credit risk is low, the application is approved.

- In the palette, expand **Basic Actions** and select **Invoke**.
- Click under **Map Credit Checking Result** to place the activity on the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** to: **Credit Risk Assessment**

- __ d. Switch to the **Details** tab in the **Properties** view.
- __ e. Beside the **Partner** field, click **Browse** and select **CreditRiskAssessmentPartner** from the “Select a Partner” dialog box.



- __ f. Click **OK**.
- __ g. Verify that the **Interface** field is set to **CreditRiskAssessment**, the **Operation** is set to **InputCriterion**, and the **Use data type variables mapping** check box is selected.



- __ h. In the Inputs Read from Variable column, click the **none** link and select **CustomerApplicationVariable** from the list.

 **Note** _____

Do **not** select **CustomerApplicationVariable2**.

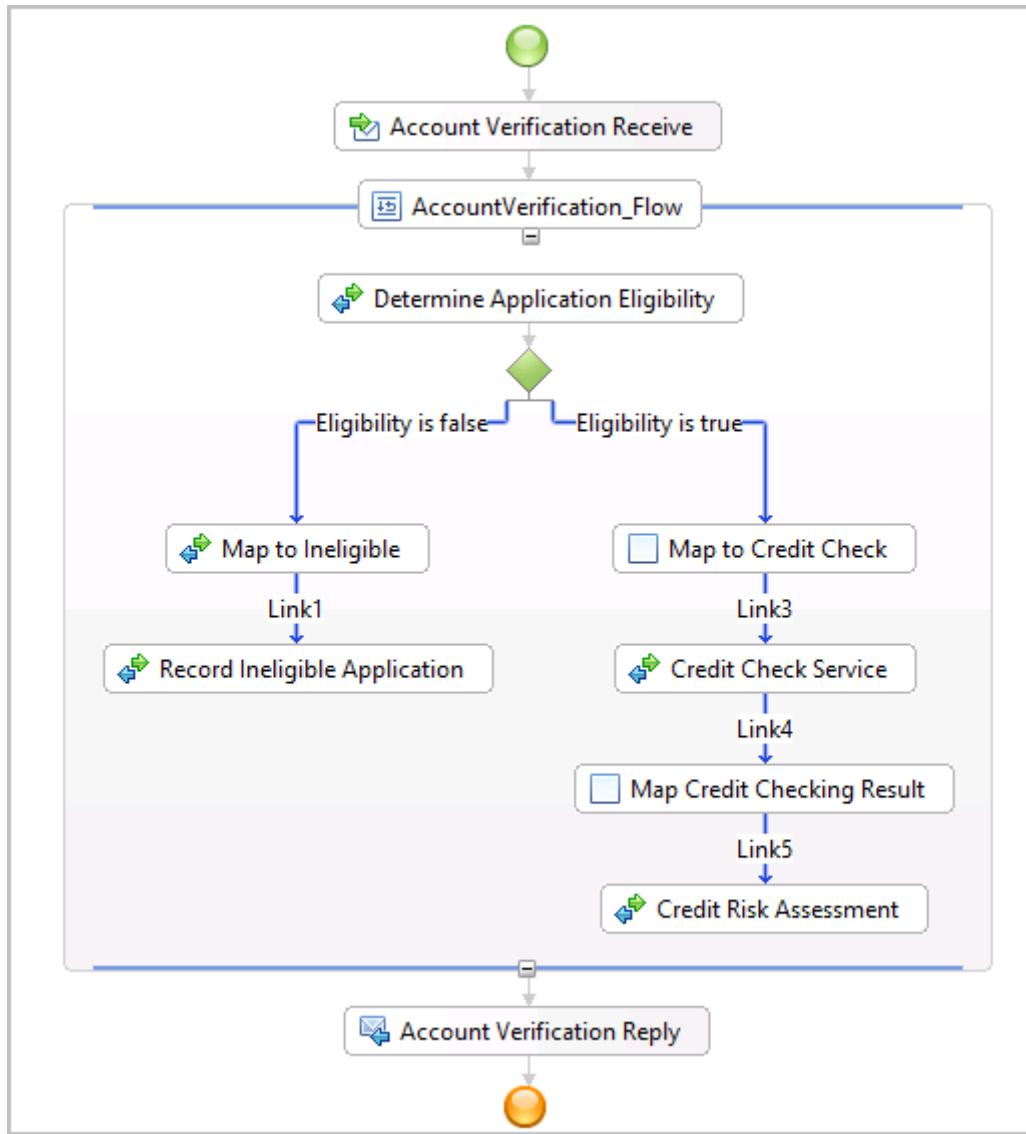
- __ i. In the Outputs Store into Variable column, click the **none** link and select **CustomerApplicationVariable** from the list.

	Name	Type	Read from Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable 
	Name	Type	Store into Variable
Outputs	Output	CustomerApplication	 CustomerApplicationVariable

20. Link Map Credit Checking Result to Credit Risk Assessment.

- __ a. Right-click **Map Credit Checking Result** and click **Add a link** from the menu.

- b. Click **Credit Risk Assessment** to add the link.
- c. Accept the default link display name.
- d. To organize the contents for readability, right-click any blank space inside the **AccountVerification_Flow** activity and click **Align Parallel Activities Contents Automatically** (if not already selected). The diagram looks like the following figure:



- 21. Save your changes.
- 22. Add a snippet activity under **Credit Risk Assessment** with the **Display Name**: Assign Variable
Assign Variable copies CustomerApplicationVariable to CustomerApplicationVariable2. Before writing the customer application business object into the archive, this code copies the business object into a new variable.
 - a. In the palette, expand **Basic Actions** and select **Snippet**.
 - b. Click under **Credit Risk Assessment** to add the snippet to the diagram.

- ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Assign Variable
- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. Click **Java**.
- ___ f. Copy the "Assign Variable" snippet code from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt. The file is still open.

```
// "Assign Variable" snippet
//*****"Assign Variable" snippet*****


if (CustomerApplicationVariable2 == null){
    com.ibm.websphere.bo.BOFactory factory =
    (com.ibm.websphere.bo.BOFactory) new com.ibm.websphere.sca.ServiceManager()
        .locateService("com/ibm/websphere/bo/BOFactory");
    CustomerApplicationVariable2 = factory.create(
        "http://FoundationLibrary/creditserviceitems", "CustomerApplication");
}
CustomerApplicationVariable2=CustomerApplicationVariable;
```



Note

This snippet can be replaced with an Assign activity. The purpose of using it as written is to show you the **BOFactory** and **ServiceManager().locateService()** APIs.

- ___ g. Paste the snippet in the expression window. Alternatively, you can enter the following text:

```
if (CustomerApplicationVariable2 == null) {
    com.ibm.websphere.bo.BOFactory factory =
    (com.ibm.websphere.bo.BOFactory) new
    com.ibm.websphere.sca.ServiceManager().locateService("com/ibm/websphere/b
    o/BOFactory");
    CustomerApplicationVariable2 = factory.create(
        "http://FoundationLibrary/creditserviceitems", "CustomerApplication");
}
CustomerApplicationVariable2=CustomerApplicationVariable;
```

- ___ 23. Save your changes. Continue to ignore any errors in the **Problems** view.
- ___ 24. Link **Credit Risk Assessment** to **Assign Variable** and change the link **Display Name** to: Credit risk is HIGH
 - ___ a. Right-click **Credit Risk Assessment** and click **Add a link** from the menu.
 - ___ b. Click **Assign Variable** to add the link.
 - ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Credit risk is HIGH
- ___ 25. Set the link condition so the process flows from Credit Risk Assessment to Assign Variable when the customer's creditRisk is HIGH.

If the credit risk is HIGH, more steps are taken to gather more documentation before reaching Final Application Review.

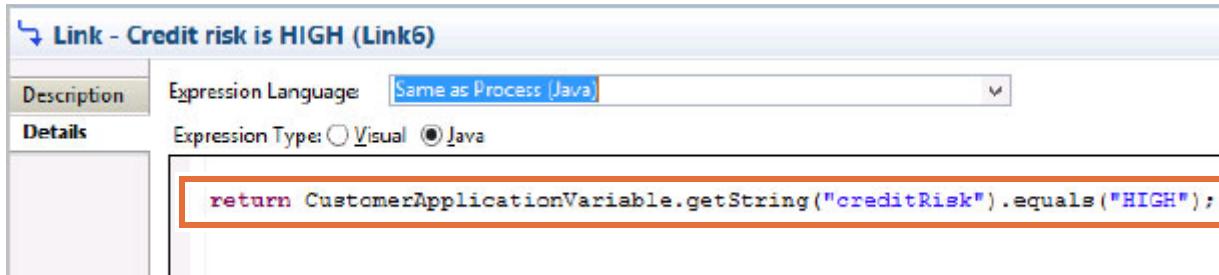
- __ a. Select the link between **Credit Risk Assessment** and **Assign Variable**.
- __ b. Switch to the **Details** tab in the **Properties** view.
- __ c. In the **Expression Language** field, select **Same as Process (Java)**.
- __ d. For **Expression Type**, click **Java**.
- __ e. Copy the Credit Risk Assessment --> Assign Variable code snippet from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.

```
//-----  
// Credit Risk Assessment --> Assign Variable  
//-----
```

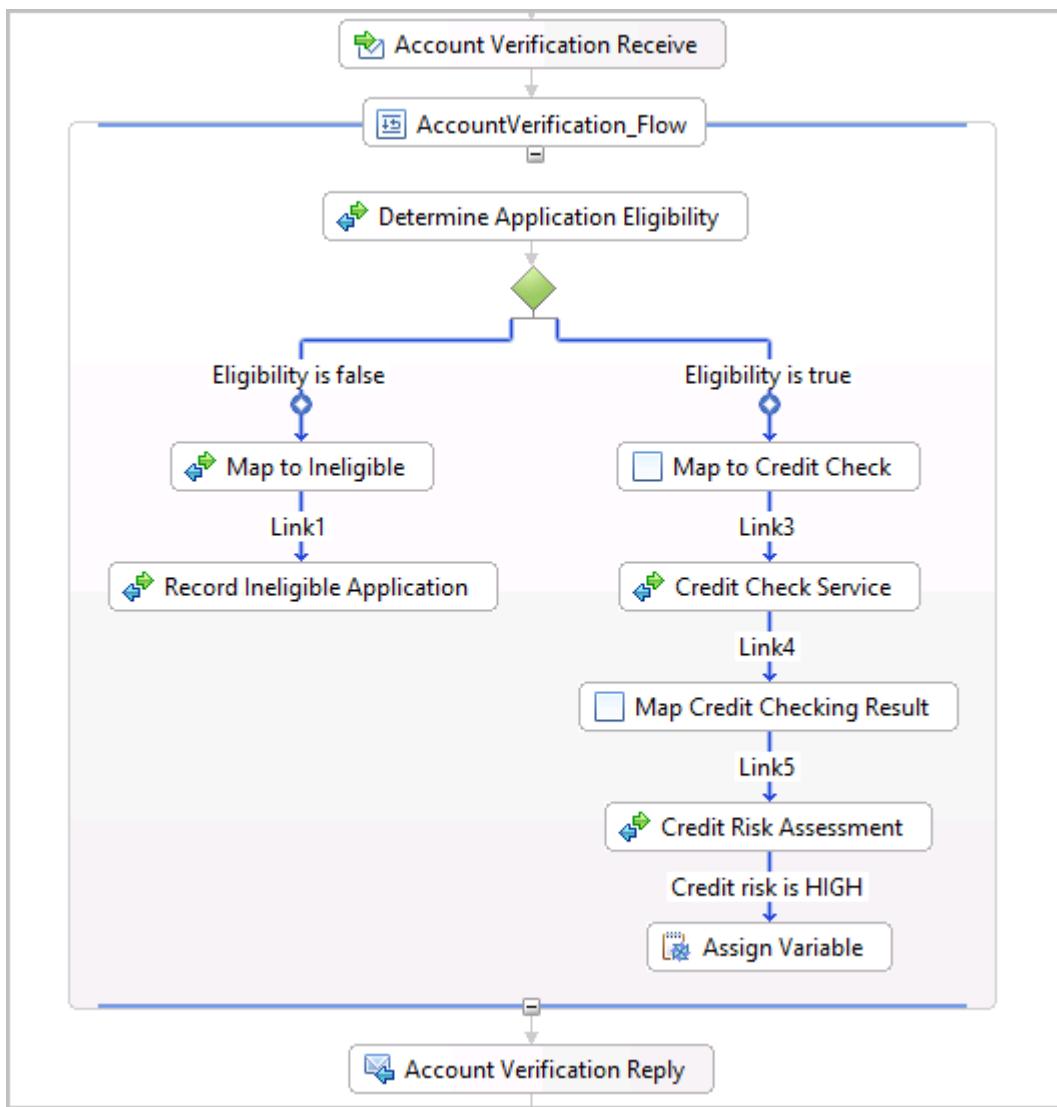
```
return CustomerApplicationVariable.getString("creditRisk").equals("HIGH");
```

- __ f. Paste the snippet in the expression window over the existing text. Alternatively, you can replace the code with the following text:

```
return  
CustomerApplicationVariable.getString("creditRisk").equals("HIGH");
```



- ___ 26. Save your changes. To organize the contents for readability, right-click any blank space inside the **AccountVerification_Flow** activity and click **Align Parallel Activities Contents Automatically**. The diagram looks like the following figure:



- ___ 27. Add a While Loop activity under **Assign Variable** with the **Display Name**: While More Documents Required

The While More Documents Required loop continues to request documentation while the comment attribute is equal to None. When an employee reviews the application, the comment attribute is populated and the loop ends.

- Expand **Structures** in the palette and select **While Loop**.
- Click under **Assign Variable** to add the activity to the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** of the loop to: While More Documents Required
- Switch to the **Details** tab in the **Properties** view.
- In the **Expression Language** field, select **Same as Process (Java)**.

__ f. For **Expression Type**, click **Java**.

__ g. Copy the "While Loop" condition code from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.

```
/*
 * "while Loop" condition
 */
```

```
return CustomerApplicationvariable2.getString("comments").equals("None");
```

__ h. Paste the snippet in the expression window over the existing text. Alternatively, you can replace the code with the following text:

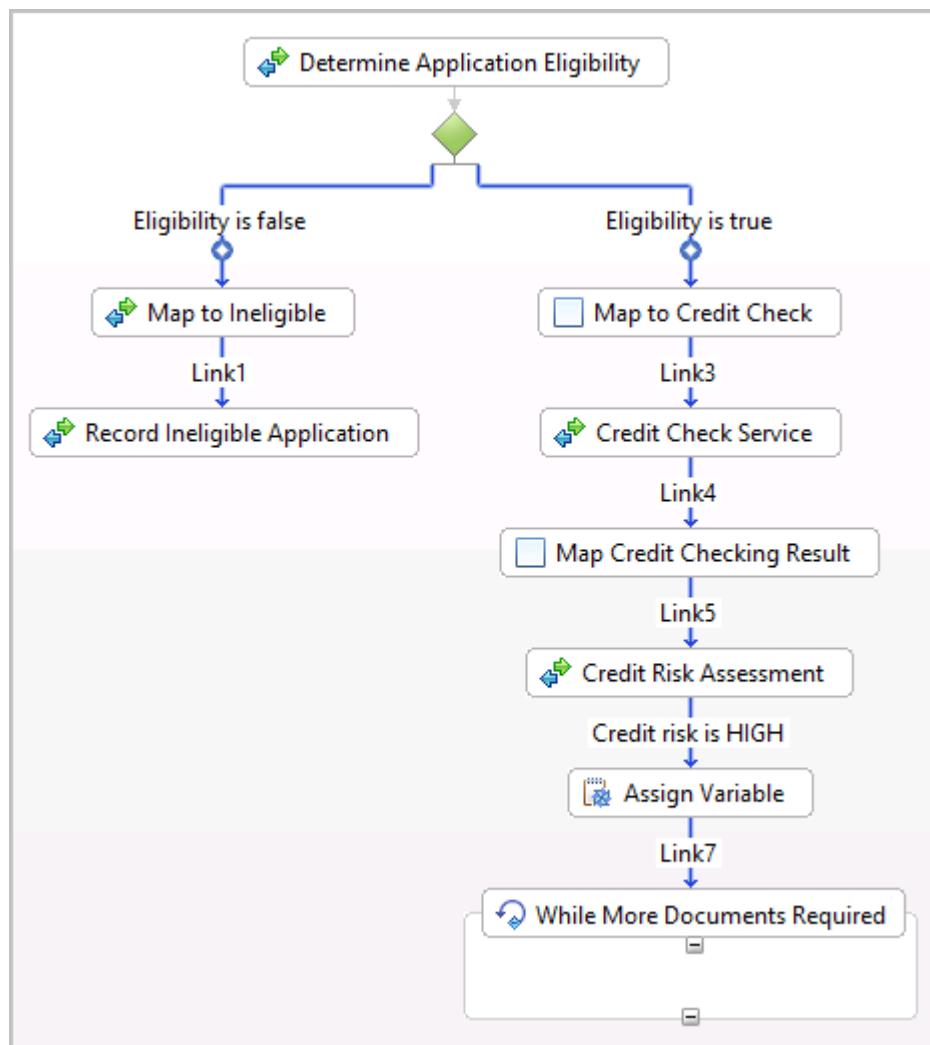
```
return CustomerApplicationVariable2.getString("comments").equals("None");
```

28. Link **Assign Variable** to **While More Documents Required**.

__ a. Right-click **Assign Variable** and click **Add a link** from the menu.

__ b. Click **While More Documents Required** to add the link.

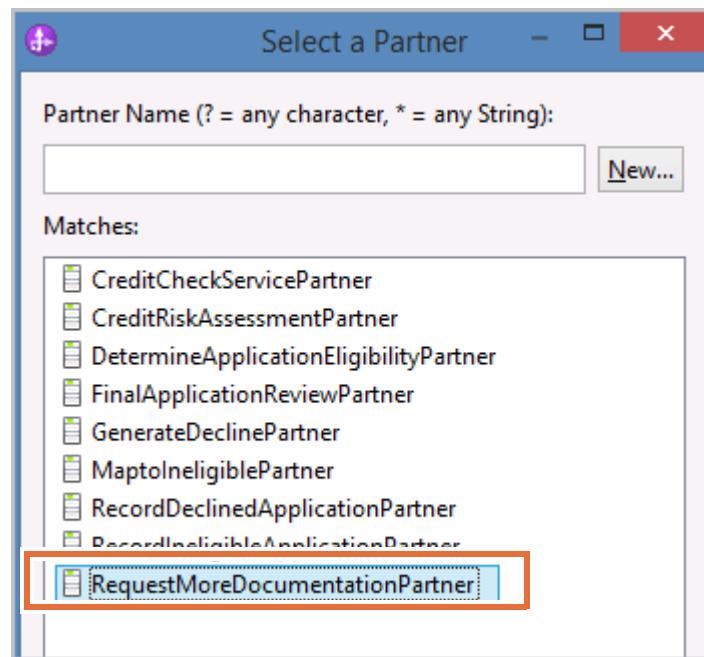
__ c. Accept the default link display name. The current path in your diagram looks like the following figure:



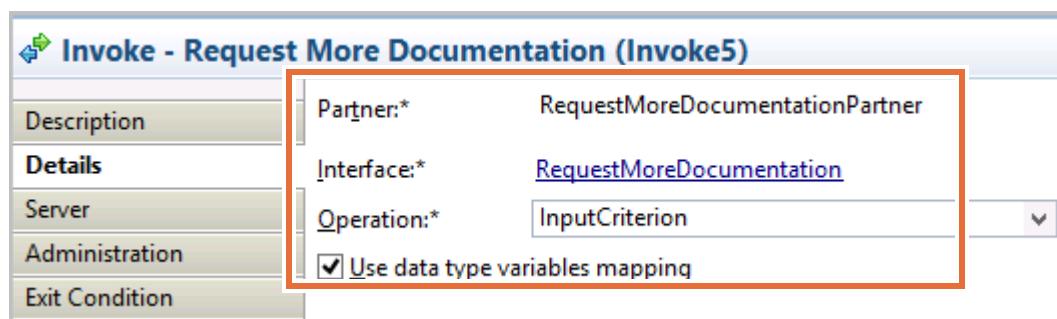
- ___ 29. Add an invoke activity inside the while loop with the **Display Name**: Request More Documentation

The Request More Documentation activity invokes the InputCriterion operation of the RequestMoreDocumentation interface of RequestMoreDocumentationPartner. You implement the human task that is associated with this activity in a later exercise.

- In the palette, expand **Basic Actions** and select **Invoke**.
- Click inside the while loop to add the activity to the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** to: Request More Documentation
- Switch to the **Details** tab in the **Properties** view.
- Beside the **Partner** field, click **Browse** and select **RequestMoreDocumentationPartner** from the “Select a Partner” dialog box.



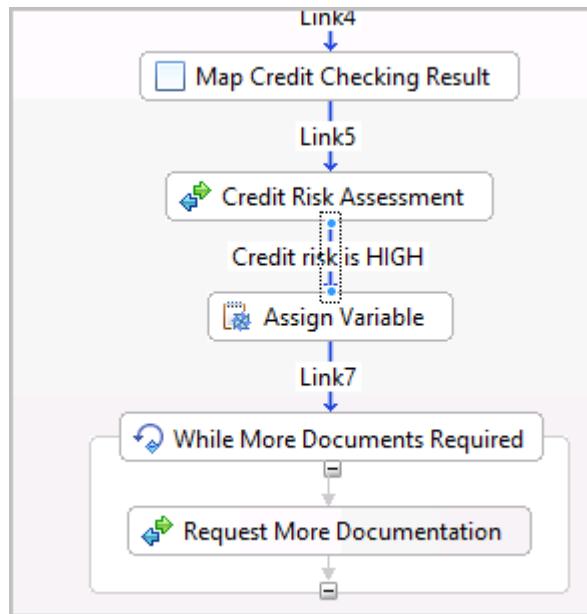
- Click **OK**.
- Verify that the **Interface** field is set to **RequestMoreDocumentation**, the **Operation** field is set to **InputCriterion**, and the **Use data type variables mapping** option is selected.



- __ h. In the **Inputs Read from Variable** column, click the **none** link and select **CustomerApplicationVariable2** from the list.
- __ i. In the **Outputs Store into Variable** column, click the **none** link and select **CustomerApplicationVariable2** from the list.

	Name	Type	Read from Variable	Store into Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable2	CustomerApplicationVariable2
	Name	Type		
Outputs	Output	CustomerApplication		CustomerApplicationVariable2

- __ 30. Save your changes. The current path in your diagram looks like the following figure:



Note

See the model at the beginning of the lab. The **More Documentation Required** activity in the model corresponds to the preceding **Request More Documentation** invoke. The invoked service is implemented in the upcoming labs.

- __ 31. Add a snippet activity after the while loop with the **Display Name**: **Merge Assign**
The Merge Assign activity merges the data from the Request More Documentation service. This activity can also be implemented as an assign.
 - __ a. In the palette, expand **Basic Actions** and select **Snippet**.
 - __ b. Click beneath the **While More Documents Required** loop to add the snippet to the diagram.
 - __ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: **Merge Assign**

- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. Click **Java**.
- ___ f. Copy the "Merge Assign" snippet code from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.

```
/*
*****"Merge Assign" snippet*****
if (CustomerApplicationVariable == null){
    com.ibm.websphere.bo.BOFactory factory =
        (com.ibm.websphere.bo.BOFactory) new
    com.ibm.websphere.sca.ServiceManager().locateService("com/ibm/websphere/bo/BOFactory");
    CustomerApplicationVariable = factory.create(
    "http://FoundationLibrary/creditserviceitems", "CustomerApplication");
}
CustomerApplicationVariable=CustomerApplicationVariable2;
```

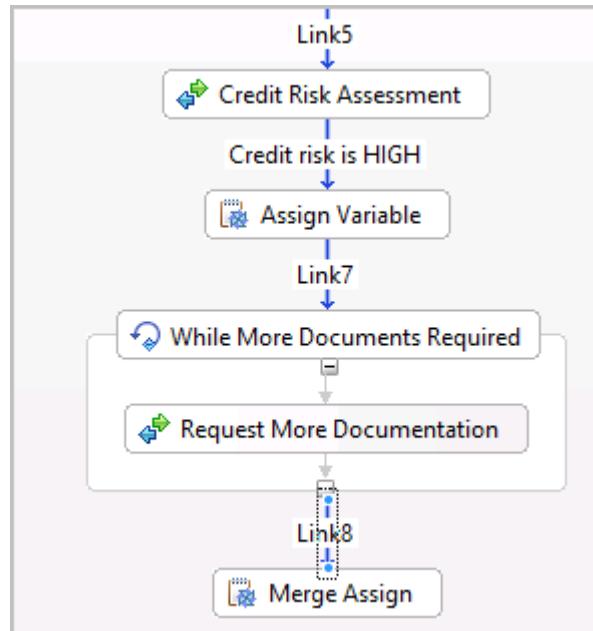
- ___ g. Paste the snippet in the expression window. Alternatively, you can replace the code with the following text:

```
if (CustomerApplicationVariable == null) {
com.ibm.websphere.bo.BOFactory factory =
    (com.ibm.websphere.bo.BOFactory) new
com.ibm.websphere.sca.ServiceManager().locateService("com/ibm/websphere/b
o/BOFactory");
CustomerApplicationVariable = factory.create(
    "http://FoundationLibrary/creditserviceitems", "CustomerApplication");
}
CustomerApplicationVariable=CustomerApplicationVariable2;
```

___ 32. Link While More Documents Required to Merge Assign.

- ___ a. Right-click **While More Documents Required** and click **Add a link** from the menu.
- ___ b. Click **Merge Assign** to add the link.
- ___ c. Accept the default link display name.

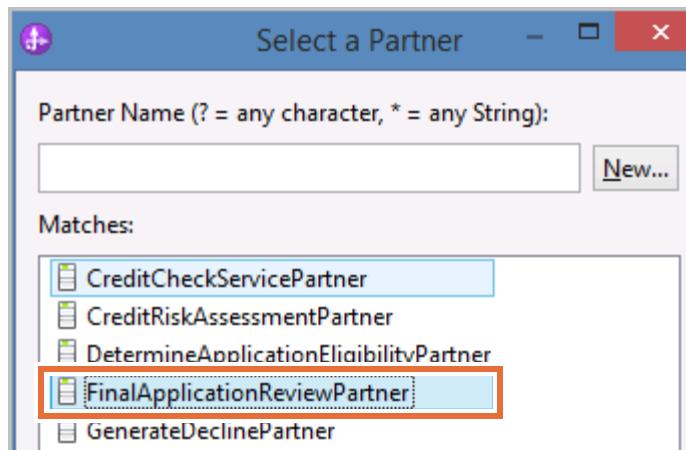
- ___ 33. Save your changes.



- ___ 34. Add an invoke activity under Merge Assign with the **Display Name**: Final Application Review

The Final Application Review activity invokes the InputCriterion operation of the FinalApplicationReview interface of FinalApplicationReviewPartner. This activity invokes a human task that allows an employee to review an application. You implement the human task in a later exercise.

- In the palette, expand **Basic Actions** and select **Invoke**.
- Click under **Merge Assign** to add the activity to the diagram.
- In the **Properties** view on the **Description** tab, change the **Display Name** to: **Final Application Review**
- Switch to the **Details** tab in the **Properties** view.
- Beside the **Partner** field, click **Browse** and select **FinalApplicationReviewPartner** from the “Select a Partner” dialog box.



- ___ f. Click **OK**.
- ___ g. Verify that the **Interface** field is set to `FinalApplicationReview`, the **Operation** field is set to `InputCriterion`, and the **Use data type variables mapping** check box is selected.

Invoke - Final Application Review (Invoke6)

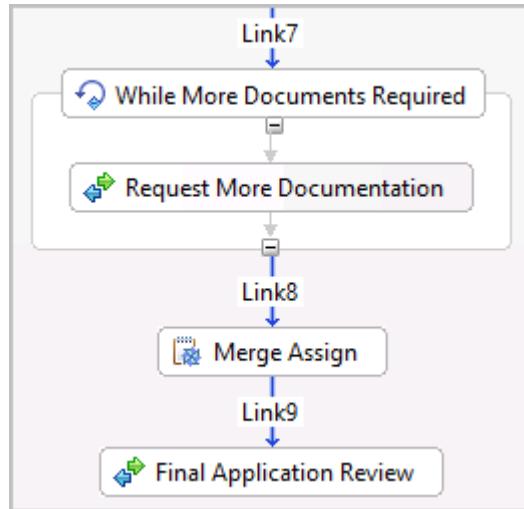
Description	Partner:*	FinalApplicationReviewPartner
Details	Interface:*	FinalApplicationReview
Server	Operation:*	InputCriterion
Administration	<input checked="" type="checkbox"/> Use data type variables mapping	
Exit Condition		

- ___ h. In the **Inputs Read from Variable** column, click the **none** link and select `CustomerApplicationVariable` from the list.
- ___ i. In the **Outputs Store into Variable** column, click the **none** link and select `CustomerApplicationVariable` from the list.

	Name	Type	Read from Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable
	Name	Type	Store into Variable
Outputs	Output	CustomerApplication	CustomerApplicationVariable

- ___ 35. Save your changes. Continue to ignore any errors in the **Problems** view.
- ___ 36. Link **Merge Assign** to **Final Application Review**.
 - ___ a. Right-click **Merge Assign** and click **Add a link** from the menu.
 - ___ b. Click **Final Application Review** to add the link.

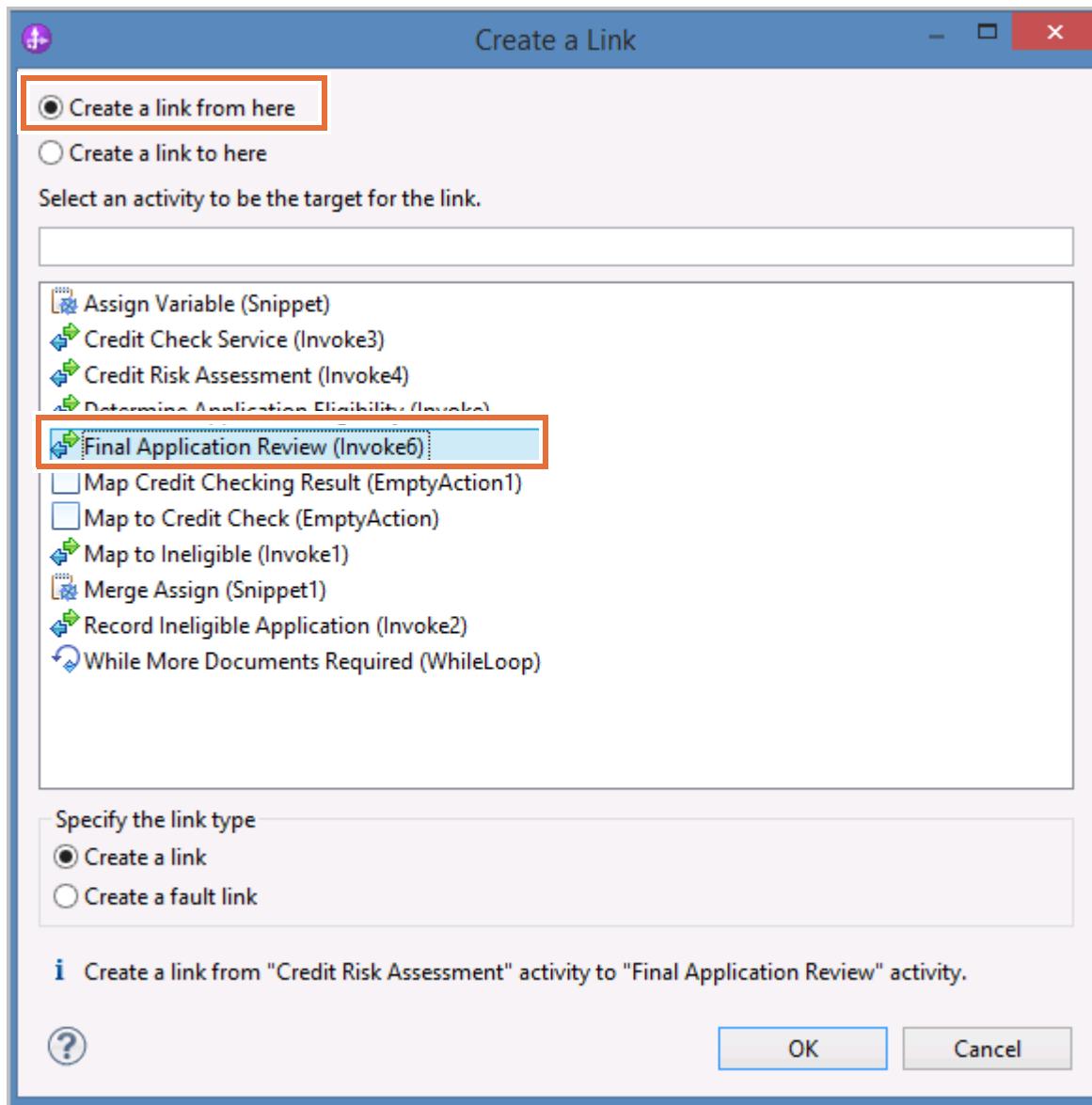
- ___ c. Accept the default link display name and save your changes. The current path in your process looks like the following diagram:



___ 37. Link **Credit Risk Assessment** to **Final Application Review**.

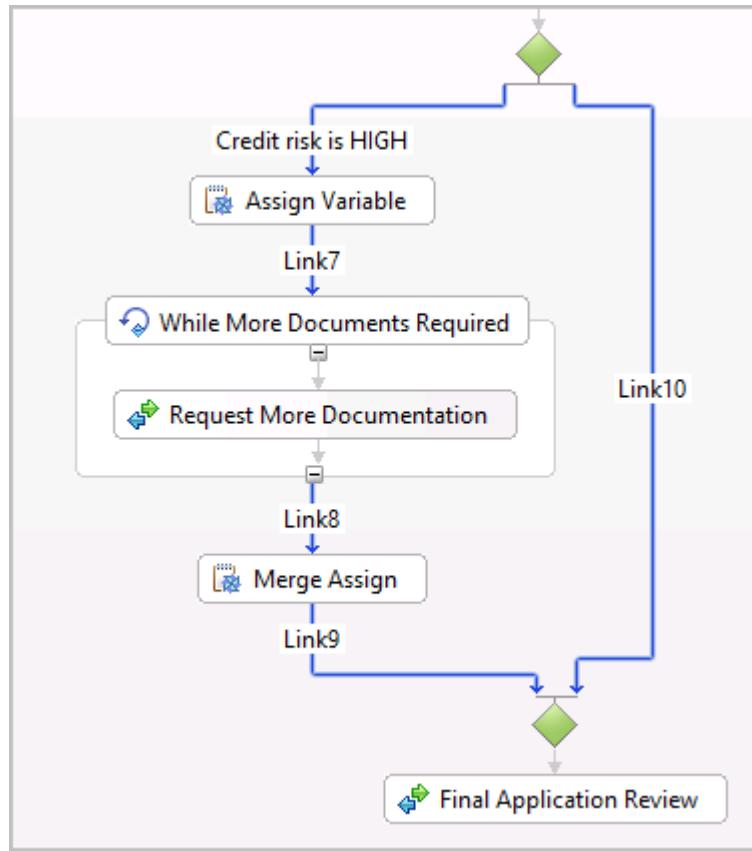
- ___ a. Right-click **Credit Risk Assessment** and click **Create a Link From/To Here** from the menu. This option can be used to link to activities that are outside your viewing area.

- ___ b. In the “Create a Link” dialog box, take the following actions:
- Verify that **Create a link from here** is selected.
 - Select **Final Application Review** from the list.
 - Verify that **Create a link** is selected in the **Specify the link type** section.



- ___ c. Click **OK**.
- ___ d. Accept the default link display name.

- __ e. The current path in your process looks like the following diagram:



- __ 38. Set the link condition between Credit Risk Assessment and Final Application Review so that the process flows directly to Final Application Review when creditRisk is MED (short for medium). Change the link **Display Name** to: Credit risk is MED

If the credit risk is MED, the application is sent directly to Final Application Review. No additional documentation is required, but the credit risk is not low enough to warrant preapproval of the application without review.

- __ a. Click the link between **Credit Risk Assessment** and **Final Application Review**.
- __ b. In the **Properties** view on the **Description** tab, change the **Display Name** to: Credit risk is MED
- __ c. Switch to the **Details** tab in the **Properties** view.
- __ d. In the **Expression Language** field, select **Same as Process (Java)**.
- __ e. For **Expression Type**, click **Java**.
- __ f. Copy the Credit Risk Assessment --> Final Application Review code snippet from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.

```

// Credit Risk Assessment --> Final Application Review
//-
return CustomerApplicationvariable.getstring("creditRisk").equals("MED");
  
```

- ___ g. Paste the snippet in the expression window over the existing text. Alternatively, you can manually replace the code with the following text:

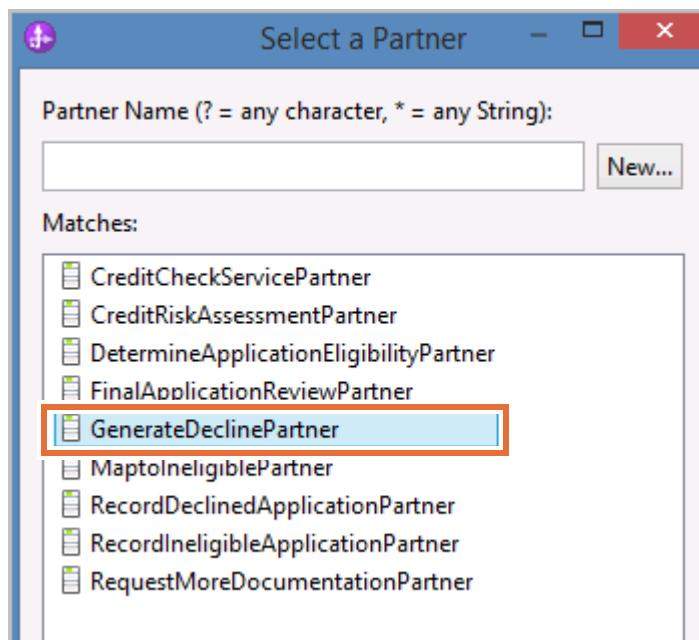
```
return CustomerApplicationVariable.getString("creditRisk").equals("MED");
```

- ___ 39. Save your changes.

- ___ 40. Add an invoke activity under **Final Application Review** with the **Display Name**: Generate Decline

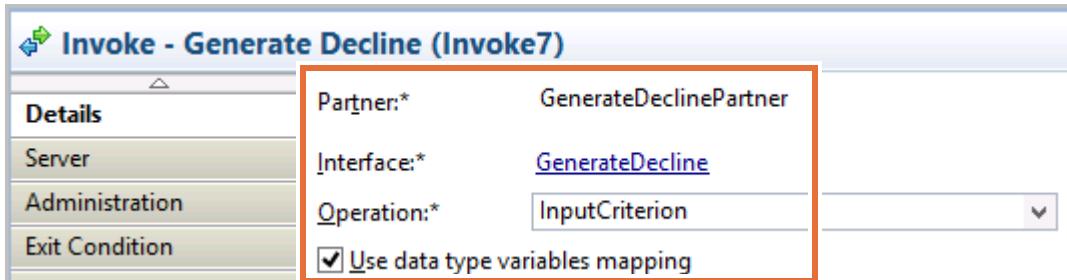
The Generate Decline activity invokes the InputCriterion operation of the GenerateDecline interface of GenerateDeclinePartner.

- ___ a. In the palette, expand **Basic Actions** and select **Invoke**.
- ___ b. Click under **Final Application Review** to add the activity to the diagram.
- ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Generate Decline
- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. Beside the **Partner** field, click **Browse** and select **GenerateDeclinePartner** from the "Select a Partner" dialog box.



- ___ f. Click **OK**.

- __ g. Verify that the **Interface** field is set to `GenerateDecline`, the **Operation** is set to `InputCriterion`, and the **Use data type variables mapping** option is selected.

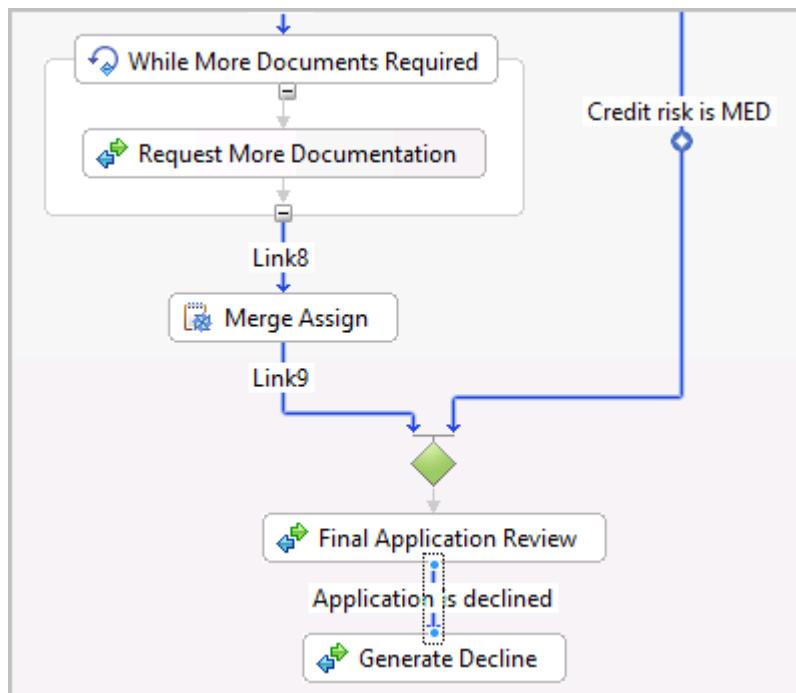


- __ h. In the **Inputs Read from Variable** column, click the **none** link and select **CustomerApplicationVariable** from the list.
- __ i. In the **Outputs Store into Variable** column, click the **none** link and select **MessageVariable** from the list.

	Name	Type	Read from Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable
	Name	Type	Store into Variable
Outputs	Output	Message	MessageVariable

- __ 41. Link **Final Application Review** to **Generate Decline** and change the **Display Name** to: Application is declined
- __ a. Right-click **Final Application Review** and click **Add a link** from the menu.
- __ b. Click **Generate Decline** to add the link.
- __ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Application is declined

The current path in your diagram looks like the following figure:



42. Set the link condition between Final Application Review and Generate Decline so the process flows to Generate Decline when applicationDecision is false.

If the account reviewer denies the customer application, the application decision attribute is set to `false`. The Generate Decline and Record Declined Application archive the declined application. These services are implemented in a later exercise.

- __ a. Click the link between **Final Application Review** and **Generate Decline**.
 - __ b. Switch to the **Details** tab in the **Properties** view.
 - __ c. In the **Expression Language** field, select **Same as Process (Java)**.
 - __ d. For **Expression Type**, click **Java**.
 - __ e. Copy the Final Application Review --> Generate Decline code snippet from C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.

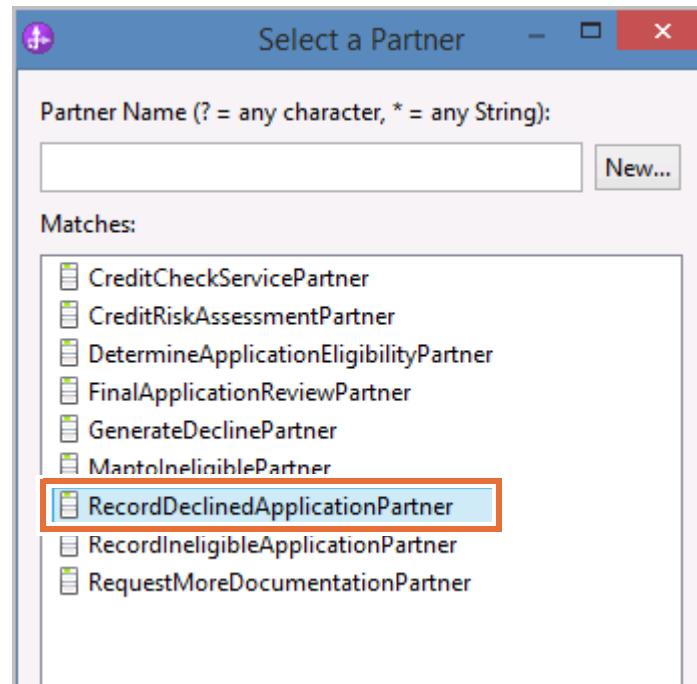
```
//-----  
// Final Application Review --> Generate Decline  
//-----  
  
return !CustomerApplicationvariable.getBoolean("applicationDecision");
```
 - __ f. Paste the snippet in the expression window over the existing text. Alternatively, you can replace the code with the following text:

```
return !CustomerApplicationVariable.getBoolean ("applicationDecision");
```

43. Add an invoke activity under **Generate Decline** with the **Display Name**: Record Declined Application

The Record Declined Application activity invokes the InputCriterion operation of the RecordDeclinedApplication interface of RecordDeclinedApplicationPartner.

- a. In the palette, expand **Basic Actions** and select **Invoke**.
- b. Click under **Generate Decline** to add the invoke activity to the diagram.
- c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Record Declined Application
- d. Switch to the **Details** tab in the **Properties** view.
- e. Beside the **Partner** field, click **Browse** and select **RecordDeclinedApplicationPartner** from the “Select a Partner” dialog box.



- f. Click **OK**.
- g. Verify that the **Interface** field is set to `RecordDeclinedApplication`, the **Operation** is set to `InputCriterion`, and the **Use data type variables mapping** check box is selected.

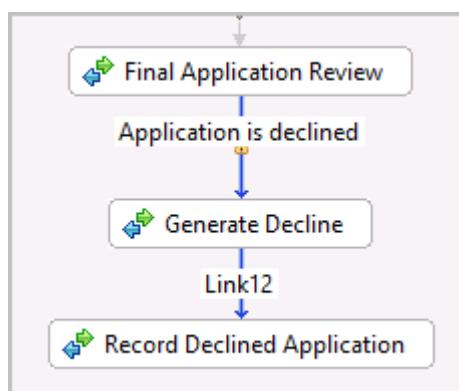
Invoke - Record Declined Application (Invoke8)	
Description	Partner: [*] <code>RecordDeclinedApplicationPartner</code>
Details	Interface: [*] <code>RecordDeclinedApplication</code>
Server	Operation: [*] <code>InputCriterion</code>
Administration	<input checked="" type="checkbox"/> Use data type variables mapping
Exit Condition	

- ___ h. In the **Inputs Read from Variable** column, click the **none** link and select **CustomerApplicationVariable** from the list.

	Name	Type	Read from Variable
Inputs	Input	CustomerApplication	CustomerApplicationVariable

- ___ 44. Link **Generate Decline** to **Record Declined Application**.

- ___ a. Right-click **Generate Decline** and click **Add a link** from the menu.
- ___ b. Click **Record Declined Application** to add the link.
- ___ c. Accept the default link display name. The current path in your diagram looks like the following figure:

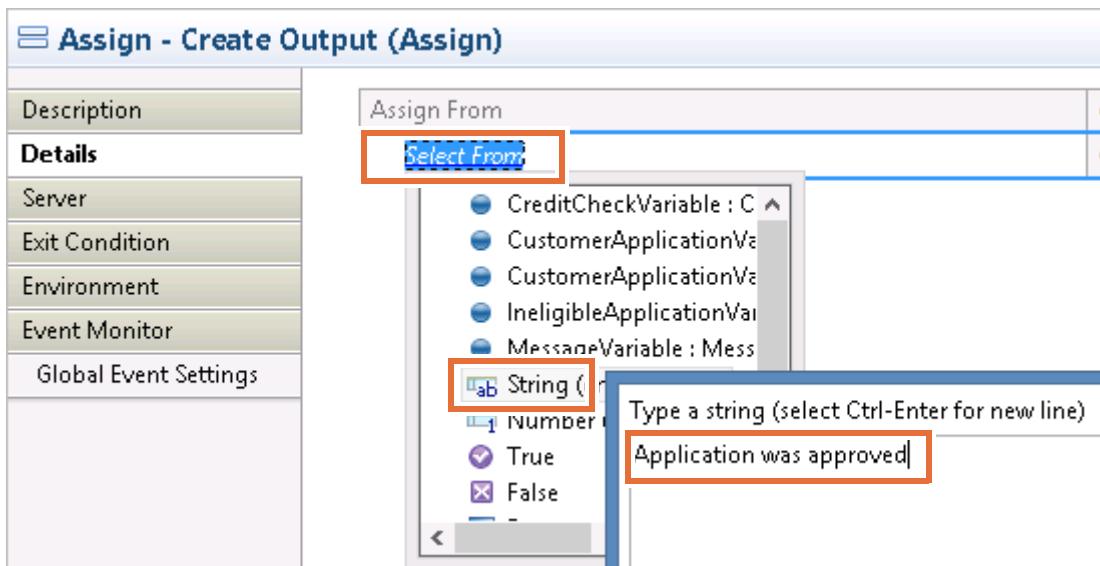


- ___ 45. Add an assign activity to the right and under **Final Application Review** with the **Display Name**: Create Output

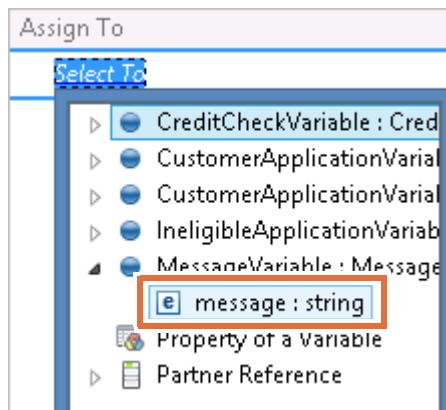
Create Output sets the process output message to: Application was approved. The process returns this message when creditRisk is HIGH or MED, and the person who reviews the application approves it.

- ___ a. In the palette, expand **Basic Actions** and select **Assign**.
- ___ b. Click to the right and under **Final Application Review** to add the **Assign** activity to the diagram.
- ___ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Create Output
- ___ d. Switch to the **Details** tab in the **Properties** view.
- ___ e. In the **Assign From** column, click the **Select From** link.

- __ f. Click **String** and type the following text in the window: Application was approved

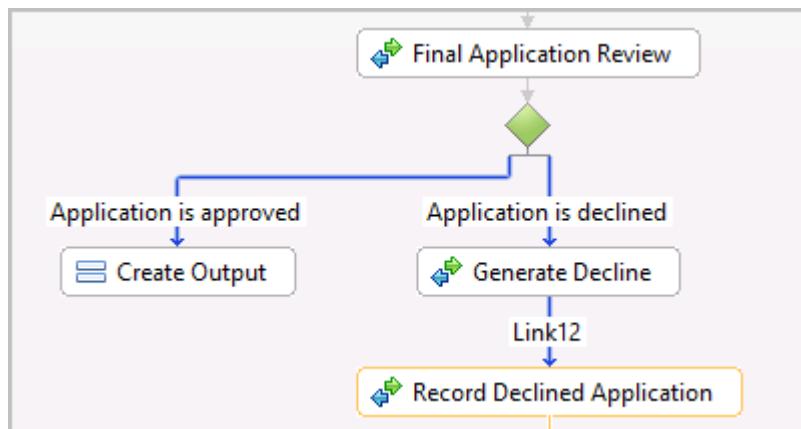


- __ g. Press Enter.
 __ h. In the **Assign To** column, click the **Select To** link.
 __ i. Expand **MessageVariable : Message** and select **message : string**.



- __ j. Save your changes. Continue to ignore any errors in the Problems view.
 __ 46. Link **Final Application Review** to **Create Output**. Change the link **Display Name** to: Application is approved
 __ a. Right-click **Final Application Review** and click **Add a link** from the menu.
 __ b. Click **Create Output** to add the link.
 __ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Application is approved

- ___ d. Save your changes. The current path in your diagram looks like the following figure:



- ___ 47. Set the link condition between Final Application Review and Create Output so that the process flows to Create Output if applicationDecision is set to true. If the reviewer determines that the customer application deserves to be approved, the application decision attribute is set to true.

- ___ a. Select the link between **Final Application Review** and **Create Output**.
 - ___ b. Switch to the **Details** tab in the **Properties** view.
 - ___ c. In the **Expression Language** field, select **Same as Process (Java)** from the list.
 - ___ d. For **Expression Type**, click **Java**.
 - ___ e. Copy the Final Application Review --> Create Output code snippet from
C:\labfiles\Support Files\Ex7\AccountVerification_snippets.txt.
- ```

// Final Application Review --> Create output
return CustomerApplicationvariable.getBoolean("applicationDecision");

```
- \_\_\_ f. Paste the snippet in the expression window over the existing text. Alternatively, you can replace the code with the following text:

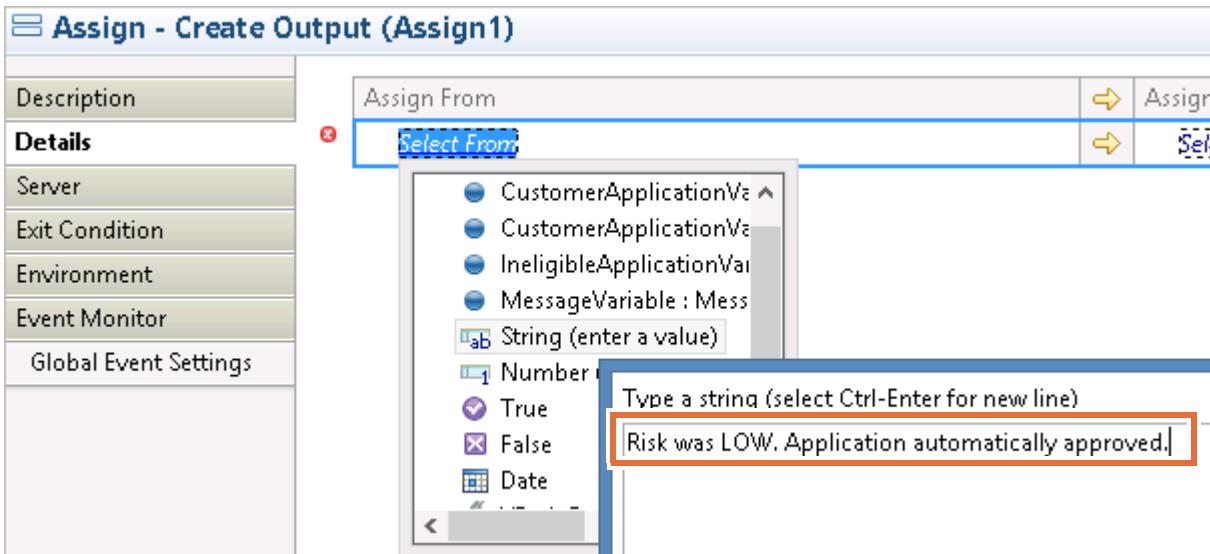
```
return CustomerApplicationVariable.getBoolean("applicationDecision");
```

- \_\_\_ 48. Save your changes.
- \_\_\_ 49. Add an assign activity to the right and under **Credit Risk Assessment** with the **Display Name**: Create Output

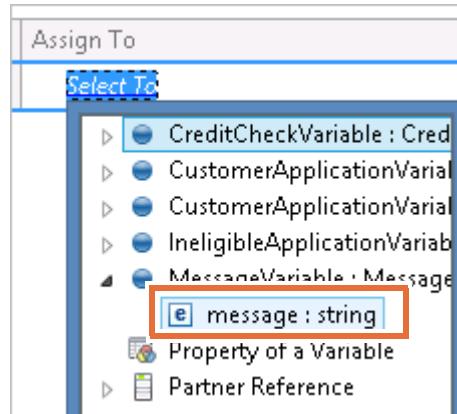
This Create Output activity sets the process output message to: Risk was LOW. Application automatically approved. The process returns this message when creditRisk is LOW, and the application flows directly from Credit Risk Assessment to Account Verification Reply.

- \_\_\_ a. In the palette, expand **Basic Actions** and select **Assign**.
- \_\_\_ b. Click under and to the right of **Credit Risk Assessment** to add the activity to the diagram.
- \_\_\_ c. On the **Description** tab in the **Properties** view, change the **Display Name** to: Create Output

- \_\_ d. Switch to the **Details** tab in the **Properties** view.
- \_\_ e. In the **Assign From** column, click the **Select From** link.
- \_\_ f. Click **String** and type the following text in the window: Risk was LOW. Application automatically approved.

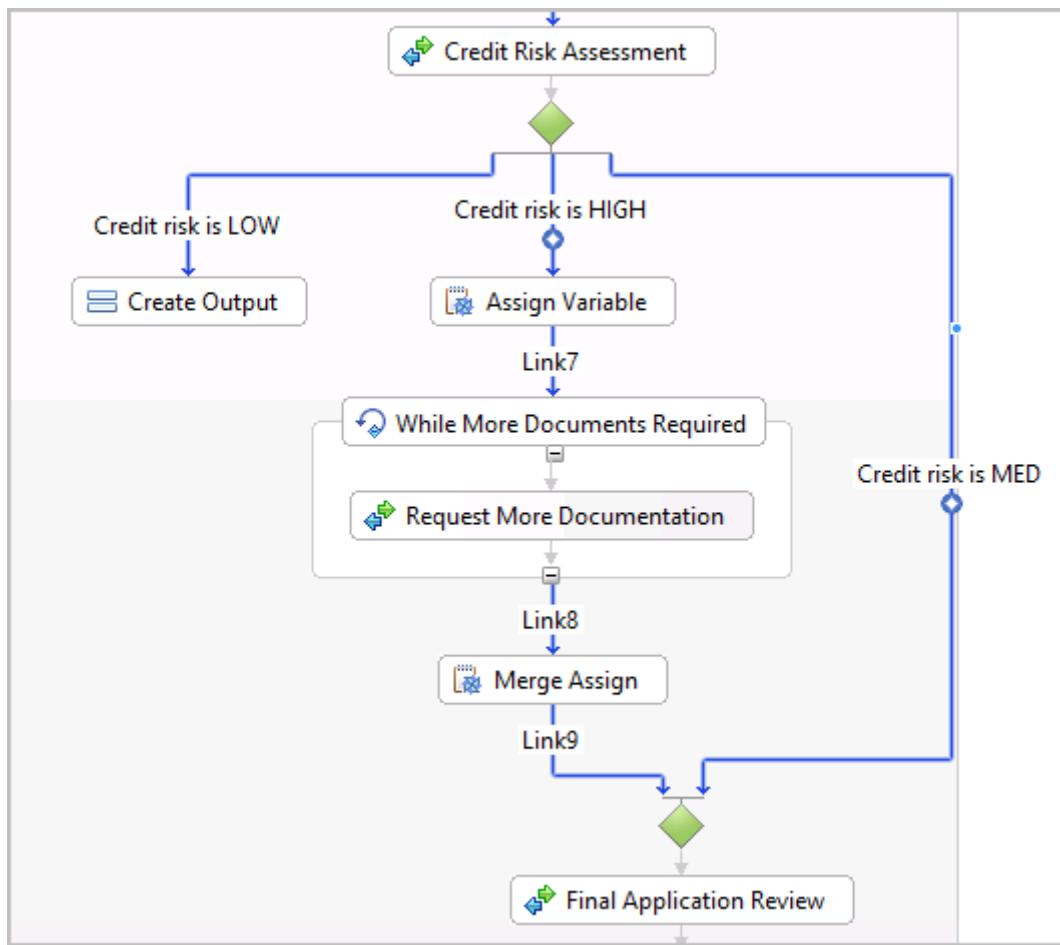


- \_\_ g. Press Enter.
- \_\_ h. In the **Assign To** column, click the **Select To** link.
- \_\_ i. Expand **MessageVariable : Message** and select **message : string**.



- \_\_ j. Save your changes. Continue to ignore any errors in the **Problems** view.
- \_ 50. Link **Credit Risk Assessment** to **Create Output** and change the link **Display Name** to: Credit risk is LOW
- \_\_ a. Right-click **Credit Risk Assessment** and click **Add a link** from the menu.
  - \_\_ b. Click **Create Output** to add the link.
  - \_\_ c. In the **Properties** view on the **Description** tab, change the **Display Name** to: Credit risk is LOW

- \_\_\_ d. Save your changes. The alignment might be slightly different from the following figure:



- \_\_\_ 51. Set the link condition between **Credit Risk Assessment** and **Create Output** so that the process flows directly to Create Output if creditRisk is **LOW**.

If the credit risk is **HIGH**, customer applications flow through Credit Risk Assessment to Assign Variable. If the credit risk is **MED** (short for “medium”), data flows through Credit Risk Assessment to Final Application Review and skips the gathering of more documentation. If the credit risk is **LOW**, data flows through Credit Risk Assessment to Create Output and then to Account Verification Reply.

- Click the link between **Credit Risk Assessment** and **Create Output**.
- Switch to the **Details** tab in the **Properties** view.
- In the **Expression Language** field, select **Same as Process (Java)**.
- For **Expression Type**, click **Java**.
- Copy the Credit Risk Assessment --> Create Output code snippet from  
C:\labfiles\Support Files\Ex7\AccountVerification\_snippets.txt.

```

//-----
// Credit Risk Assessment --> Create Output
//-----
return CustomerApplicationvariable.getstring("creditRisk").equals("Low");

```

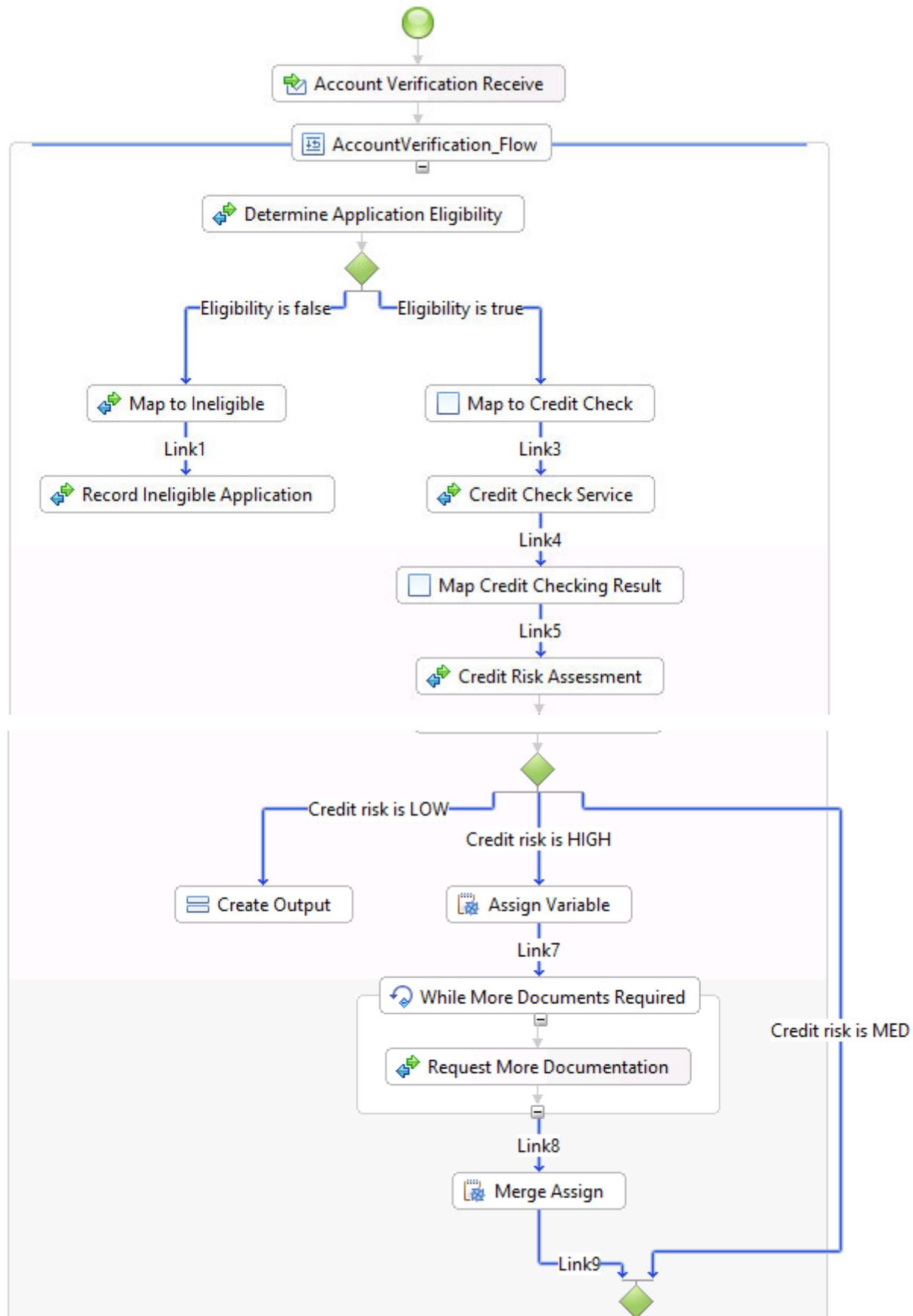
- \_\_ f. Paste the snippet in the expression window over the existing text. Alternatively, you can replace the code with the following text:

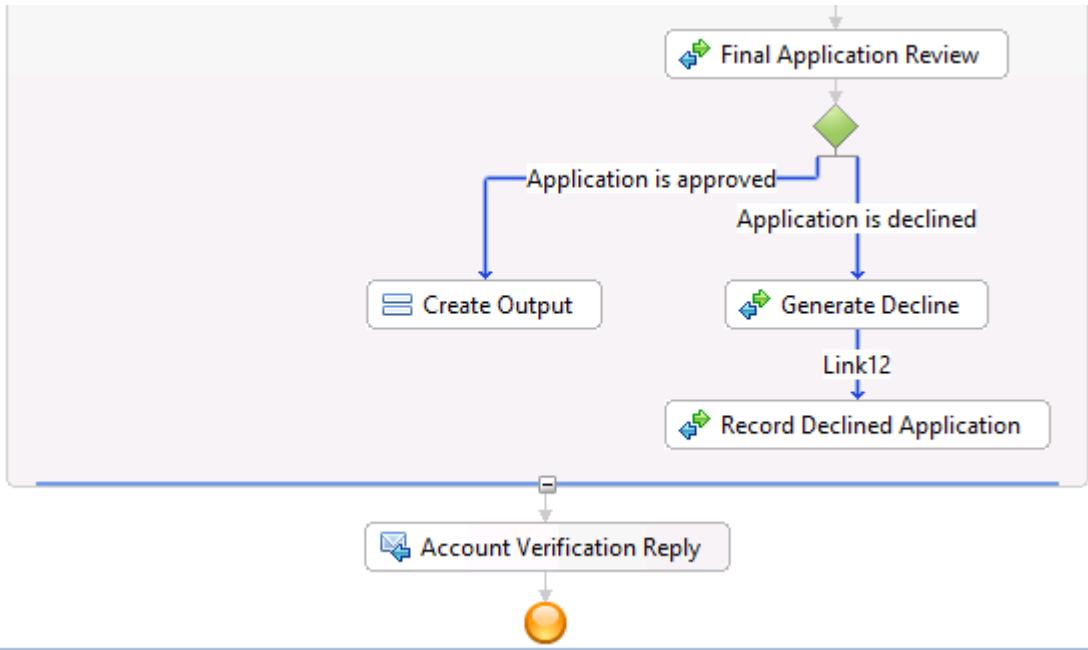
```
return CustomerApplicationVariable.getString("creditRisk").equals("LOW");
```

- \_\_ 52. Save your changes.  
\_\_ 53. Verify that the **Problems** view has no errors.  
\_\_ 54. Save your changes. Your process looks like the following diagram:



The diagram spans more than one page for clarity. You can also open the following .PNG image of the completed business process in your lab environment: C:\labfiles\Support Files\EX7\AccountVerification\_Complete.png

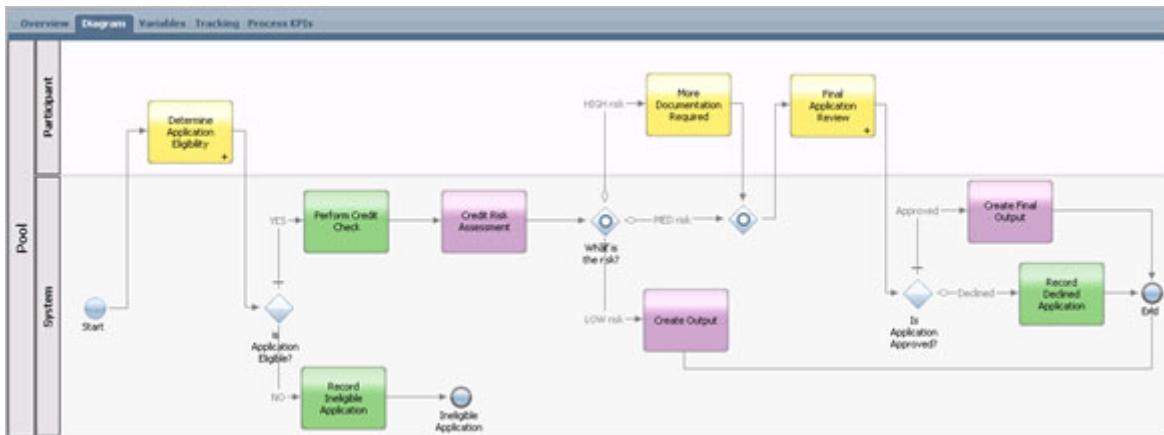




55. Close Windows Explorer and C:\labfiles\Support Files\Ex7\AccountVerification\_snippets.txt.

## Part 2: Compare the BPEL to the BPD in IBM Process Designer

Compare the business process that you built, captured on the previous page, to the “to be” model, which was mentioned earlier:



The diagrams are nearly identical.

Use the following process narrative to follow the business processes through their intended flow:

### Process narrative

- When the customer submits an application, the application must be tested for eligibility.

- If the application is ineligible, record the ineligible application in the database and stop the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is done against the customer's credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.
- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.
- If the customer is determined to be high risk, the customer must submit more documentation. Then, the customer must seek final approval for the application.
- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

## End of exercise

## Exercise review and wrap-up

In this exercise, you used IBM Integration Designer to implement basic and structured activities in a business process.

# Exercise 8. Creating a business process, part III

## What this exercise is about

In this exercise, you use IBM Integration Designer to finish building a complex BPEL business process. You use data maps to transform data, and you implement services that the process invokes. After completing the process, you add the business process to an SCA assembly diagram and test it.

## What you should be able to do

After completing this exercise, you should be able to:

- Use data maps to transform process data
- Use context variables to create a runtime process description
- Assemble an SCA application that contains a business process
- Test a business process in the IBM Integration Designer test environment

## Introduction

The Web Services Business Process Execution Language (WS-BPEL) provides syntax for specifying the behavior of a business process in a platform-independent manner. It is used to coordinate a series of service invocations to fulfill a business task. Although the language provides conditional and control logic, most of the work is done through the invoked services.

Combining BPEL with the SCA programming model allows for the coordination of SCA services into much larger units of work. Individual SCA services can be brought together and can benefit from the advanced capabilities of WS-BPEL.

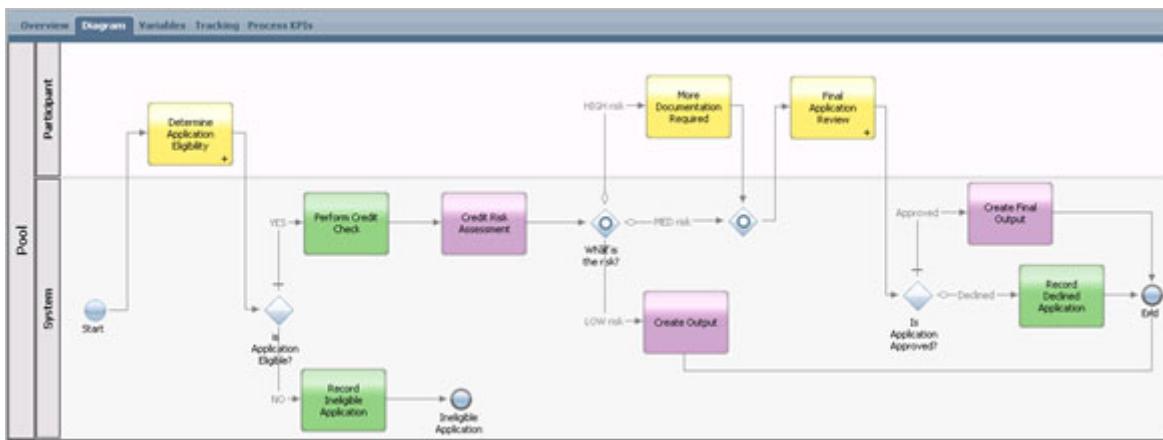
## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

## Exercise instructions

In this exercise, you create data maps to transform business objects between service requesters and providers. You also complete the core business logic for several of the services that the AccountVerification business process invokes. When you complete the service logic, you assemble the application that contains the AccountVerification process, and you test it.

Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



### Part 1: Use data maps to transform process data

The AccountVerification business process that you are implementing receives and operates on a CustomerApplication business object input. During process execution, your business process calls an external web service, CreditScoreService, which provides the credit score that is needed to determine the customer's credit risk. Unlike the AccountVerification process (which uses a CustomerApplication business object input), the CreditScoreService interface uses a CreditCheckRequest business object for both the input and the output.

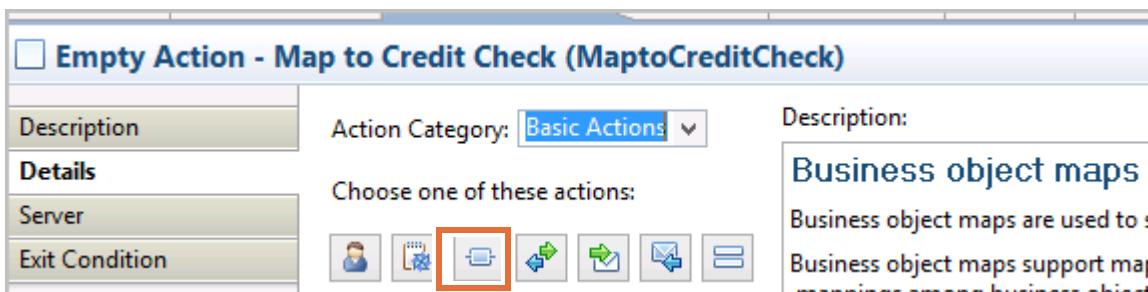
Because of this disparity, before you can invoke the CreditScoreService, you must transform the CustomerApplication business object into a CreditCheckRequest business object input. When the credit score is returned, you must transform the output from a CreditCheckRequest business object into a CustomerApplication business object.

The data map activity can be used to easily transform the data before and after service invocations in a business process. For more information about working with data maps, see the product documentation.

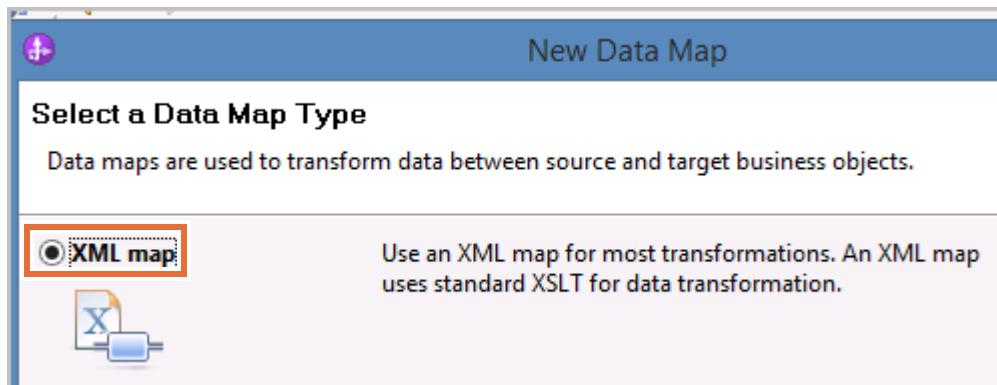
To create data maps to transform inputs and outputs:

1. Open the Exercise 8 workspace.
  - a. On your desktop, open the **Exercise Shortcuts** folder.
  - b. Double-click the **Exercise 8** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.

- \_\_\_ c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
  - \_\_\_ d. Close the **Getting Started** tab.
- \_\_\_ 2. Open the AccountVerification business process.
- \_\_\_ a. In the Business Integration view, expand **FoundationModule > Integration Logic > BPEL Processes > AccountVerification**.
  - \_\_\_ b. Double-click **AccountVerification** to open the process editor.
- \_\_\_ 3. Change the **Name** of the **Map to Credit Check** empty action to: **MaptoCreditCheck**
- \_\_\_ a. Select the **Map to Credit Check** empty action and switch to the **Description** tab in the **Properties** view.
  - \_\_\_ b. Change the value in the **Name** field from **EmptyAction** to: **MaptoCreditCheck**
  - \_\_\_ c. Save your changes.
- \_\_\_ 4. Change the implementation of the Map to Credit Check empty action to a data map. Create a **MaptoCreditCheck** data map that moves data from the **CustomerApplicationVariable** input to both a **CustomerApplicationVariable** output and a **CreditCheckVariable** output.
- You transform the data into both objects so you can preserve the existing data in **CustomerApplicationVariable** and merge the response data back into it.
- \_\_\_ a. Ensure that the **Map to Credit Check** empty action is selected, and switch to the **Details** tab in the **Properties** view.
  - \_\_\_ b. For **Choose one of these actions**, click the **Data Map** icon.



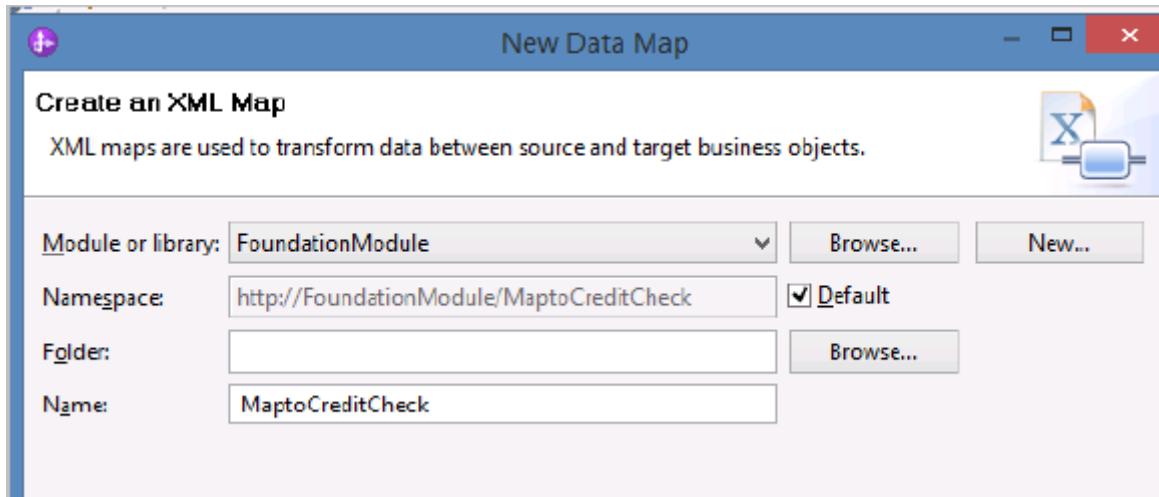
- \_\_\_ c. At the "Select a Data Map Type" dialog box, select **XML map**.



- \_\_\_ d. Click **Next**.

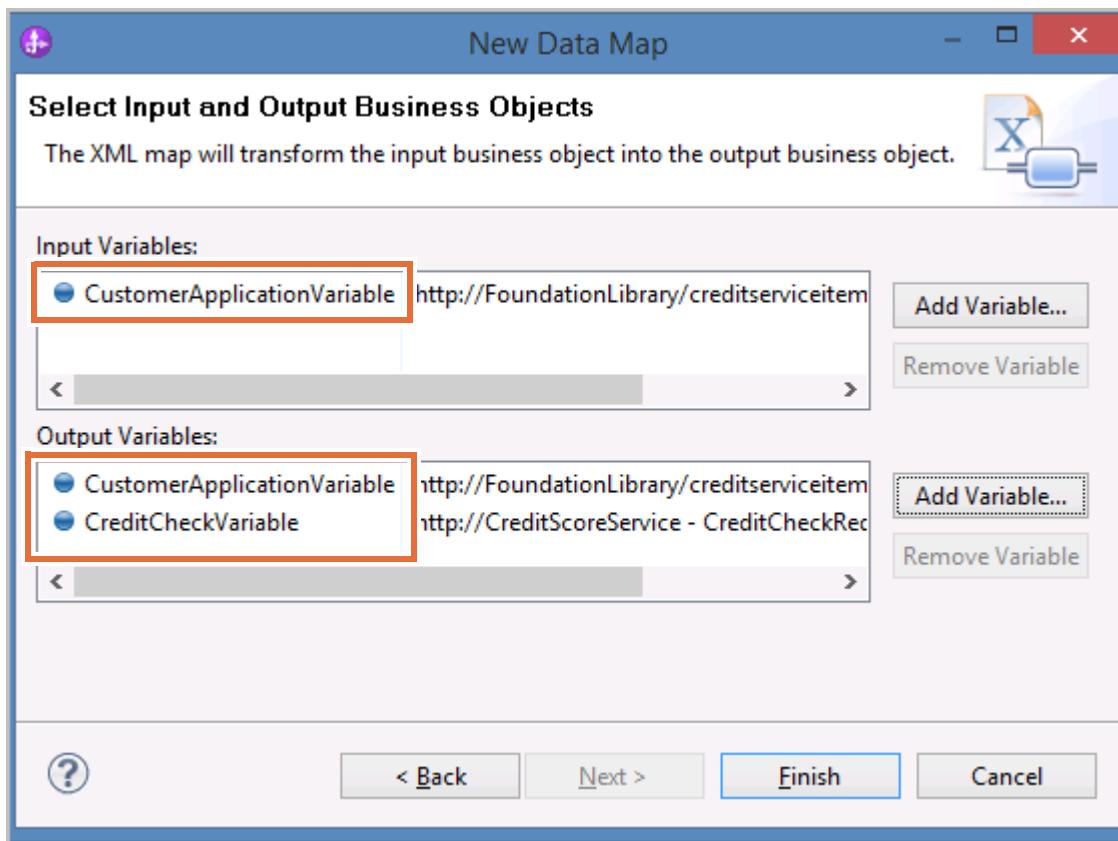
\_\_ e. At the “Create an XML Map” dialog box, follow these instructions:

- Verify that **Module or Library** is set to: FoundationModule
- Leave the **Folder** field empty
- In the **Name** field, type: MaptoCreditCheck



\_\_ f. Click **Next**.

- \_\_ g. At the “Select Input and Output Business Objects” dialog box, do the following steps:
- For **Input Variables**, click **Add Variable** and select **CustomerApplicationVariable** in the “Select a Variable” dialog box. Click **OK**.
  - For **Output Variables**, click **Add Variable** and select **CustomerApplicationVariable** in the “Select a Variable” dialog box. Click **OK**.
  - For **Output Variables**, click **Add Variable** again and select **CreditCheckVariable** in the “Select a Variable” dialog box. Click **OK**.  
(You cannot select more than one variable at the same time.)



- \_\_ h. Click **Finish** to open the XML mapping editor.

- 5. Use **Move** transformations to move the data from the **CustomerApplication** input variable to the **CustomerApplication** output variable. Use the **Automap input to output** feature to create the transformations automatically.
- a. In the data map editor, press and hold the Ctrl key, and select both the **CustomerApplication** input variable and the **CustomerApplication** output variable.

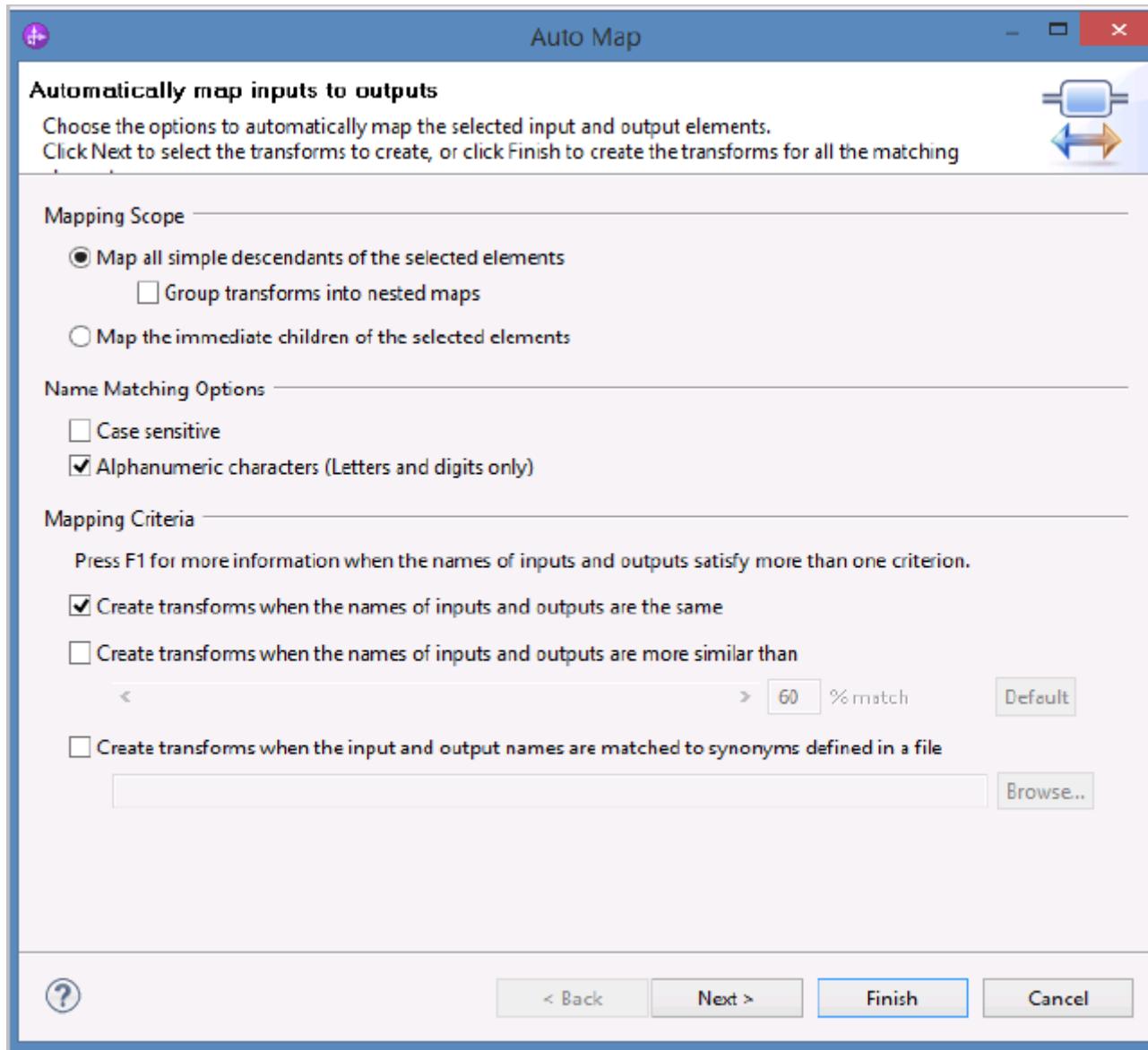
| CustomerApplication  |                     |        |         |
|----------------------|---------------------|--------|---------|
| <Click to filter...> |                     |        |         |
| e                    | accountNumber       | [0..1] | string  |
| e                    | applicationDate     | [0..1] | string  |
| e                    | applicationDecision | [0..1] | boolean |
| e                    | comments            | [0..1] | string  |
| e                    | companyName         | [0..1] | string  |
| e                    | contactFirstName    | [0..1] | string  |
| e                    | contactLastName     | [0..1] | string  |

| CustomerApplication  |                     |        |         |
|----------------------|---------------------|--------|---------|
| <Click to filter...> |                     |        |         |
| e                    | accountNumber       | [0..1] | string  |
| e                    | applicationDate     | [0..1] | string  |
| e                    | applicationDecision | [0..1] | boolean |
| e                    | comments            | [0..1] | string  |
| e                    | companyName         | [0..1] | string  |
| e                    | contactFirstName    | [0..1] | string  |
| e                    | contactLastName     | [0..1] | string  |

- b. Click the **Automap input to output** icon.

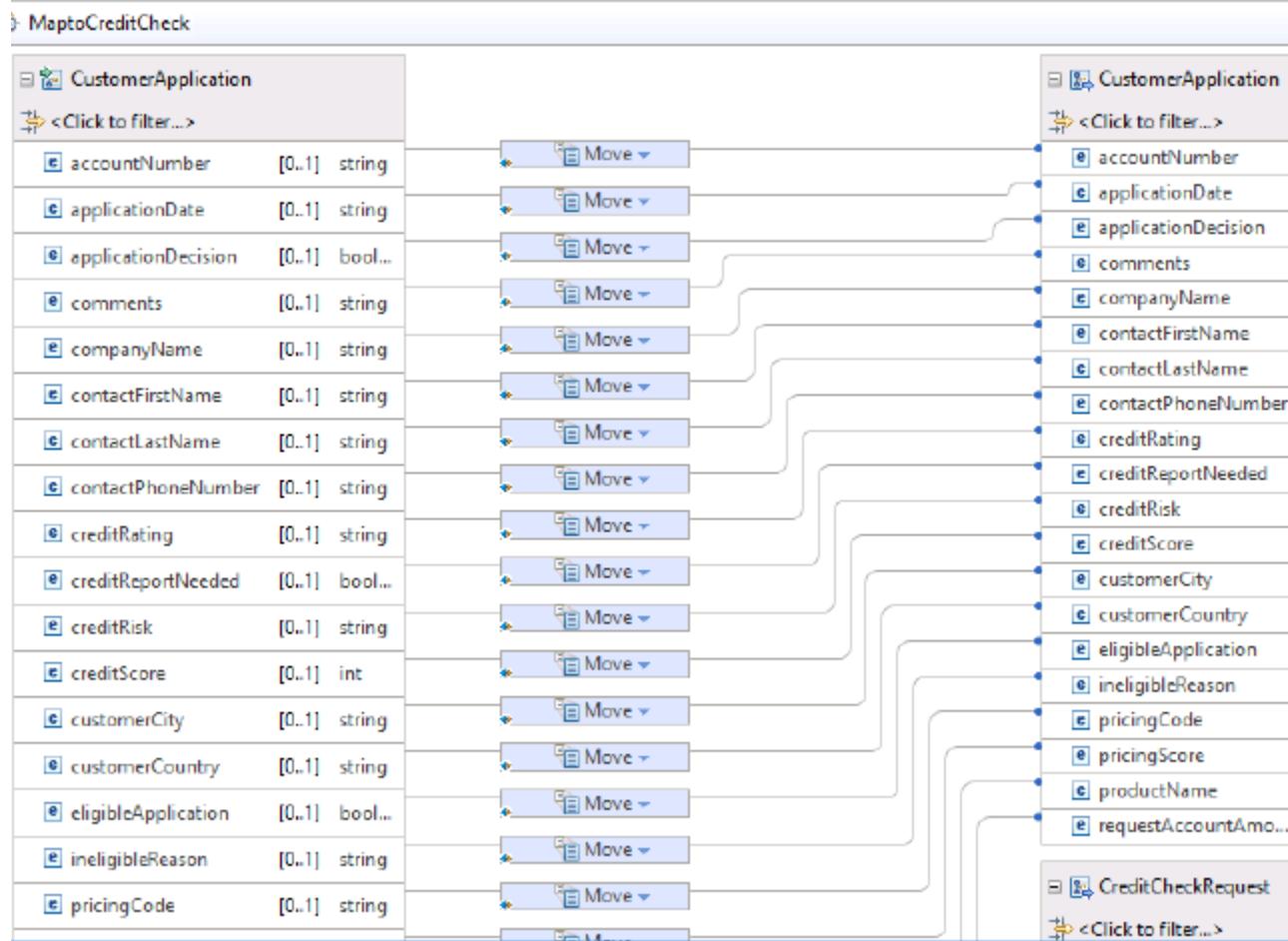


- \_\_\_ c. In the **Auto Map** window, accept defaults and click **Finish**.



- \_\_\_ d. This action uses **Move** transformations to map each of the fields from the `CustomerApplication` input to the `CustomerApplication` output.

Your map resembles the following diagram. The automap automatically mapped the accountNumber, companyName, and creditScore fields to both output variables. Some transformations are not visible in the graphic because of space constraints.

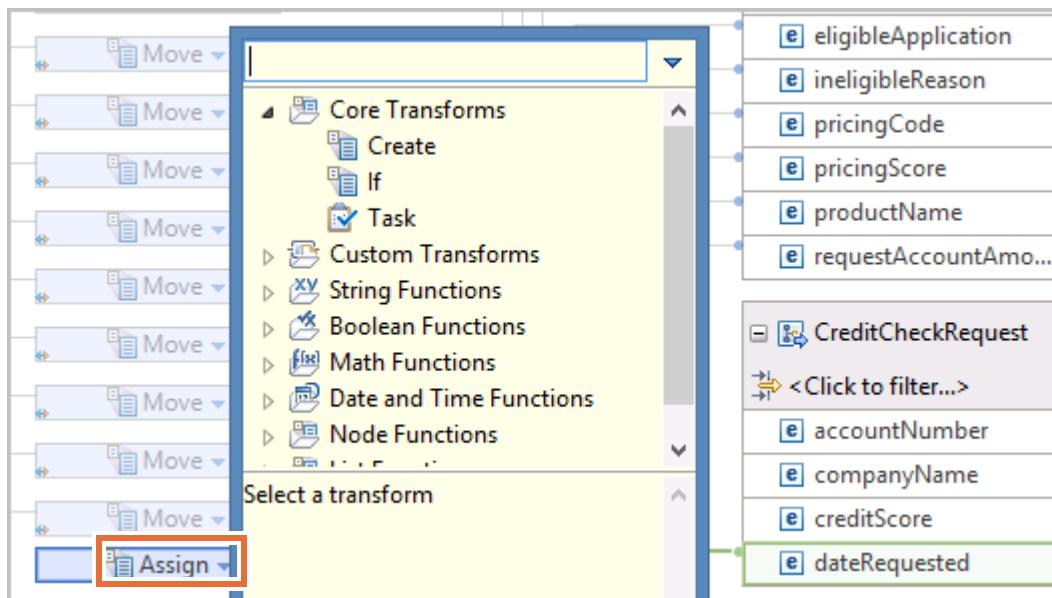


### Note

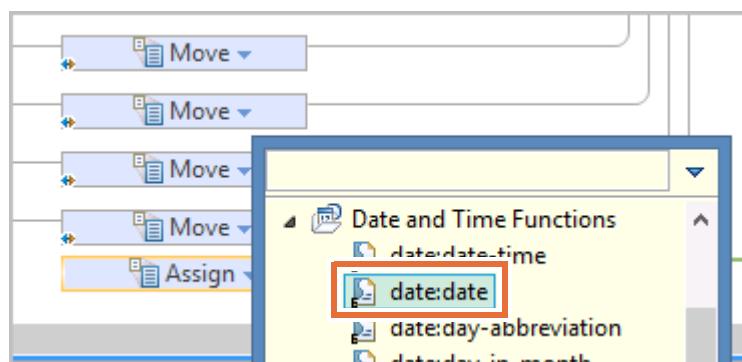
You can cut, copy, and paste transformations in the mapping editor. This option speeds development when reusing maps. For example, when building new maps from old maps, you can copy the schema from the old map to the new map or submap. If the target schema does not match the source schema that you copied, you are not allowed to paste.

- \_\_\_ 6. Create an **Assign** transformation that places the current date in the **dateRequested** attribute of the CreditCheckRequest output.
  - \_\_\_ a. Right-click **dateRequested** in the CreditCheckRequest output and click **Create Assign** from the menu.

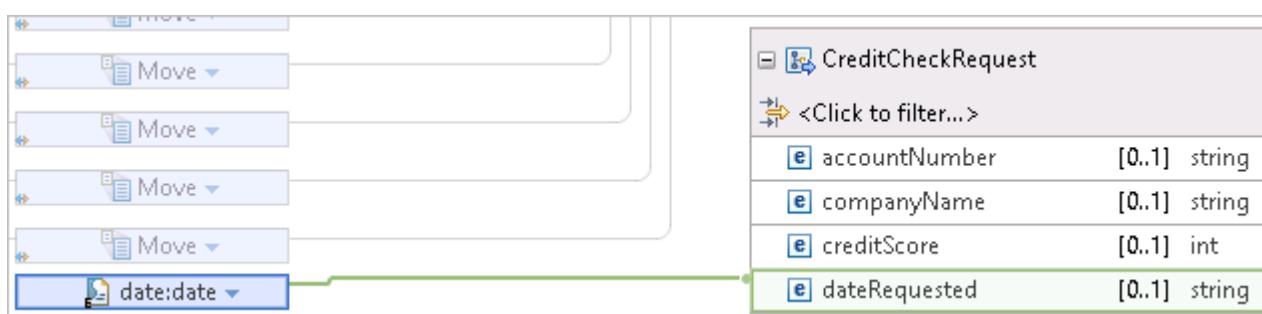
- \_\_\_ b. Click the down arrow in the Assign transformation to open the dialog box.



- \_\_\_ c. Expand Date and Time Functions and select date:date.

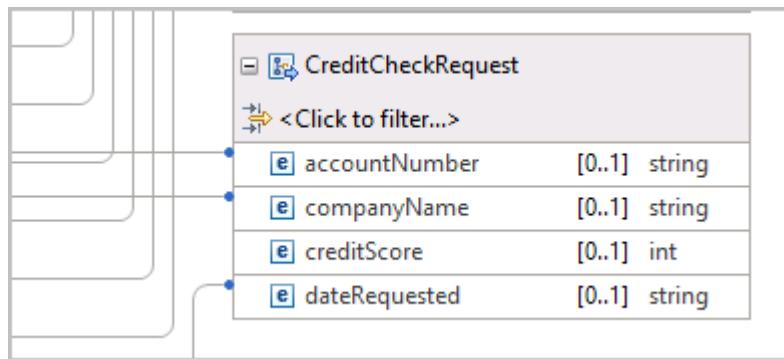


- \_\_\_ d. Your completed map resembles the following diagram. All transformations are not visible in the graphic because of space constraints.

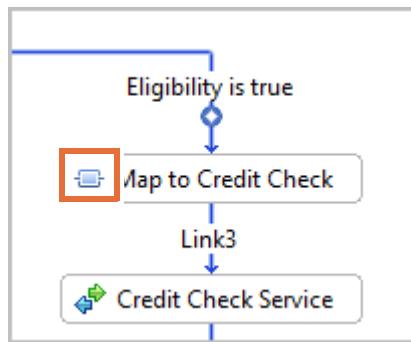


- \_\_\_ 7. Use **Move** transformations to manually map the **accountNumber**, **companyName**, and **creditScore** fields from the **CustomerApplication** input to the **CreditCheckRequest** output.
- \_\_\_ a. Hover over the **creditScore** field in the **CustomerApplication** input (on the left side of the diagram).

- \_\_\_ b. Drag the orange, circular handle from **creditScore** in the **CustomerApplication** input to **creditScore** in the **CreditCheckRequest** output. This action creates a **Move** transformation.
- \_\_\_ c. In the **CustomerApplication** input, right-click **accountNumber** and click **Create Connection**.
- \_\_\_ d. Click **accountNumber** in the **CreditCheckRequest** output. This action creates a **Move** transformation.
- \_\_\_ e. In the **CustomerApplication** input, right-click **companyName** and click **Create Connection**.
- \_\_\_ f. Click **companyName** in the **CreditCheckRequest** output. This action creates a **Move** transformation.

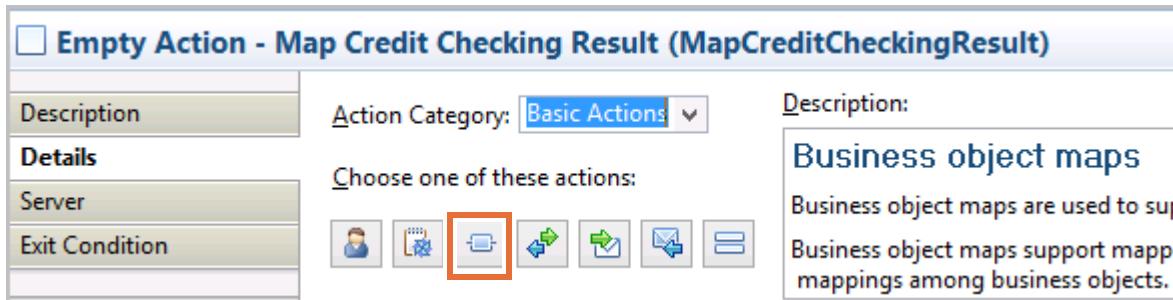


- \_\_\_ 8. Save and close the data map. Verify that the **Problems** view shows no errors. If you see any errors, then verify that you created the data map correctly.
- \_\_\_ 9. Save the changes to the business process. The icon for the Map to Credit Check activity changes to a data map icon.

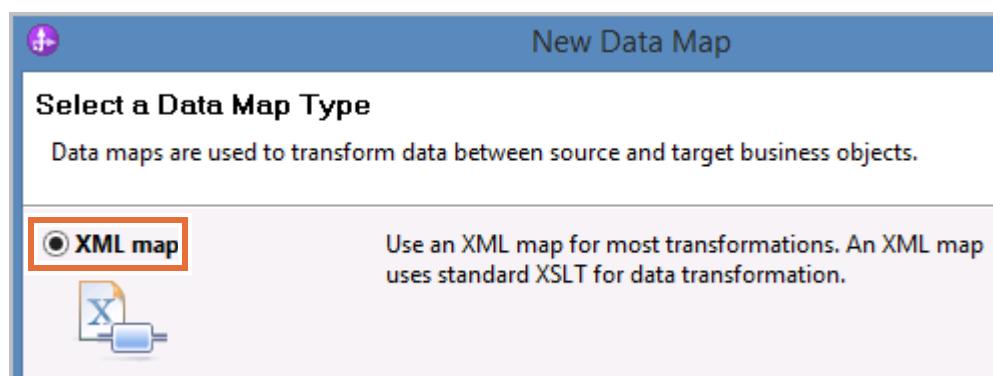


- \_\_\_ 10. Change the **Name** of the **Map Credit Checking Result** empty action to: **MapCreditCheckingResult**
  - \_\_\_ a. Select the **Map Credit Checking Result** empty action and switch to the **Description** tab in the **Properties** view.
  - \_\_\_ b. Change the value in the **Name** field from **EmptyAction1** to: **MapCreditCheckingResult**
  - \_\_\_ c. Save your changes.

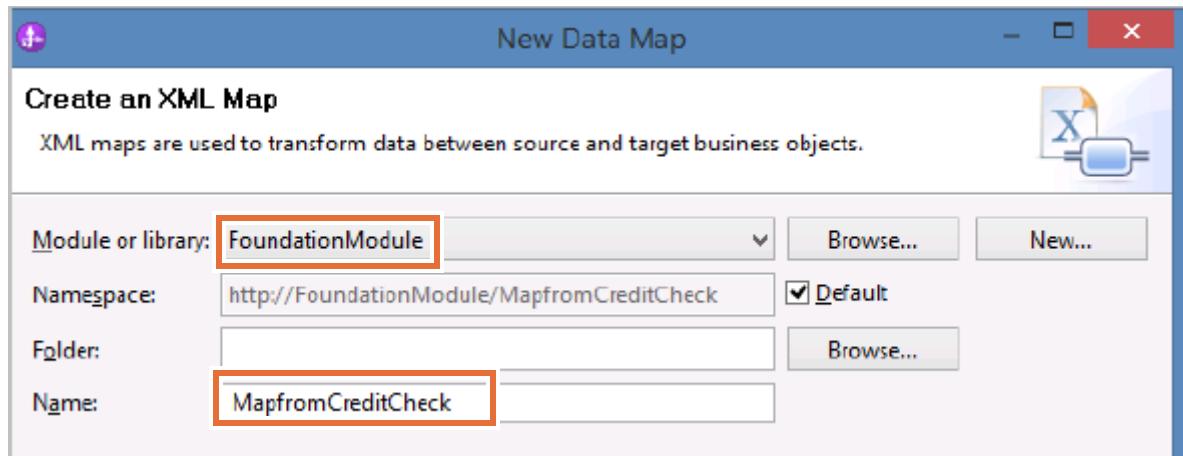
11. Change the implementation of the Map Credit Checking Result empty action to a data map. Create a MapfromCreditCheck data map that merges data from the CreditCheckVariable and CustomerApplicationVariable inputs into a CustomerApplicationVariable output.
- Select the **Map Credit Checking Result** empty action and switch to the **Details** tab in the **Properties** view.
  - For **Choose one of these actions**, click the **Data Map** icon.



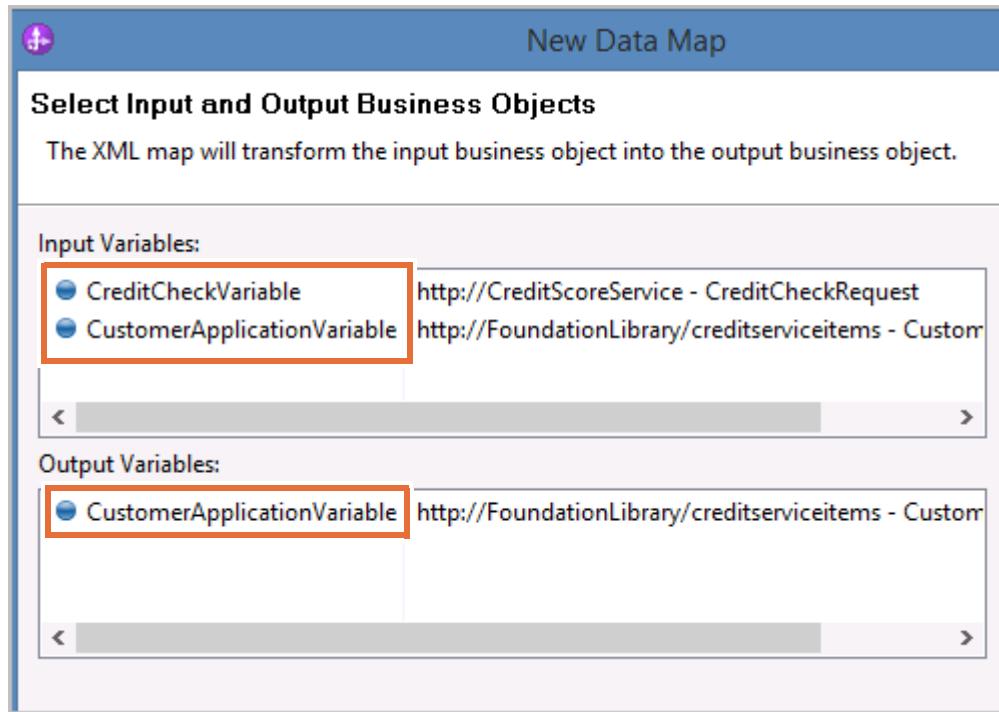
- At the “Select a Data Map Type” dialog box, select **XML map**.



- Click **Next**.
- At the “Create an XML Map” dialog box, enter the following information.
  - Verify that **Module or Library** is set to: FoundationModule
  - Leave the **Folder** field empty
  - In the **Name** field, type: MapfromCreditCheck



- \_\_\_ f. Click **Next**.
- \_\_\_ g. At the “Select Input and Output Business Objects” dialog box, do these steps.
- For **Input Variables**, click **Add Variable** and select **CreditCheckVariable** in the “Select a Variable” dialog box.
  - For **Input Variables**, click **Add Variable** and select **CustomerApplicationVariable** in the “Select a Variable” dialog box.
  - For **Output Variables**, click **Add Variable** and select **CustomerApplicationVariable**.



- \_\_\_ h. Click **Finish** to open the data map editor.
- \_\_\_ 12. Map the Credit Check result data to the CustomerApplicationVariable.
- \_\_\_ a. Hold down the Control key and select the **CustomerApplication** and **CreditCheckRequest** input variables and the **CustomerApplication** output variable.

- \_\_\_ b. Click the Automap input to output icon.

| CreditCheckRequest   |        |        |
|----------------------|--------|--------|
| <b>accountNumber</b> | [0..1] | string |
| <b>companyName</b>   | [0..1] | string |
| <b>creditScore</b>   | [0..1] | int    |
| <b>dateRequested</b> | [0..1] | string |

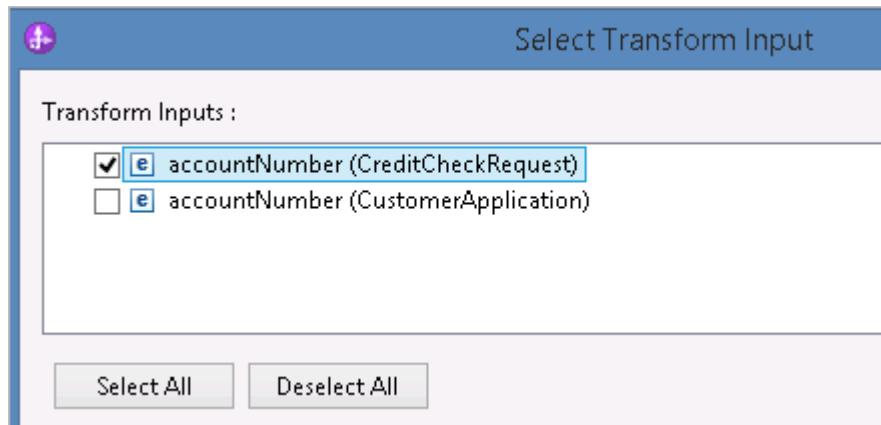
  

| CustomerApplication        |        |         |
|----------------------------|--------|---------|
| <b>accountNumber</b>       | [0..1] | string  |
| <b>applicationDate</b>     | [0..1] | string  |
| <b>applicationDecision</b> | [0..1] | boolean |
| <b>comments</b>            | [0..1] | string  |
| <b>companyName</b>         | [0..1] | string  |
| <b>contactFirstName</b>    | [0..1] | string  |
| <b>contactLastName</b>     | [0..1] | string  |
| <b>contactPhoneNumber</b>  | [0..1] | string  |
| <b>creditRating</b>        | [0..1] | string  |
| <b>creditReportNeeded</b>  | [0..1] | boolean |
| <b>creditRisk</b>          | [0..1] | string  |
| <b>creditScore</b>         | [0..1] | int     |
| <b>customerCity</b>        | [0..1] | string  |
| <b>customerCountry</b>     | [0..1] | string  |
| <b>eligibleApplication</b> | [0..1] | boolean |
| <b>ineligibleReason</b>    | [0..1] | string  |
| <b>pricingCode</b>         | [0..1] | string  |
| <b>pricingScore</b>        | [0..1] | string  |
| <b>productName</b>         | [0..1] | string  |
| <b>requestAccountA...</b>  | [0..1] | int     |

- \_\_\_ c. Accept the defaults in the Auto Map dialog box. Click **Next**.  
 \_\_\_ d. Click **accountNumber** in the Transform Outputs list. Click **Edit**.

| Transforms found :                                             |                                                  |             |
|----------------------------------------------------------------|--------------------------------------------------|-------------|
| Transform Outputs                                              | Transform Inputs                                 | Input count |
| <input checked="" type="checkbox"/> <b>CustomerApplication</b> |                                                  |             |
| <input checked="" type="checkbox"/> <b>accountNumber</b>       | accountNumber (CreditCheckRequest) , accountN... | 2           |
| <input checked="" type="checkbox"/> <b>applicationDate</b>     | applicationDate (CustomerApplication)            | 1           |
| <input checked="" type="checkbox"/> <b>applicationDecision</b> | applicationDecision (CustomerApplication)        | 1           |
| <input checked="" type="checkbox"/> <b>comments</b>            | comments (CustomerApplication)                   | 1           |
| <input checked="" type="checkbox"/> <b>companyName</b>         | companyName (CreditCheckRequest) , companyN...   | 2           |
| <input checked="" type="checkbox"/> <b>contactFirstName</b>    | contactFirstName (CustomerApplication)           | 1           |
| <input checked="" type="checkbox"/> <b>contactLastName</b>     | contactLastName (CustomerApplication)            | 1           |
| <input checked="" type="checkbox"/> <b>contactPhoneNumber</b>  | contactPhoneNumber (CustomerApplication)         | 1           |

- \_\_ e. Clear the check box next to **accountNumber (CustomerApplication)** and click **OK**.

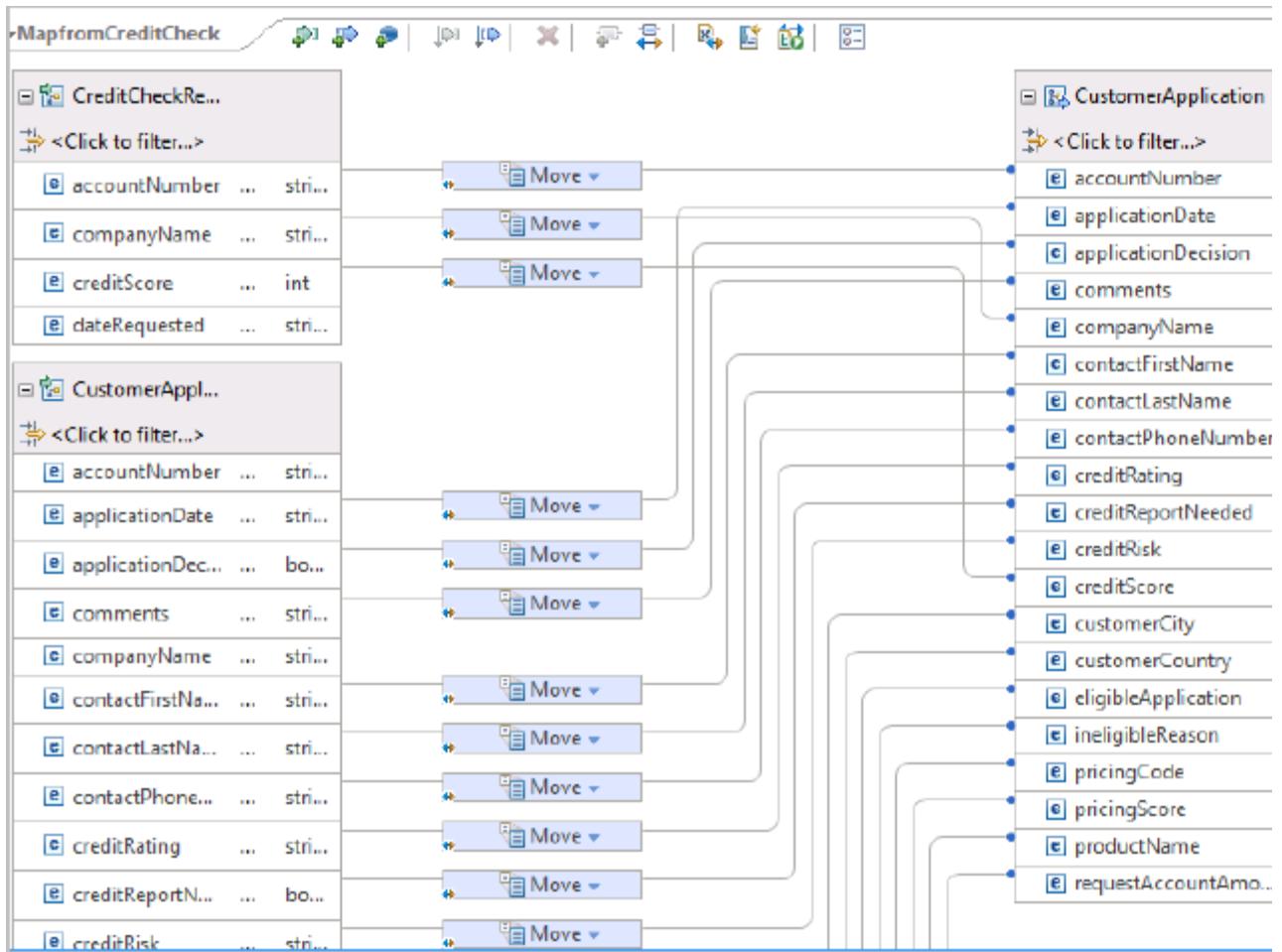


- \_\_ f. Click **companyName** in the Transform Outputs list. Click **Edit**.
- \_\_ g. Clear the check box next to **companyName (CustomerApplication)**. Click **OK**.
- \_\_ h. Click **creditScore** in the Transform Outputs list. Click **Edit**.
- \_\_ i. Clear the check box next to **creditScore (CustomerApplication)**. Click **OK**.
- \_\_ j. Verify the edits that you made. No input field has an Input count other than 1.

| Auto Map                                                  |                                           |             |
|-----------------------------------------------------------|-------------------------------------------|-------------|
| Select transforms to create                               |                                           |             |
| Transform Outputs                                         | Transform Inputs                          | Input count |
| <input type="checkbox"/> e CustomerApplication            |                                           |             |
| <input type="checkbox"/> e accountNumber                  | accountNumber (CreditCheckRequest)        | 1           |
| <input checked="" type="checkbox"/> e applicationDate     | applicationDate (CustomerApplication)     | 1           |
| <input checked="" type="checkbox"/> e applicationDecision | applicationDecision (CustomerApplication) | 1           |
| <input checked="" type="checkbox"/> e comments            | comments (CustomerApplication)            | 1           |
| <input type="checkbox"/> e companyName                    | companyName (CreditCheckRequest)          | 1           |
| <input checked="" type="checkbox"/> e contactFirstName    | contactFirstName (CustomerApplication)    | 1           |
| <input checked="" type="checkbox"/> e contactLastName     | contactLastName (CustomerApplication)     | 1           |
| <input checked="" type="checkbox"/> e contactPhoneNumber  | contactPhoneNumber (CustomerApplication)  | 1           |
| <input checked="" type="checkbox"/> e creditRating        | creditRating (CustomerApplication)        | 1           |
| <input checked="" type="checkbox"/> e creditReportNeeded  | creditReportNeeded (CustomerApplication)  | 1           |
| <input checked="" type="checkbox"/> e creditRisk          | creditRisk (CustomerApplication)          | 1           |
| <input type="checkbox"/> e creditScore                    | creditScore (CreditCheckRequest)          | 1           |
| <input checked="" type="checkbox"/> e customerCity        | customerCity (CustomerApplication)        | 1           |
| <input checked="" type="checkbox"/> e customerCountry     | customerCountry (CustomerApplication)     | 1           |
| <input checked="" type="checkbox"/> e eligibleApplication | eligibleApplication (CustomerApplication) | 1           |
| <input checked="" type="checkbox"/> e ineligibleReason    | ineligibleReason (CustomerApplication)    | 1           |
| <input checked="" type="checkbox"/> e pricingCode         | pricingCode (CustomerApplication)         | 1           |
| <input checked="" type="checkbox"/> e pricingScore        | pricingScore (CustomerApplication)        | 1           |

- \_\_ k. Click **Finish**.

- I. Your data map resembles the following figure. Some transformations are not visible in the graphic because of space constraints.

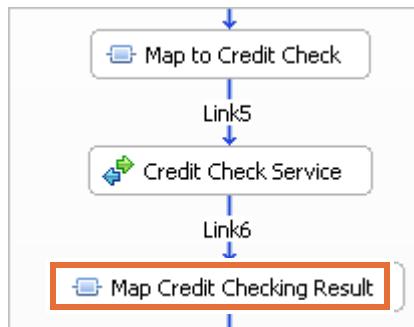


### Note

You can also manually map inputs to outputs by clicking a field in the output list, then dragging a connector line to an input field. If you select an individual transformation (by clicking the transformation name or the connector), the connector and the mapped attributes are highlighted in green. You can use this display to verify the accuracy of the transformations that you created.

- 13. Save and close the data map.

- \_\_\_ 14. Save the changes to the business process. The icon for the **Map Credit Checking Result** activity changes to a data map icon.

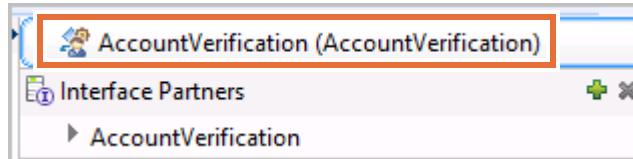


## Part 2: Use context variables to create a runtime process description

When used in descriptions for processes and human tasks, replacement expressions can be used to represent context variables that are fully resolved in the runtime environment. Context variables can come from many sources. For example, they can originate from input or output messages. Context variables can be used in many places within the tool environment, including process descriptions. For more information about replacement expressions, see the product documentation.

In this portion of the exercise, you use context variables to create a process description. At run time, when the context variables are resolved, the replacement expression creates a meaningful description of process instances when you view them in clients such as the Business Process Choreographer Explorer. For process instances, you use information in the process input message to populate the process description at run time.

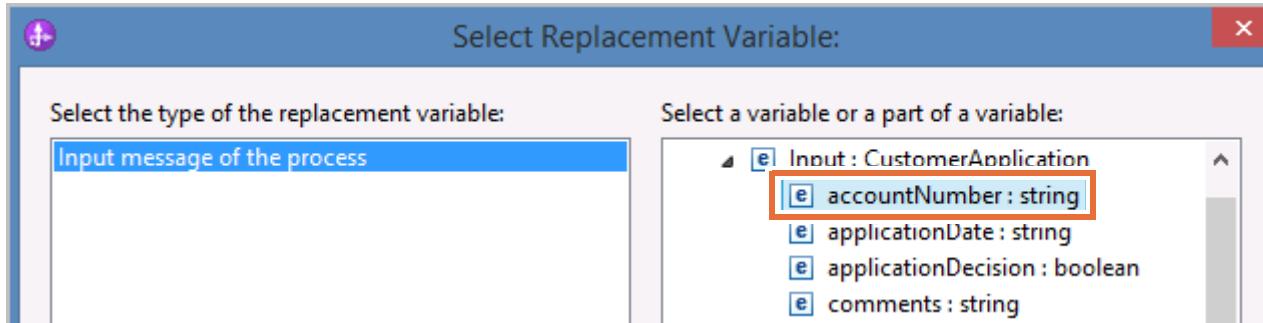
- \_\_\_ 1. Create a process description that contains context variables.
- \_\_\_ a. Click any blank space in the process editor *outside* the `AccountVerification_Flow` activity. You can also click the `AccountVerification` icon at the top of the tray.



- \_\_\_ b. Switch to the **Description** tab in the **Properties** view.
- \_\_\_ c. In the **Description** field, type the following text: Account verification for (add a space at the end of the text)

| Process - AccountVerification (AccountVerification) |                      |                                             |
|-----------------------------------------------------|----------------------|---------------------------------------------|
| <b>Description</b>                                  | <b>Name:</b> *       | AccountVerification                         |
| <b>Details</b>                                      | <b>Display name:</b> | AccountVerification                         |
| <b>Server</b>                                       | <b>Namespace:</b> *  | http://FoundationModule/AccountVerification |
| <b>Administration</b>                               | <b>Description:</b>  | Account verification for                    |
| <b>Java Imports</b>                                 |                      |                                             |

- \_\_\_ d. To the right of the **Description** field, click **Insert Variable**.
- \_\_\_ e. In the “Select Replacement Variable” dialog box, in the “Select a variable or a part of a variable” window, expand **InputCriterionParameters : InputCriterion > Input : CustomerApplication**.
- \_\_\_ f. Select **accountNumber : string**.



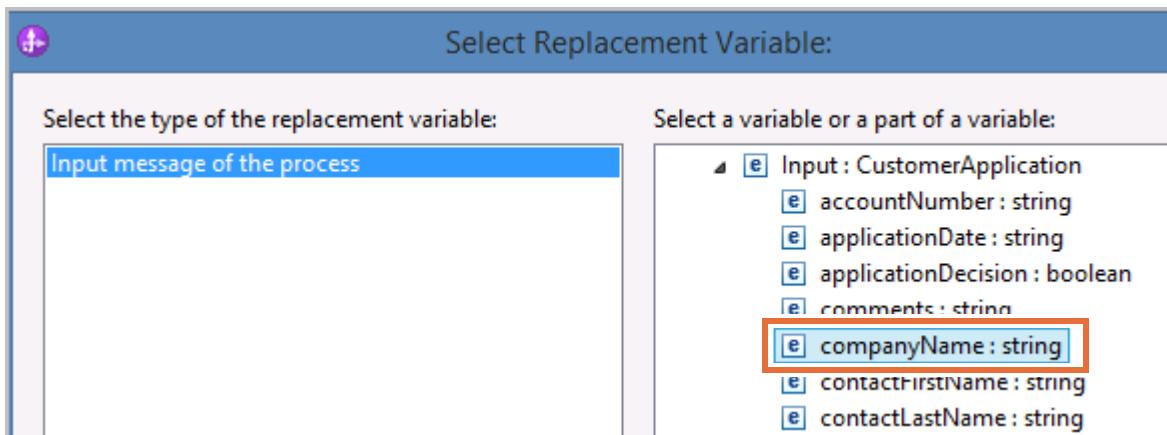
Note the replacement variable syntax at the bottom of the dialog box.

The replacement variable will be: %InputCriterionParameters\Input\accountNumber%

- \_\_\_ g. Click **OK**. The replacement variable expression is displayed in the **Description** field.

|                      |                                                                            |
|----------------------|----------------------------------------------------------------------------|
| <u>Name:</u> *       | AccountVerification                                                        |
| <u>Display name:</u> | AccountVerification                                                        |
| <u>Namespace:</u> *  | http://FoundationModule/AccountVerification                                |
| <u>Description:</u>  | Account verification for%\\InputCriterionParameters\\Input\\accountNumber% |

- \_\_\_ h. Insert a space at the end of the **Description** field.
- \_\_\_ i. Click **Insert Variable**.
- \_\_\_ j. In the “Select Replacement Variable” dialog box, in the “Select a variable or a part of a variable” window, expand **InputCriterionParameters : InputCriterion > Input : CustomerApplication**.
- \_\_\_ k. Select **companyName : string**.



- \_\_ I. Click **OK**. The completed expression resembles the following text:

```
Account verification for %InputCriterionParameters\Input/accountNumber%
%InputCriterionParameters\Input/companyName%
```

When you view a process instance at run time, this expression is resolved into a description such as: Account verification for IBM007 IBM

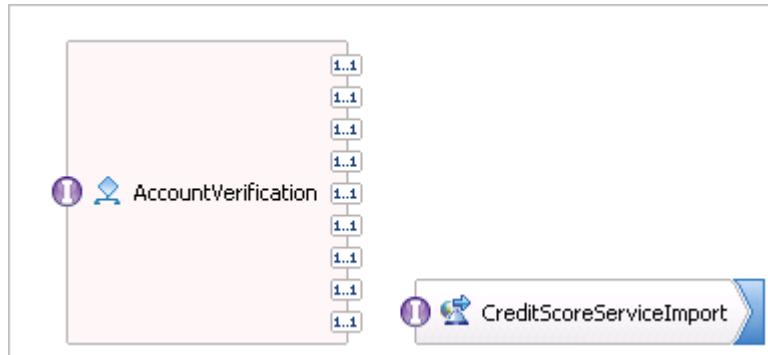
- \_\_ 2. Save your changes.

### **Part 3: Assemble an SCA application that contains a business process**

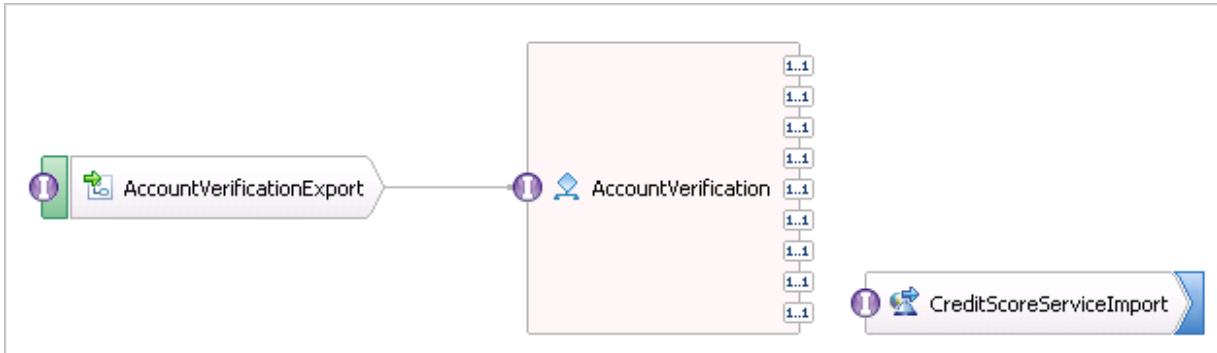
In this portion of the exercise, you modify the assembly diagram for `FoundationModule` and implement the services that activities in the business process invoke. Some of the services are implemented with simple Java code that is designed to simulate actual services. Others, such as human tasks and adapters, are implemented in later exercises.

To assemble the module and implement the services:

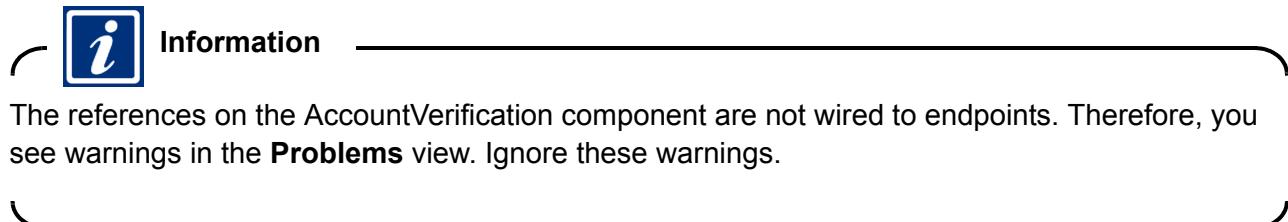
- \_\_ 1. Add the `AccountVerification` service component to the `FoundationModule` assembly diagram and generate an export component with an SCA binding.
  - \_\_ a. In the Business Integration view, expand **FoundationModule**.
  - \_\_ b. Double-click **Assembly Diagram** to open the assembly editor.
  - \_\_ c. Expand **FoundationModule > Integration Logic > BPEL Processes > AccountVerification**.
  - \_\_ d. Drag the **AccountVerification** business process onto the assembly diagram.



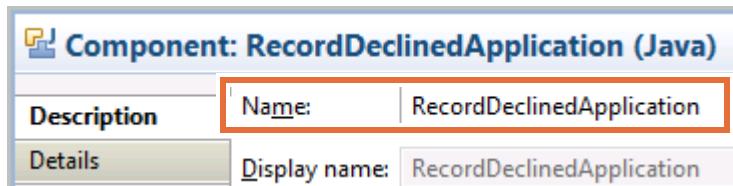
- \_\_ e. Right-click **AccountVerification** and click **Generate Export > SCA Binding** from the menu.



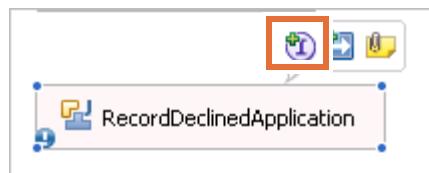
- ## 2. Save your changes.



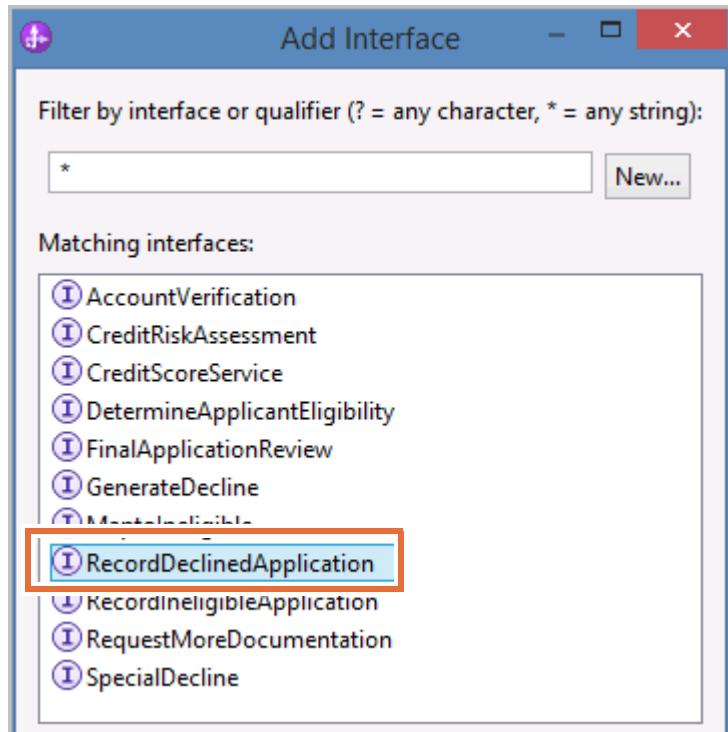
- \_\_\_ 3. Add a Java component to the assembly diagram and name the component:  
RecordDeclinedApplication
    - \_\_\_ a. In the assembly editor palette, expand **Components** and select **Java**.
    - \_\_\_ b. Click any blank space on the assembly diagram to add the component.
    - \_\_\_ c. With the component that is selected, switch to the **Description** tab in the **Properties** view.
    - \_\_\_ d. Change the **Name** to: RecordDeclinedApplication



- \_\_ 4. Add the RecordDeclinedApplication interface to the Java component.
    - \_\_ a. Select the **RecordDeclinedApplication** component in the assembly diagram.
    - \_\_ b. Click the **Add Interface** icon in the action bar.



- \_\_ c. In the Add Interface dialog box, select **RecordDeclinedApplication**.

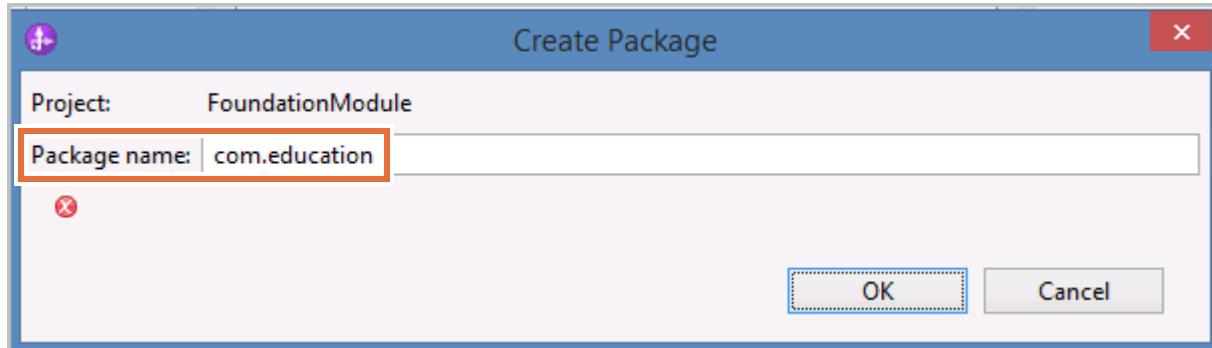


#### Information

By using the **Add Interface** dialog box, you can filter interfaces by type: Java, WSDL, or both.

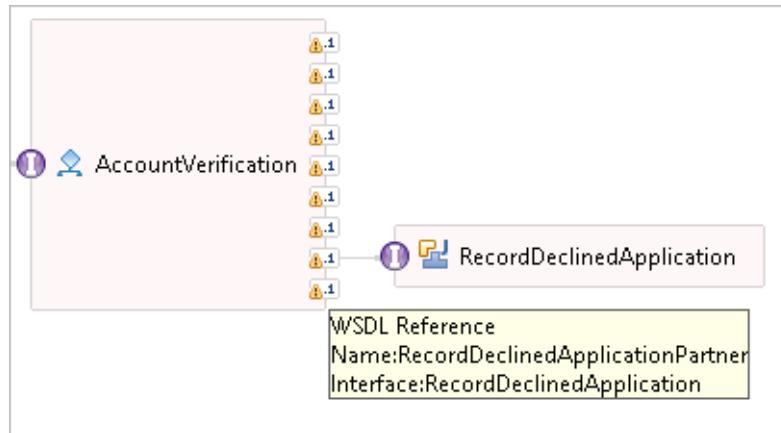
- \_\_ d. Click **OK**.
- \_\_ 5. Use the snippet in  
C:\labfiles\Support Files\Ex8\RecordDeclinedApplicationImpl\_snippet.txt to generate the implementation for the Java component. Generate the implementation in a new package that is named: RecordDeclinedApplication
- \_\_ a. Right-click the **RecordDeclinedApplication** Java component and click **Generate Implementation** from the menu.
- \_\_ b. In the Generate Implementation dialog box, click **New Package**.

- \_\_\_ c. In the Create Package dialog box, in the **Package Name** field, type: com.education

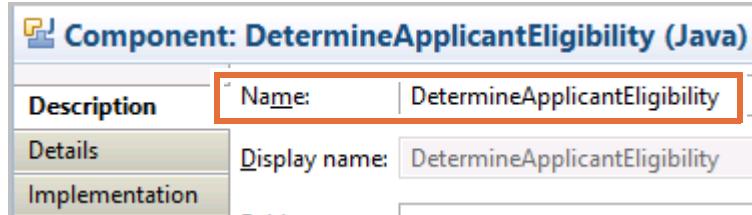


- \_\_\_ d. Click **OK**.
- \_\_\_ e. With **com.education** selected in the Generate Implementation dialog box, click **OK**.  
This action opens `RecordDeclinedApplicationImpl.java` in the Java editor.
- \_\_\_ f. In the **RecordDeclinedApplicationImpl.java** tab, scroll to the `public void InputCriterion` declaration at the end of the file.
- \_\_\_ g. In Windows Explorer, browse to `C:\labfiles\Support Files\Ex8`.
- \_\_\_ h. Open `RecordDeclinedApplicationImpl_snippet.txt` in a text editor such as Notepad.
- \_\_\_ i. Copy the text in `RecordDeclinedApplicationImpl_snippet.txt` and paste the content over the comment lines (the green text that begins with `//`) in the `InputCriterion` method.
- ```
public void InputCriterion(DataObject input) {
    System.out.println("[Java] Record Declined Application - begins");
    String ret = "Compliance Logging: Application from customer "
        + input.getString("companyName") + " has been declined.";
    System.out.println("[Java] Record Declined Application - ends");
}
```
- ___ j. Alternatively, enter the following code:
- ```
System.out.println("[Java] Record Declined Application - begins");
String ret = "Compliance Logging: Application from customer " +
 input.getString("companyName") + " has been declined.";
System.out.println("[Java] Record Declined Application - ends");
```
- \_\_\_ k. Press **Ctrl+S** to save your changes and **Ctrl + W** to close the Java editor.
- \_\_\_ l. Close the `RecordDeclinedApplicationImpl_snippet.txt` file. Leave Windows Explorer open.
- \_\_\_ 6. Wire the `RecordDeclinedApplication` Java component to the `RecordDeclinedApplicationPartner` reference of the `AccountVerification` process.
- \_\_\_ a. Right-click **RecordDeclinedApplication** and click **Wire to existing** from the menu.

- \_\_\_ b. The Java component is automatically wired to the **RecordDeclinedApplicationPartner** reference. To verify this connection, hover the mouse over the reference and the name that is displayed in a dialog box.

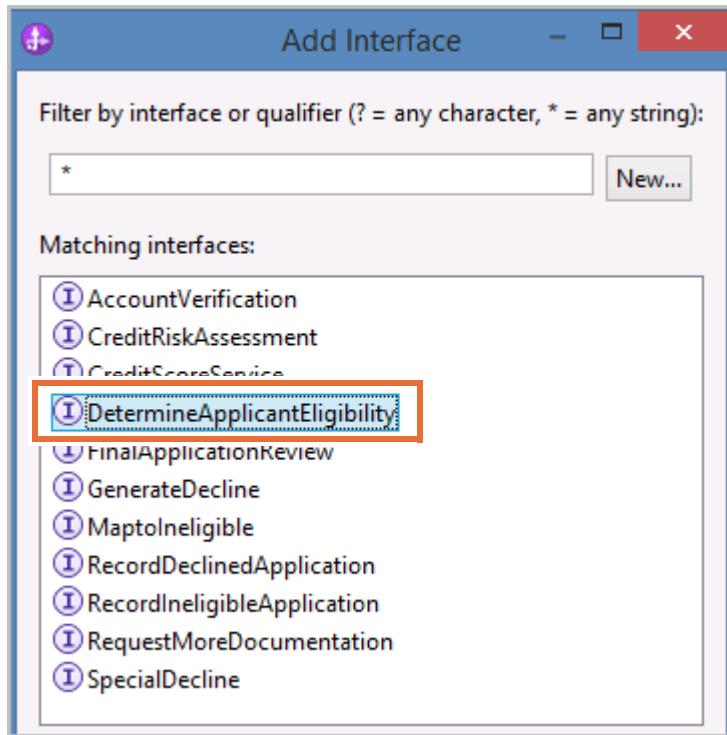


- \_\_\_ 7. Save your changes.
- \_\_\_ 8. Add a Java component that is named `DetermineApplicantEligibility` to the `FoundationModule` assembly diagram.
- In the assembly editor palette, expand **Components** and select **Java**.
  - Click any blank space on the diagram to add the component.
  - While the component is selected, switch to the **Description** tab in the **Properties** view.
  - Change the **Name** to: `DetermineApplicantEligibility`



- \_\_\_ 9. Add the `DetermineApplicantEligibility` interface to the component.
- Select the `DetermineApplicantEligibility` component, and click the **Add Interface** icon in the action bar.

- \_\_ b. In the Add Interface dialog box, select the **DetermineApplicantEligibility** interface.



- \_\_ c. Click **OK**.
10. Use the snippet code in `C:\labfiles\Support Files\Ex8\` `DetermineApplicantEligibilityImpl_InputCriterion.txt` to generate the implementation for the **DetermineApplicantEligibility** component. The snippet code returns values for each of the four test cases: IBM, ACME, TestCo, and AbcCo. For IBM, TestCo, and ACME, `eligibleApplication` is true by default. For AbcCo, `eligibleApplication` is false.
- \_\_ a. Right-click the **DetermineApplicantEligibility** component and click **Generate Implementation** from the menu.
  - \_\_ b. In the Generate Implementation dialog box, select **com.education** and click **OK** to open `DetermineApplicantEligibilityImpl.java` in the Java editor.
  - \_\_ c. Click the plus symbol to expand the `import` section and add the following two `import` statements:

```
import java.text.DateFormat;
import java.util.Date;

package com.education;

import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import java.text.DateFormat;
import java.util.Date;
```

- \_\_ d. In `DetermineApplicantEligibilityImpl.java`, browse to the `public DataObject InputCriterion` declaration at the end of the file.
- \_\_ e. In Windows Explorer, browse to `C:\labfiles\Support Files\Ex8\` and open `DetermineApplicantEligibility_InputCriterion.txt` in a text editor.
- \_\_ f. Copy the text in `DetermineApplicantEligibility_InputCriterion.txt` and paste it over the green comments section (the text that begins with `//`) in the `InputCriterion` method.



**Important**

Remove `return null;` from the end of the `InputCriterion` method.

- \_\_\_ g. Alternatively, enter the following code for the `InputCriterion` method:

```

System.out.println("[Java] Determine Applicant Eligibility - begins");
String date =
DateFormat.getDateInstance(DateFormat.MEDIUM).format(new Date());

if (credappin.getString("companyName").equals("ACME")) {
 credappin.setString("applicationDate", date);
 credappin.setString("companyName", "ACME");
 credappin.setString("customerCity", "Berlin");
 credappin.setString("customerCountry", "Germany");
 credappin.setString("contactFirstName", "Torsten");
 credappin.setString("contactLastName", "Frings");
 credappin.setString("contactPhoneNumber", "905-555-7234");
 credappin.setInt("requestAccountAmount", 10000);
 credappin.setInt("creditScore", 0);
 credappin.setString("productName", "Labels");
 credappin.setInt("pricingCode", 23);
 credappin.setString("comments", "None");
 credappin.setString("creditRating", "C+");
 credappin.setString("pricingScore", "17");
 credappin.setBoolean("creditReportNeeded", true);
 credappin.setString("creditRisk", "MED");
 credappin.setBoolean("applicationDecision", true);
 credappin.setBoolean("eligibleApplication", true);
 credappin.setString("accountNumber", "ACM002");
} else if (credappin.getString("companyName").equals("AbcCo")) {
 credappin.setString("applicationDate", date);
 credappin.setString("companyName", "AbcCo");
 credappin.setString("customerCity", "Madrid");
 credappin.setString("customerCountry", "Spain");
 credappin.setString("contactFirstName", "Fernando");
 credappin.setString("contactLastName", "Torres");
 credappin.setString("contactPhoneNumber", "312-555-9725");
 credappin.setInt("requestAccountAmount", 20000);
 credappin.setInt("creditScore", 0);
 credappin.setString("productName", "Tacks");
 credappin.setInt("pricingCode", 51);
 credappin.setString("comments", "Bad credit");
 credappin.setString("creditRating", "F");
 credappin.setString("pricingScore", "31");
 credappin.setBoolean("creditReportNeeded", true);
 credappin.setString("creditRisk", "HIGH");
 credappin.setBoolean("applicationDecision", false);
 credappin.setBoolean("eligibleApplication", false);
 credappin.setString("ineligibleReason", "Bad credit");
 credappin.setString("accountNumber", "ABC001");
}

```

```
 } else if (credappin.getString("companyName").equals("IBM")) {
 credappin.setString("applicationDate", date);
 credappin.setString("companyName", "IBM");
 credappin.setString("customerCity", "Boston");
 credappin.setString("customerCountry", "USA");
 credappin.setString("contactFirstName", "Landon");
 credappin.setString("contactLastName", "Donovan");
 credappin.setString("contactPhoneNumber", "547-555-3172");
 credappin.setInt("requestAccountAmount", 30000);
 credappin.setInt("creditScore", 0);
 credappin.setString("productName", "Pens");
 credappin.setInt("pricingCode", 34);
 credappin.setString("comments", "None");
 credappin.setString("creditRating", "A++");
 credappin.setString("pricingScore", "32");
 credappin.setBoolean("creditReportNeeded", true);
 credappin.setString("creditRisk", "LOW");
 credappin.setBoolean("applicationDecision", true);
 credappin.setBoolean("eligibleApplication", true);
 credappin.setString("accountNumber", "IBM007");
 } else if (credappin.getString("companyName").equals("TestCo")) {
 credappin.setString("applicationDate", date);
 credappin.setString("companyName", "TestCo");
 credappin.setString("customerCity", "Chicago");
 credappin.setString("customerCountry", "USA");
 credappin.setString("contactFirstName", "Jane");
 credappin.setString("contactLastName", "Doe");
 credappin.setString("contactPhoneNumber", "872-555-9915");
 credappin.setInt("requestAccountAmount", 50000);
 credappin.setInt("creditScore", 0);
 credappin.setString("productName", "Chairs");
 credappin.setInt("pricingCode", 57);
 credappin.setString("comments", "None");
 credappin.setString("creditRating", "C");
 credappin.setString("pricingScore", "11");
 credappin.setBoolean("creditReportNeeded", true);
 credappin.setString("creditRisk", "HIGH");
 credappin.setBoolean("applicationDecision", false);
 credappin.setBoolean("eligibleApplication", true);
 credappin.setString("accountNumber", "TEST001");
 }
 System.out.println("[Java] Determine Applicant Eligibility -
ends");
 return credappin;
```



### Note

`DetermineApplicantEligibilityImpl.java` sets the CustomerApplication data based on the value in the `companyName` field. If the `companyName` is `ACME`, the output CustomerApplication data has the following values:

- `applicationDate`: date
- `companyName`: ACME
- `customerCity`: Berlin
- `customerCountry`: Germany
- `contactFirstName`: Torsten
- `contactLastName`: Frings
- `contactPhoneNumber`: 905-555-7234
- `requestAccountAmount`: 10000
- `creditScore`: 0
- `productName`: Labels
- `pricingCode`: 23
- `comments`: None
- `creditRating`: C+
- `pricingScore`: 17
- `creditReportNeeded`: true
- `creditRisk`: MED
- `applicationDecision`: true
- `eligibleApplication`: true
- `accountNumber`: ACM002

If the `companyName` is `AbcCo`, the output CustomerApplication data has the following values:

- `applicationDate`: date
- `companyName`: AbcCo
- `customerCity`: Madrid
- `customerCountry`: Spain
- `contactFirstName`: Fernando
- `contactLastName`: Torres
- `contactPhoneNumber`: 312-555-9725
- `requestAccountAmount`: 20000
- `creditScore`: 0
- `productName`: Tacks
- `pricingCode`: 51
- `comments`: Bad credit
- `creditRating`: F
- `pricingScore`: 31
- `creditReportNeeded`: true
- `creditRisk`: HIGH
- `applicationDecision`: false
- `eligibleApplication`: false (Note: the value is false for AbcCo.)
- `ineligibleReason`: Bad credit
- `accountNumber`: ABC001

If the companyName is IBM, the output CustomerApplication data has the following values:

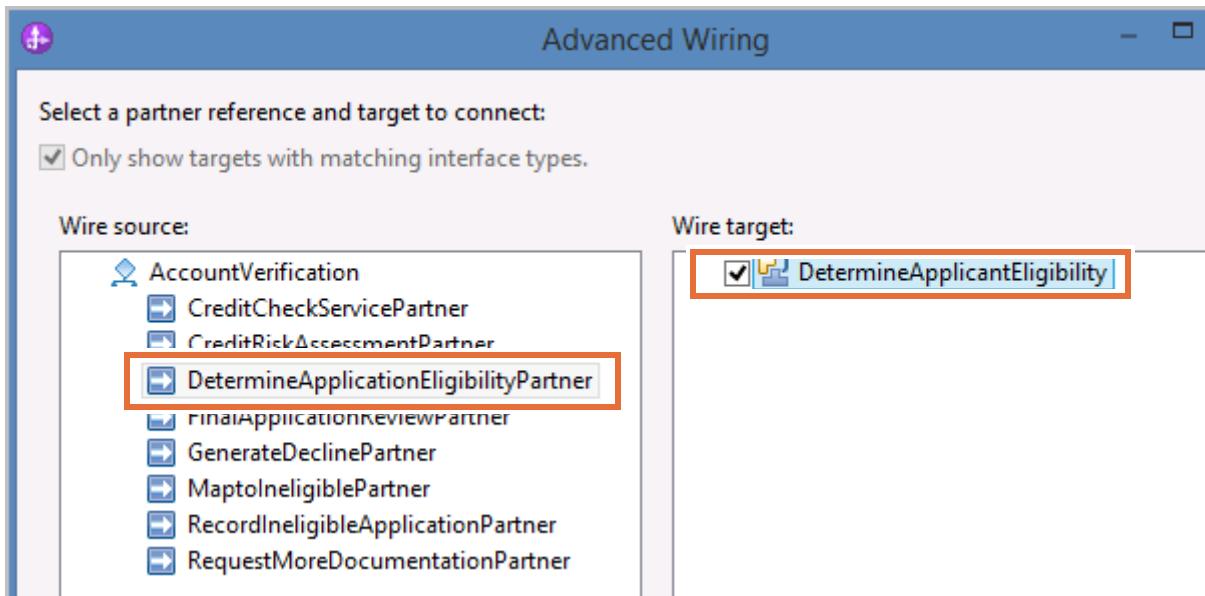
- applicationDate: date
- companyName: IBM
- customerCity: Boston
- customerCountry: USA
- contactFirstName: Landon
- contactLastName: Donovan
- contactPhoneNumber: 547-555-3172
- requestAccountAmount: 30000
- creditScore: 0
- productName: Pens
- pricingCode: 34
- comments: None
- creditRating: A++
- pricingScore: 32
- creditReportNeeded: true
- creditRisk: LOW
- applicationDecision: true
- eligibleApplication: true
- accountNumber: IBM007

If the companyName is TestCo, then the output CustomerApplication data has the following values:

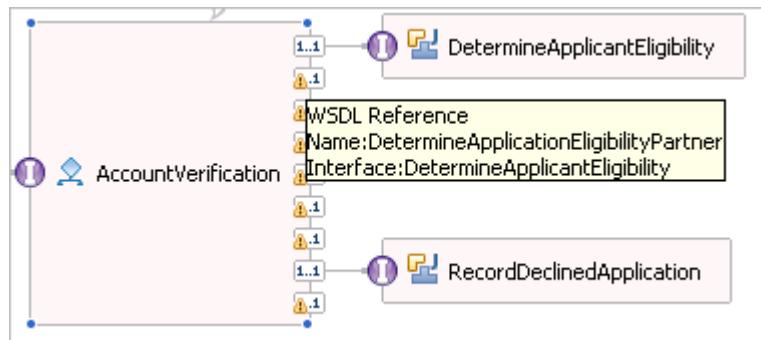
- applicationDate: date
- companyName: TestCo
- customerCity: Chicago
- customerCountry: USA
- contactFirstName: Jane
- contactLastName: Doe
- contactPhoneNumber: 872-555-9915
- requestAccountAmount: 50000
- creditScore: 0
- productName: Chairs
- pricingCode: 57
- comments: None
- creditRating: C
- pricingScore: 11
- creditReportNeeded: true
- creditRisk: HIGH
- applicationDecision: false
- eligibleApplication: true
- accountNumber: TEST001

- 
- \_\_ h. Press Ctrl+S to save your changes and Ctrl + W to close the Java editor.
  - \_\_ i. Close DetermineApplicantEligibility\_InputCriterion.txt and close Windows Explorer.

11. Wire the DetermineApplicantEligibility Java component to DetermineApplicationEligibilityPartner of the AccountVerification process.
- Right-click **AccountVerification** and select **Wire (Advanced)** from the menu.
  - In the “Wire source” window of the Advanced Wiring dialog box, select **DetermineApplicationEligibilityPartner**.
  - In the “Wire target” window, select the **DetermineApplicantEligibility** Java component.

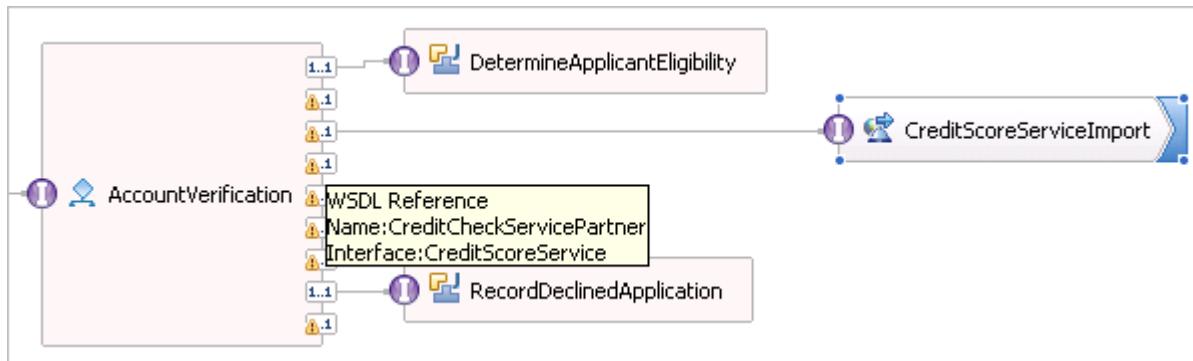


- Click **OK**.
- Verify the wiring by hovering the mouse over the reference. The dialog box indicates that the component is wired to the **DetermineApplicationEligibilityPartner** reference.

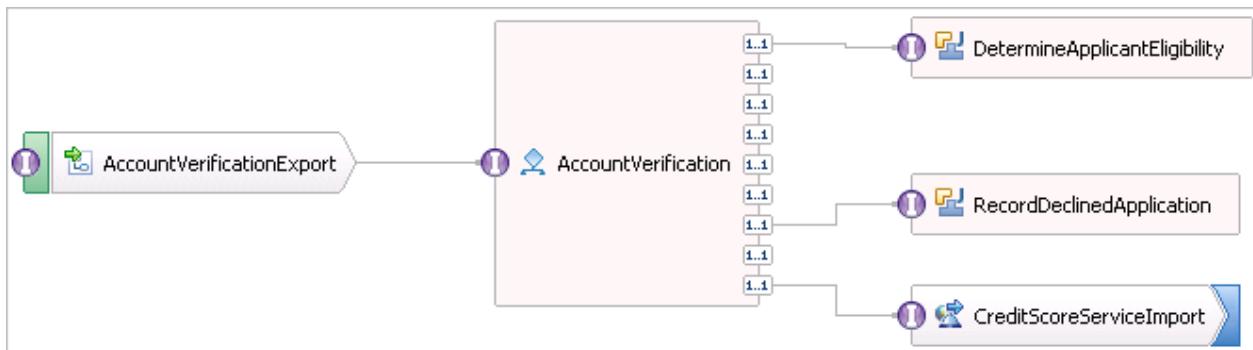


When you add the AccountVerification component to the FoundationModule assembly diagram, the order of the attached reference partners might differ from the diagram pictured. This difference does not indicate a problem. The order of the reference partners on the assembly diagram component does not reflect the order in which the process calls the reference partners.

- \_\_\_ 12. Save your changes.
- \_\_\_ 13. Wire the CreditScoreServiceImport component to the CreditCheckServicePartner reference of the AccountVerification process.
  - \_\_\_ a. Right-click **CreditScoreServiceImport** and click **Wire to Existing** from the menu.
  - \_\_\_ b. Verify the wiring by hovering the mouse over the reference. The dialog box indicates that the component is wired to the **CreditCheckServicePartner** reference.



- \_\_\_ c. Save the assembly diagram. It resembles the following figure:



#### **Part 4: Test a business process in the IBM Integration Designer test environment**

In this portion of the exercise, you test the FoundationModule (containing your business process) in the integrated test client. Services that are not yet implemented are emulated.

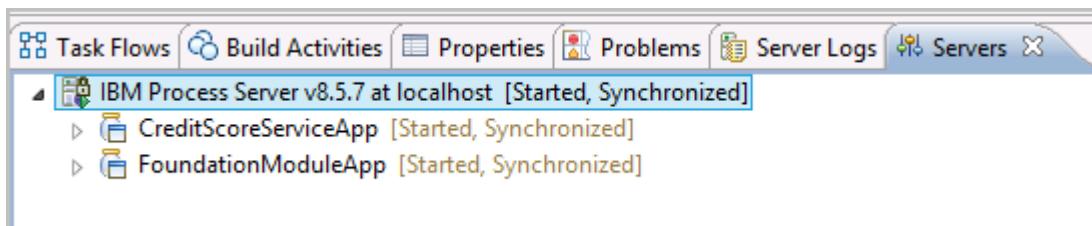
To test the application:

- \_\_\_ 1. Start the server (if it is not already running) and deploy **FoundationModuleApp** and **CreditScoreServiceApp** to the server.
  - \_\_\_ a. If necessary, in the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost**, and click **Start**. Wait for the startup process to complete before continuing.  
The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
  - \_\_\_ b. When the server is started successfully, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.

- \_\_\_ c. Click **Add All** and click **Finish**.
- \_\_\_ d. Wait until the modules are published and started. Publishing is complete when no messages appear in the IBM Integration Designer status bar such as **Publishing FoundationModuleApp**.

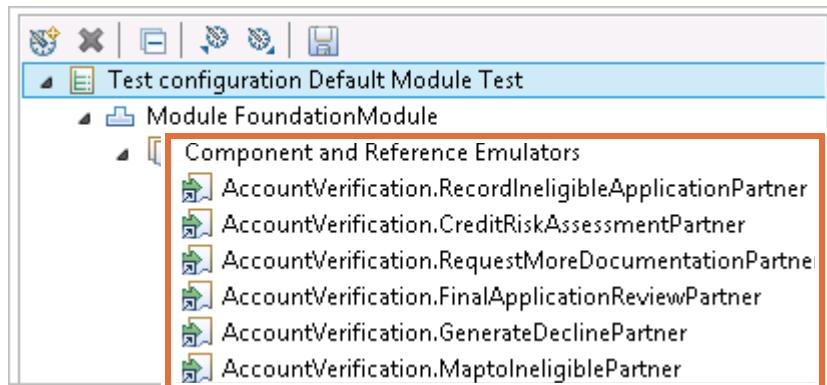
The applications are started when you see the messages **Application started: FoundationModuleApp** and **Application started: CreditScoreServiceApp** in the **Server Logs** view.

You can also verify the status of the modules by expanding the server.



If any module has a **Stopped** status, then right-click the module and click **Restart** from the menu. If prompted to do so, republish the module. Wait for the server status to change to **Started, Synchronized**. If the server has a status of **Started, Publishing**, then clicking the server refreshes the status to **Started, Synchronized**. Continue to the next step when the status is changed to **Started**.

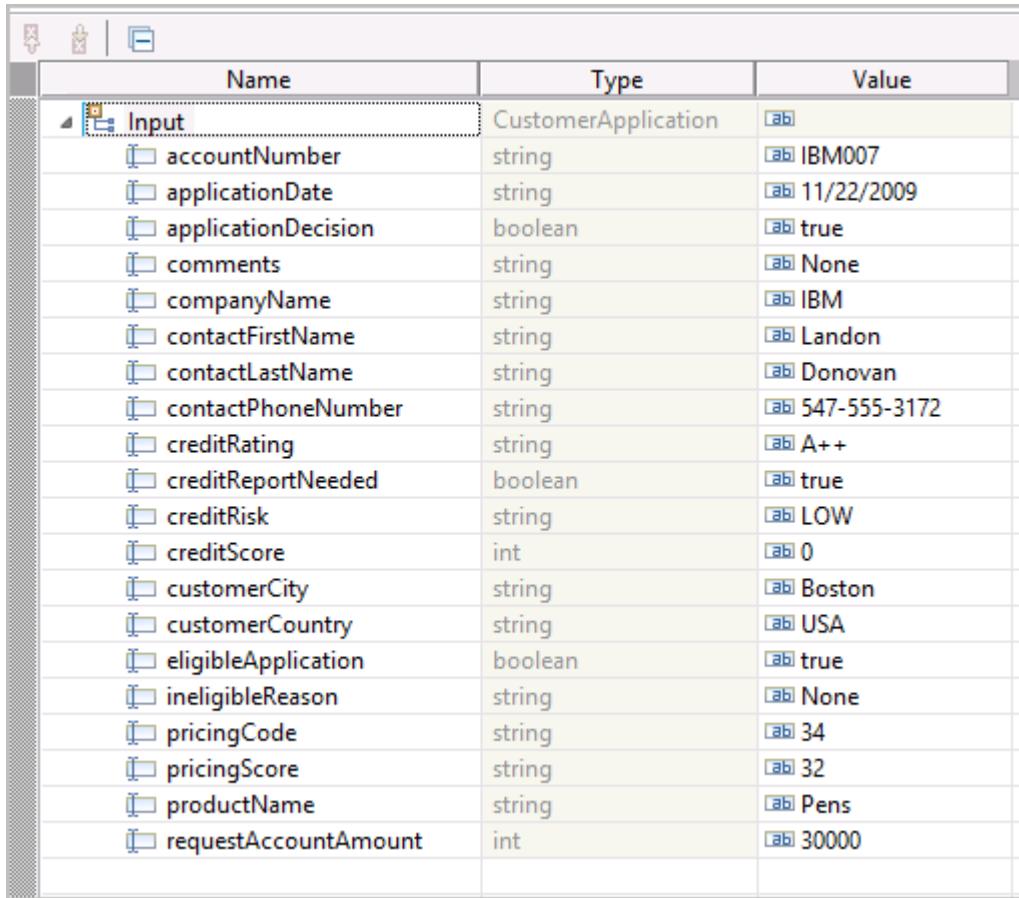
- \_\_\_ 2. Use the **Test Component** option to test the **AccountVerification** process.
- \_\_\_ a. In the assembly diagram, right-click **AccountVerification**, and click **Test Component** from the menu.
- \_\_\_ b. When the integrated test client opens, switch to the **Configurations** tab. The unwired references of the business processes are emulated.



- \_\_\_ c. Switch to the **Events** tab.
- \_\_\_ d. In the **Initial request parameters** section, right-click **Input** and click **Import from File** from the menu.
- \_\_\_ e. In the "Import from File" dialog box, browse to `C:\labfiles\Support Files\Ex8\`.
- \_\_\_ f. Select **EX8\_Test\_Data.xml** and click **Open** to populate the input parameters with the required test data.

\_\_ g. Alternatively, you can manually enter the following values:

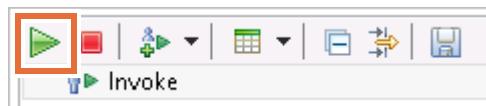
- accountNumber: IBM007
- applicationDate: 6/20/2014
- applicationDecision: true
- comments: None
- companyName: IBM
- contactFirstName: Landon
- contactLastName: Donovan
- contactPhoneNumber: 547-555-3172
- creditRating: A++
- creditReportNeeded: true
- creditRisk: LOW
- creditScore: 0
- customerCity: Boston
- customerCountry: USA
- eligibleApplication: true
- ineligibleReason: None
- pricingCode: 34
- pricingScore: 32
- productName: Pens
- requestAccountAmount: 30000



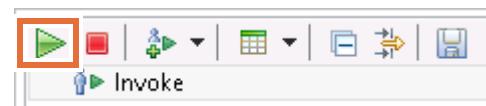
The screenshot shows a table titled "Input" which lists various parameters and their values. The columns are "Name", "Type", and "Value".

| Name                 | Type                | Value        |
|----------------------|---------------------|--------------|
| Input                | CustomerApplication |              |
| accountNumber        | string              | IBM007       |
| applicationDate      | string              | 11/22/2009   |
| applicationDecision  | boolean             | true         |
| comments             | string              | None         |
| companyName          | string              | IBM          |
| contactFirstName     | string              | Landon       |
| contactLastName      | string              | Donovan      |
| contactPhoneNumber   | string              | 547-555-3172 |
| creditRating         | string              | A++          |
| creditReportNeeded   | boolean             | true         |
| creditRisk           | string              | LOW          |
| creditScore          | int                 | 0            |
| customerCity         | string              | Boston       |
| customerCountry      | string              | USA          |
| eligibleApplication  | boolean             | true         |
| ineligibleReason     | string              | None         |
| pricingCode          | string              | 34           |
| pricingScore         | string              | 32           |
| productName          | string              | Pens         |
| requestAccountAmount | int                 | 30000        |

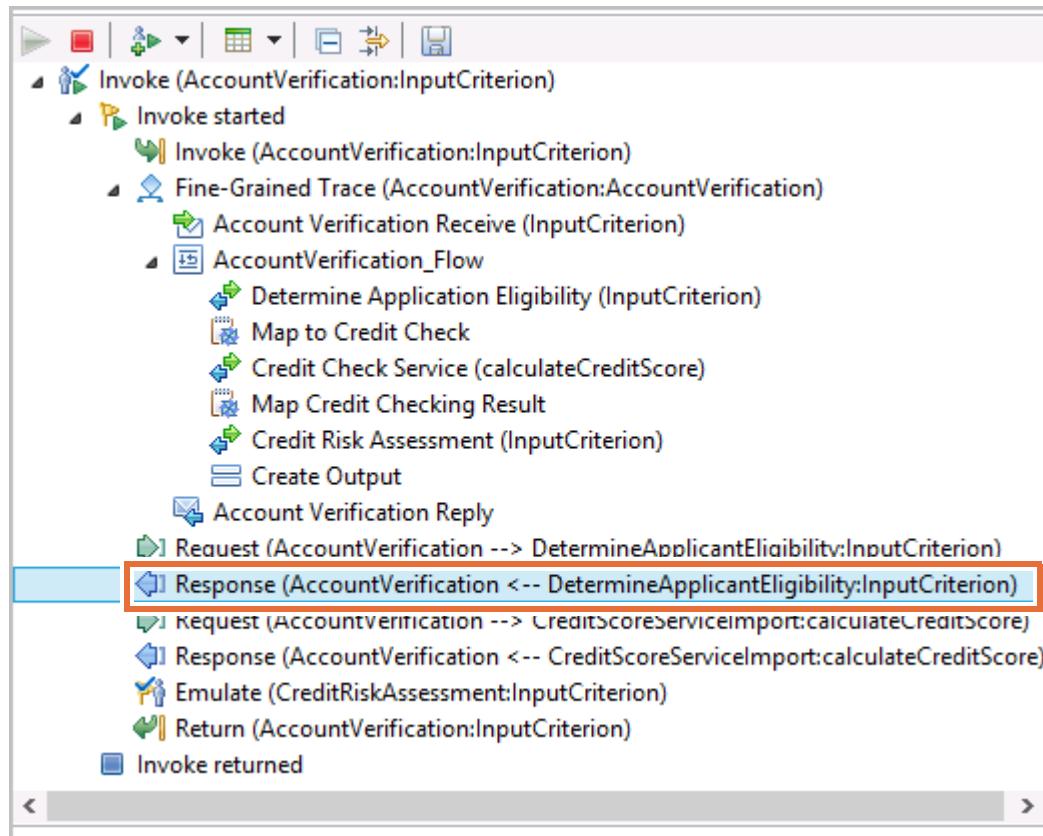
- \_\_ h. Click **Continue** on the Events toolbar to run the test.



- \_\_ i. For the deployment location, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_ j. At the User Login dialog box, accept the default values for **User ID** and **Password** and click **OK**.
- \_\_ k. The test pauses at **Emulate (CreditRiskAssessment:InputCriterion)**; click **Continue** in the Events toolbar to resume the test.

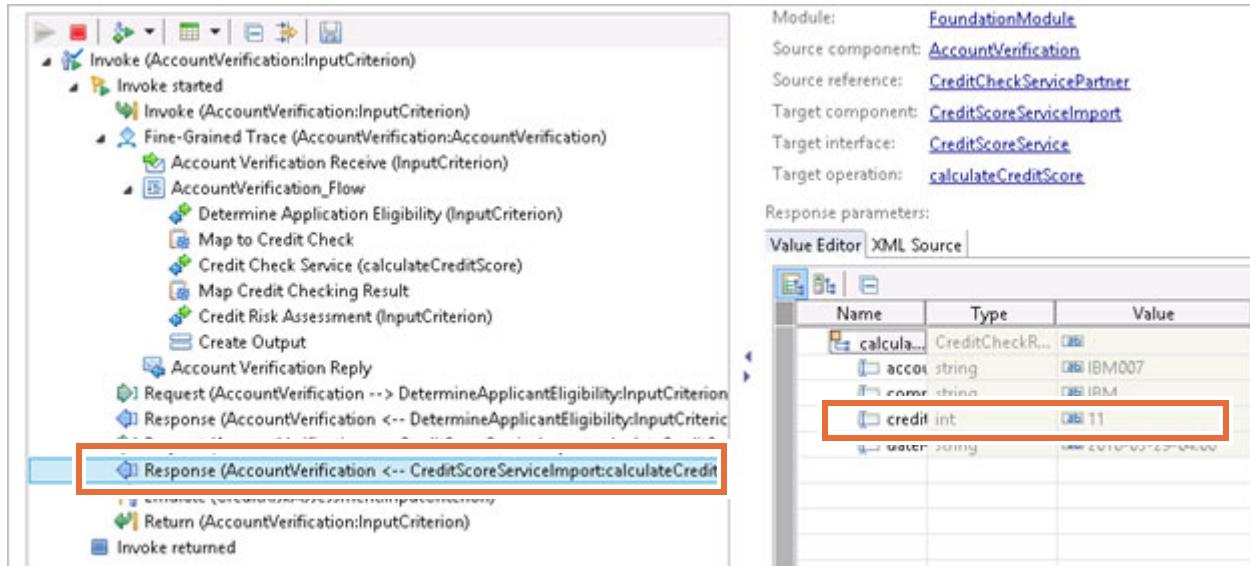


- \_\_ l. Notice the first response that is returned from the **DetermineApplicantEligibility** service.

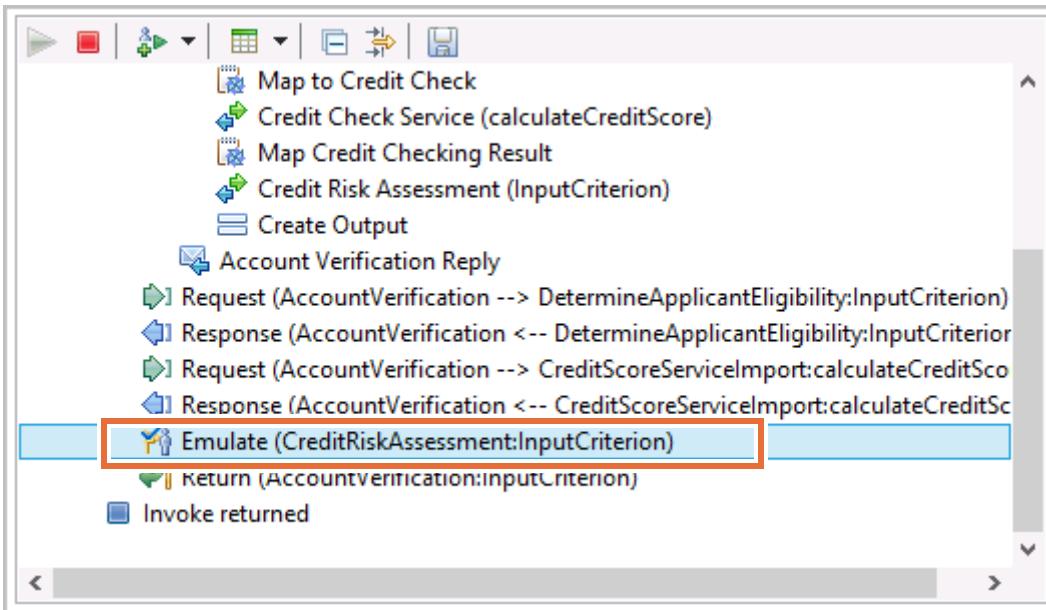


Based on the response from the **DetermineApplicantEligibility** service, the process takes either the eligible application path or the ineligible application path. In three of the four test cases (**IBM**, **TestCo**, and **ACME**), **eligibleApplication** is set to **true**. If the **companyName** is set to **AbcCo**, the **DetermineApplicantEligibility** service returns an **eligibleApplication** value of **false**. Because your test data uses **IBM** for the **companyName**, the eligible application path is followed.

- \_\_\_ m. Notice the second response that is from the CreditScoreService. If the application is eligible, the data is transformed and sent to the CreditScoreService. Because your test data was for IBM, the web service returns a credit score of 11. This score is a high credit score and represents a low credit risk. Select the **Response (AccountVerification <- CreditScoreServiceImport:calculateCreditScore)** event and examine the data in the **Response parameters** section. A creditScore of 11 is returned.



- \_\_\_ n. The test stops at the first service emulation. Because the CreditRiskAssessment service is not implemented, the service is emulated and requires you to manually send the response data back to the business process. Select **Emulate (CreditRiskAssessment:InputCriterion)**.

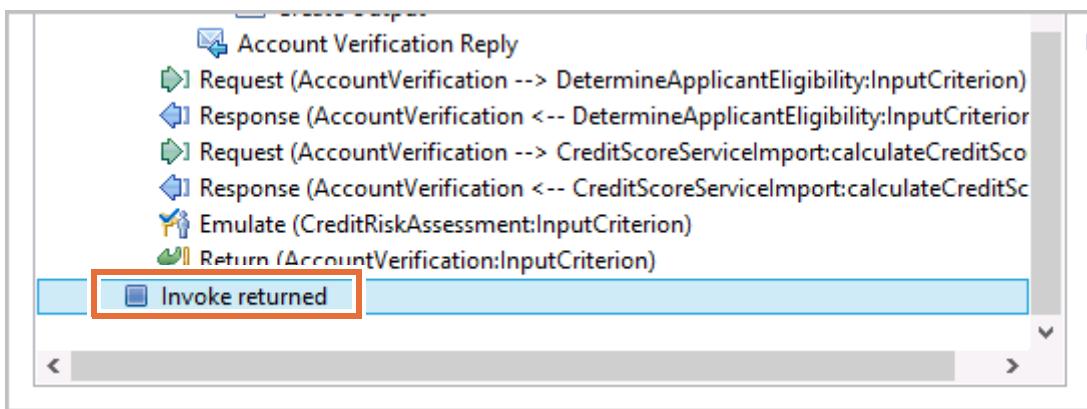


- \_\_\_ o. In the **Output parameters** section, scroll to the **creditRisk** attribute.

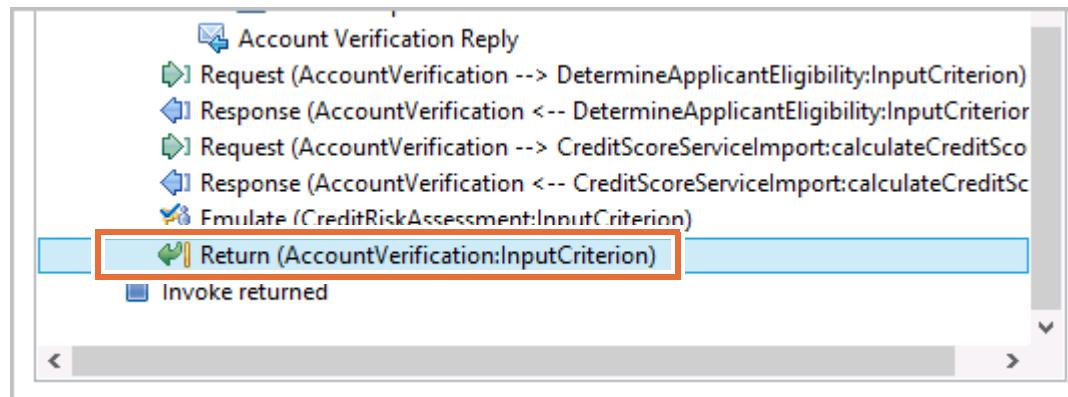
- \_\_ p. Verify that **LOW** is displayed in the **Value** column.

| Name               | Type    | Value        |
|--------------------|---------|--------------|
| contactLastName    | string  | Donovan      |
| contactPhoneNum    | string  | 547-555-3172 |
| creditRating       | string  | A++          |
| creditReportNeeded | boolean | true         |
| <b>creditRisk</b>  | string  | <b>LOW</b>   |
| creditScore        | int     | 111          |
| customerCity       | string  | Boston       |
| customerCountry    | string  | USA          |

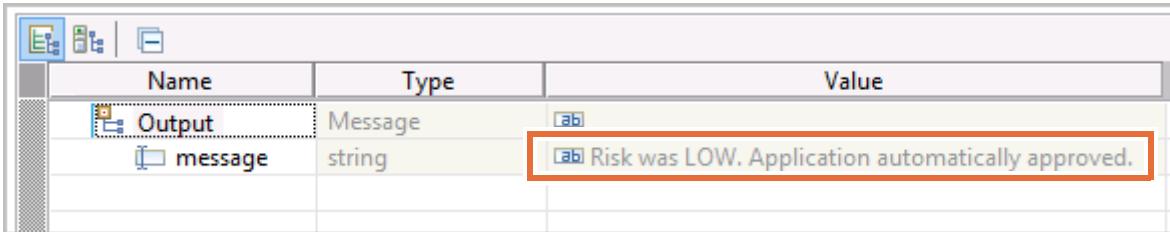
- \_\_ q. The blue, square **Invoke returned** stop node indicates completion of the test trace.



- \_\_ r. Because the **creditRisk** was **LOW**, the application was automatically approved. Select the **Return (AccountVerification:InputCriterion)** event.



- \_\_ s. Examine the output in the **Return parameters** section.



| Name    | Type    | Value                                                |
|---------|---------|------------------------------------------------------|
| Output  | Message | ab                                                   |
| message | string  | ab Risk was LOW. Application automatically approved. |

- \_\_ t. If you are prompted to save the test trace, close the **FoundationModule\_Test** tab and click **No**.



#### Note

In future exercises, as you implement the remaining services that your application uses, you use the prebuilt test cases to test other paths through the business process.

- \_\_ 3. Remove the projects and (optionally) stop the server.
- \_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.
  - \_\_ b. Click **Remove All** and click **Finish**.
  - \_\_ c. (Optional) Stop the server.
- \_\_ 4. Close IBM Integration Designer.

## End of exercise

## Exercise review and wrap-up

In this exercise, you completed a complex business process and tested it in the IBM Integration Designer test environment.



# Exercise 9. Creating business rules

## What this exercise is about

In this exercise, you create rule sets and decision tables that contain business rules. You create a rule group component, incorporate the rule group component in an assembly diagram, and test it. Finally, you use the Business Rules Manager web client to interact with business rules at run time.

## What you should be able to do

After completing this exercise, you should be able to:

- Create rule sets and decision tables that contain business rules
- Create a rule group component
- Incorporate a rule group component in an assembly diagram
- Test a business rule group in the integrated test client
- Use the Business Rule Manager web client to interact with business rules at run time

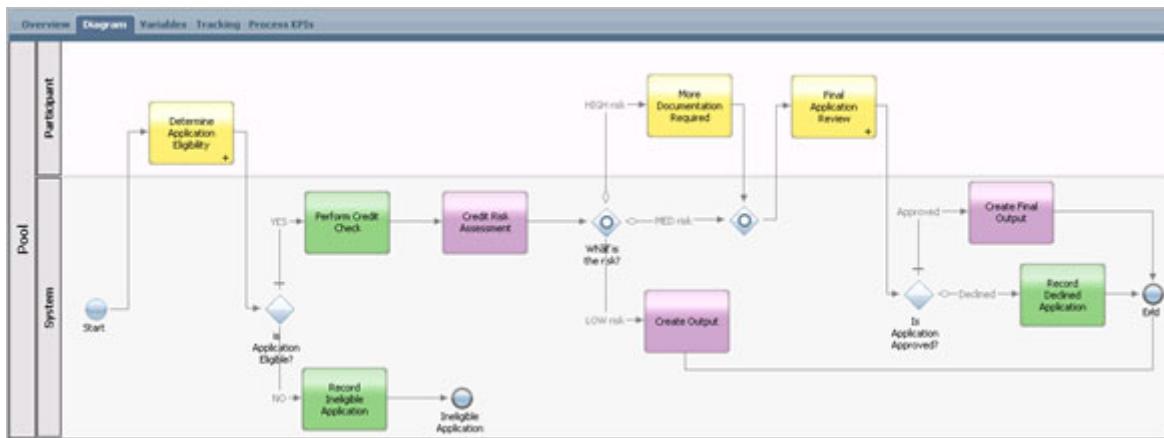
## Introduction

A business rule captures and implements business policies and practices. A rule can enforce business policy, incorporate a decision, or infer new data from existing data.

Rules capture decision-making logic in the form of if-then statements. These if-then statements are grouped into rule sets and decision tables. A rule set captures decision-making business logic in the form of a series of if-then statements. A decision table captures multi-conditional decision-making business logic in tabular format. Rule sets and decision tables are incorporated into rule groups. Rule groups are exposed as service components and wired to other components in assembly diagrams.

See the process application model that was created in a previous exercise. In this lab, you implement the Credit Risk Assessment activity that is shown in the model. This activity is implemented as a business rule group.

Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

## Exercise instructions

In this exercise, you implement the service that determines a customer's credit risk. To determine the credit risk, a series of business rules are used to evaluate the credit score that the credit score service returns. For educational purposes, you implement the rule logic in both a decision table and a rule set. However, only the rule set is used in your application.

### **Part 1: Create a rule group component that contains business rules in rule sets and decision tables**

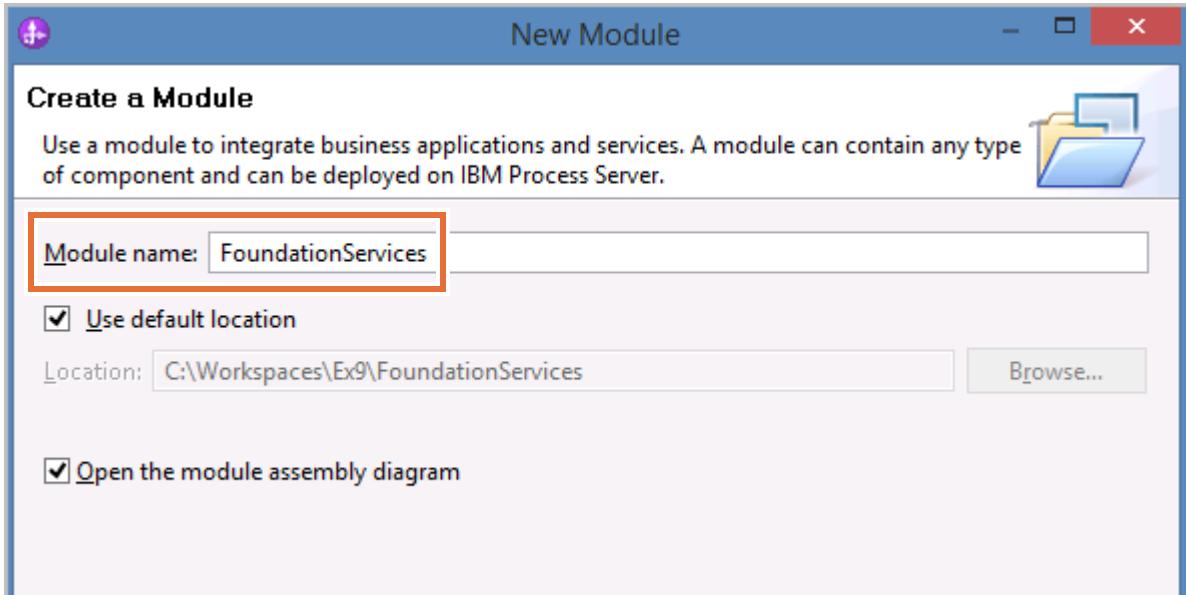
In this portion of the exercise, you create the business rules that the credit risk assessment service uses. If the creditScore value is less than 4, then the creditRisk is `HIGH`. If the creditScore value is in the range of 4 – 7, then the creditRisk is `MED` (short for medium). If the creditScore value is in the range of 8 – 11, then the creditRisk is `LOW`.

Customer applications that are `HIGH` risk require more documentation and are subject to more review. Customer applications that are `MED` risk require more review but not more documentation. Customer applications that are `LOW` risk are automatically approved.

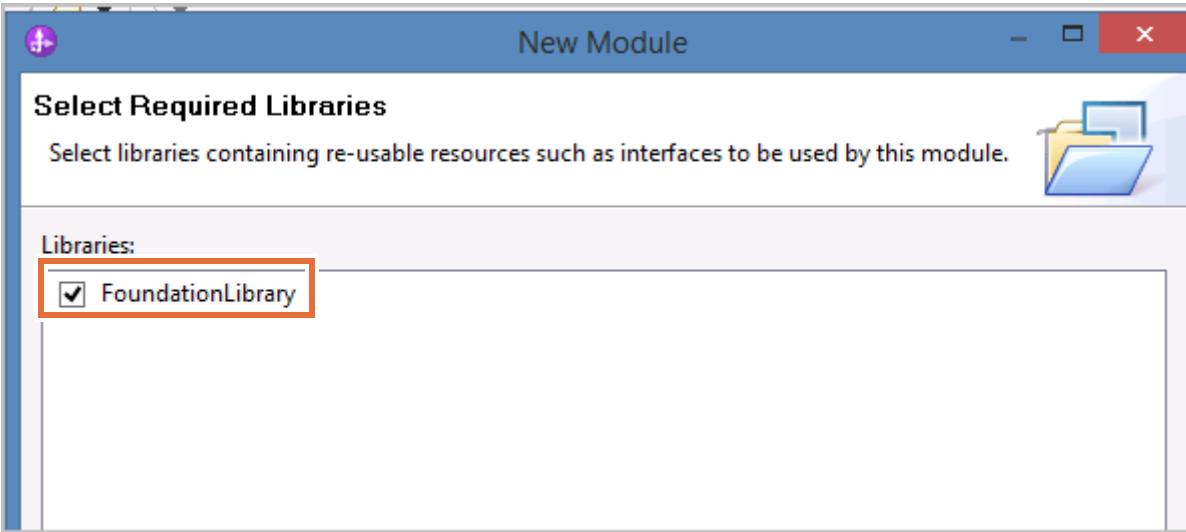
To create the risk assessment business rules:

- 1. Open the Exercise 9 workspace.
  - a. On your desktop, open the **Exercise Shortcuts** folder.
  - b. Double-click the **Exercise 9** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
  - d. Close the **Getting Started** tab.
- 2. Create a module that is named `FoundationServices` to store the rule group. This module is used in later exercises to hold other service implementations that the `AccountVerification` business process invokes.
  - a. Click **File > New > Module** from the menu options.

- \_\_ b. Type FoundationServices in the **Module name** field.



- \_\_ c. Click **Next**.  
\_\_ d. In the Select Required Libraries window, select **FoundationLibrary**.



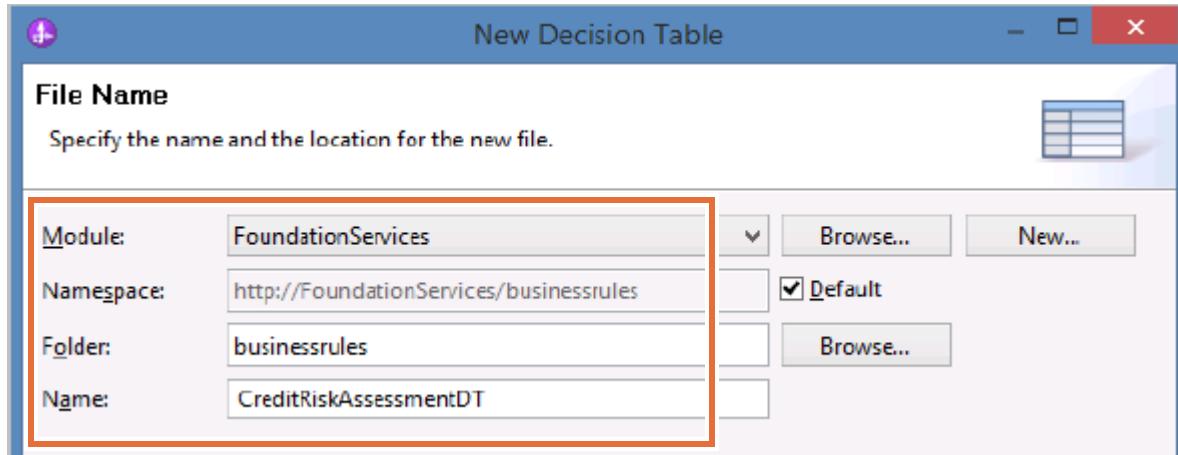
- \_\_ e. Click **Finish**.

## Create the decision table for the rule group

- \_\_ 1. Create a CreditRiskAssessment business rule group in the FoundationServices module in the `businessrules/` folder that uses the CreditRiskAssessment interface. Create a CreditRiskAssessmentDT decision table in the rule group.
  - \_\_ a. Right-click the **FoundationServices** module and click **New > Rules > Decision Table** from the menu.

- \_\_\_ b. In the **New Decision Table** dialog box, enter the following information.

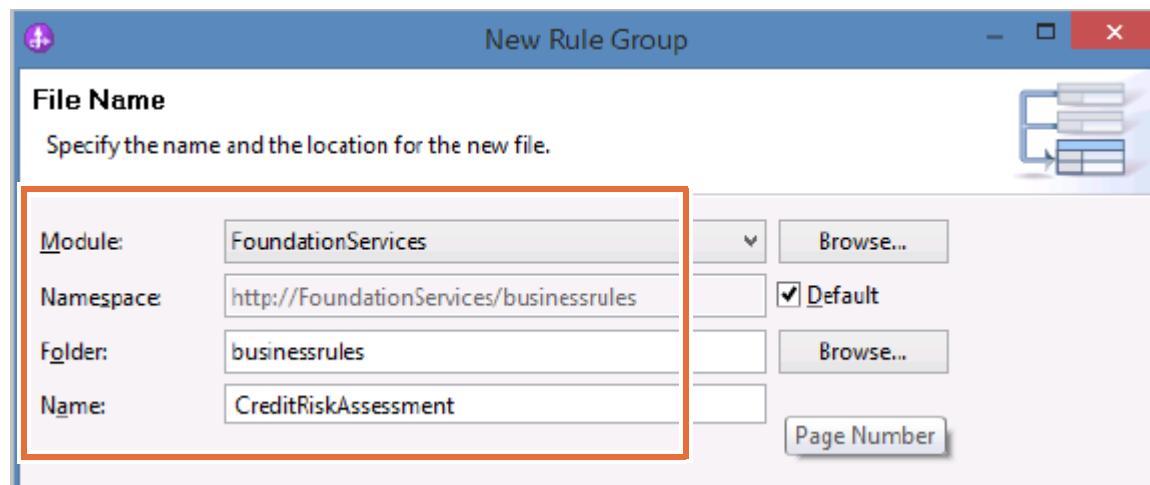
- Click **Browse** beside the **Folder** field, click **New Folder**, type `businessrules/` in the **Folder Name** field, and click **OK** twice. When you are returned to the **File Name** window, the slash character is removed from the folder name.
- Type `CreditRiskAssessmentDT` in the **Name** field.



- \_\_\_ c. Accept the remaining default options and click **Next**.

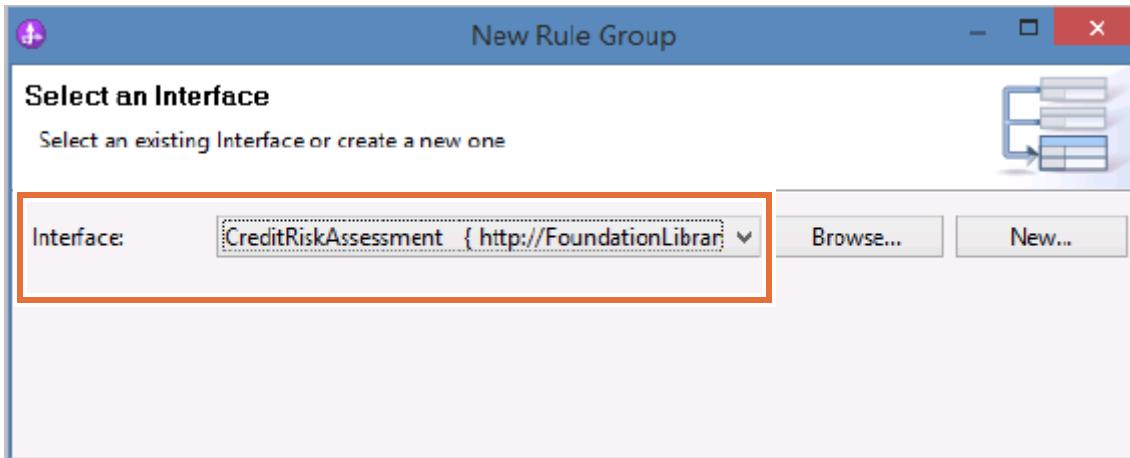
- \_\_\_ d. In the “Interface and Operation” window, follow these instructions:

- Click **New** beside the **Rule Group** field to open the **File Name** window.
- In the **File Name** window, click **Browse** beside the **Folder** field, select the **businessrules** folder from the list, and click **OK**.
- When you are returned to the **File Name** window, type `CreditRiskAssessment` in the **Name** field.

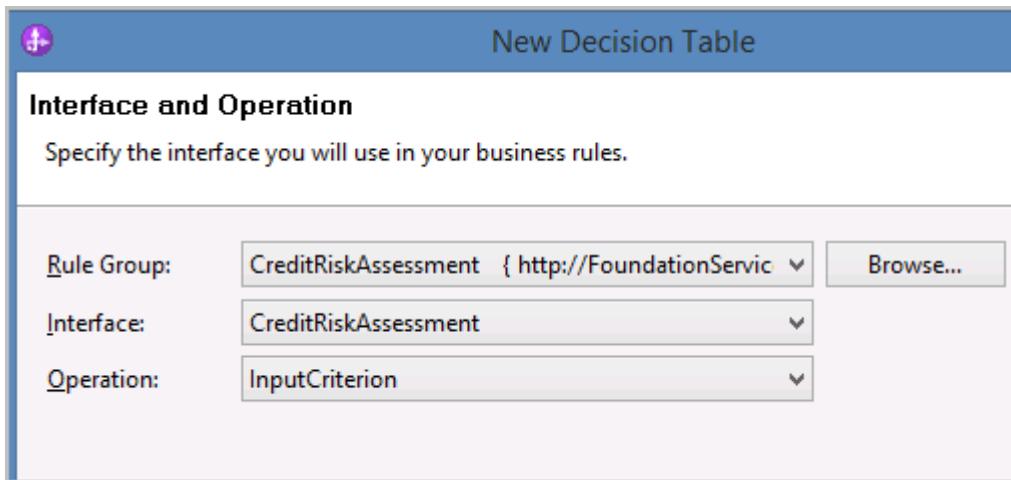


- \_\_\_ e. Accept the remaining default options and click **Next**.

- \_\_\_ f. In the “Select an Interface” window, select the **CreditRiskAssessment** interface.



- \_\_\_ g. Click **Finish**. When you are returned to the New Decision Table dialog box, on the “Interface and Operation” window, verify that **Rule Group** is set to CreditRiskAssessment, **Interface** is set to CreditRiskAssessment, and **Operation** is set to InputCriterion.



- \_\_\_ h. Click **Finish** to complete the decision table wizard.
- \_\_\_ 2. Create an initialization action rule that is named `MapInputToOutput` that copies the data from the Input business object and assigns it to the Output business object. The rule logic is implemented by using the following expression: `Output = copyBO(Input)`
- \_\_\_ a. In the **Initialize** section of the decision table editor, click the **Add an Initialization Action Rule** icon.

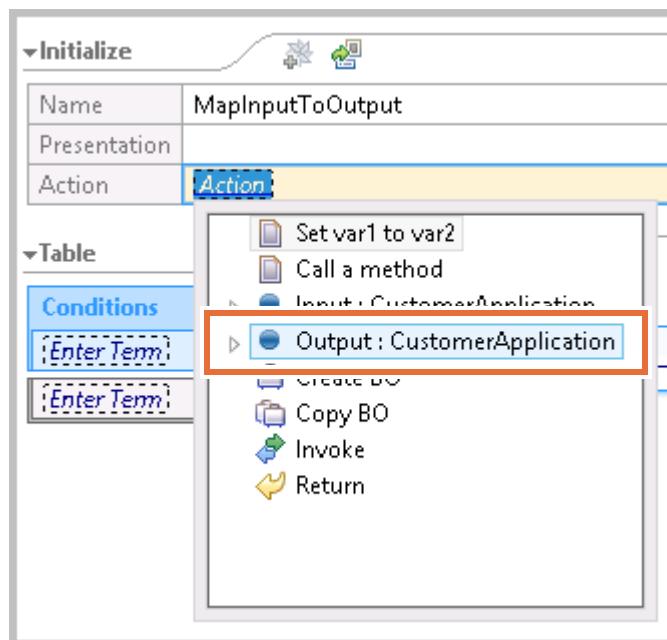


- \_\_\_ b. Change the action rule **Name** from `Rule1` to: `MapInputToOutput`

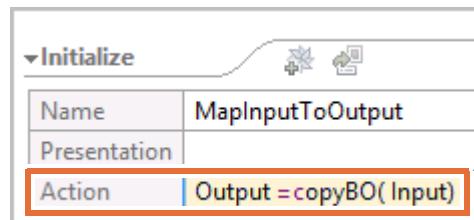
- \_\_\_ c. In the **Action** field, click the **Action** link to create an expression with the expression builder.



- \_\_\_ d. Select **Output : CustomerApplication** from the list.



- \_\_\_ e. Select the **=** operator (the first item in the list).  
 \_\_\_ f. Select **Copy BO**.  
 \_\_\_ g. Select **Input: CustomerApplication**.  
 \_\_\_ h. Click the **Action** field to close the expression builder. Your completed action rule resembles the following figure:



**Note**

You can type the entire expression manually instead of using the expression builder, but the expression builder helps eliminate the possibility of typographical errors.

- \_\_\_ 3. Save your changes. You can ignore any errors in the **Problems** view.
- \_\_\_ 4. Use the following business rules to implement the decision table.
  - If the **creditScore** value is less than 4, then the **creditRisk** is **HIGH**.
  - If the **creditScore** value is in the range of 4 – 7, then the **creditRisk** is **MED**.
  - If the **creditScore** value is in the range of 8 – 11, then the **creditRisk** is **LOW**.
  - Otherwise, set the **creditRisk** to **HIGH**.
- \_\_\_ a. In the **Table** section of the editor, click the **Enter Term** link in the first row of the **Conditions** column.
- \_\_\_ b. Using the expression builder, expand **Output : CustomerApplication** and select **creditScore** (click the word **Conditions** to close the expression builder). Alternatively, type: `Output.creditScore`

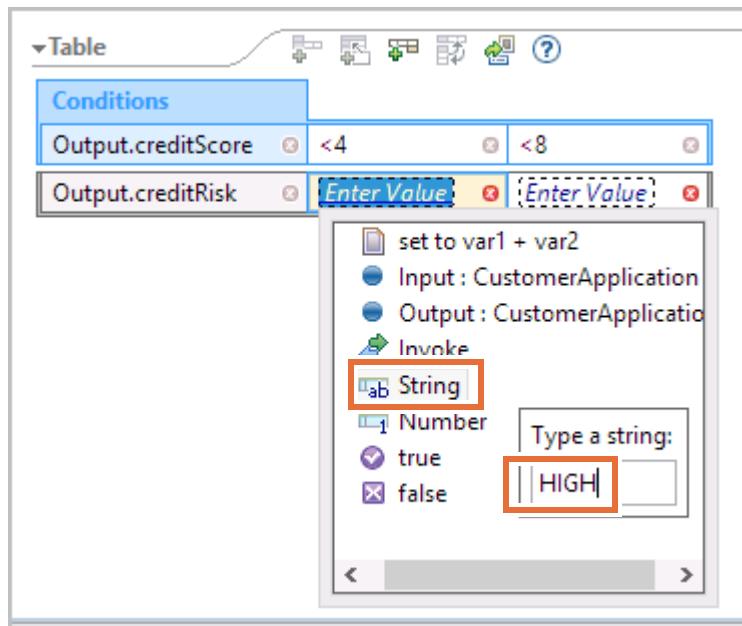
| Conditions                      |               |               |  |
|---------------------------------|---------------|---------------|--|
| <code>Output.creditScore</code> | (Enter Value) | (Enter Value) |  |
| [Enter Term]                    |               |               |  |
| Actions                         |               |               |  |

- \_\_\_ c. Click the **Enter Term** link in the second row of the **Conditions** column.
- \_\_\_ d. Using the expression builder, expand **Output : CustomerApplication** and select **creditRisk**. Alternatively, type: `Output.creditRisk`

| Conditions                      |               |               |  |
|---------------------------------|---------------|---------------|--|
| <code>Output.creditScore</code> | (Enter Value) | (Enter Value) |  |
| <code>Output.creditRisk</code>  | (Enter Value) | (Enter Value) |  |
| Actions                         |               |               |  |

- \_\_\_ e. Click the **Enter Value** link in the first row, second column, and type the expression: `<4`
- \_\_\_ f. Click the **Enter Value** link in the first row, third column, and type the expression: `<8`
- \_\_\_ g. Click the **Enter Value** link in the second row, second column.

- \_\_ h. Select **String**, type **HIGH** in the **Type a string** field, and press Enter. This rule sets **Output.creditRisk** to **HIGH** when the credit score is less than 4.

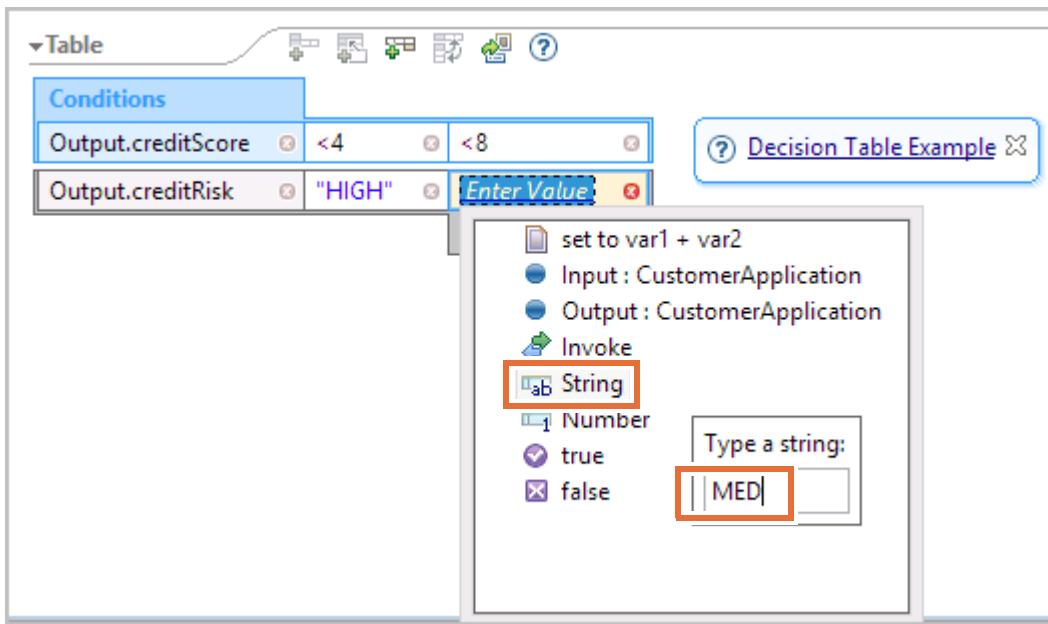


### Note

Though you entered **HIGH** in the **Type a string** field without quotation marks, the value is displayed in the decision table *with* quotation marks, which is normal for a string value.

- \_\_ i. Click the **Enter Value** link in the second row, third column.

- \_\_ j. Select **String**, type `MED` in the **Type a string** field, and press Enter. This rule sets `Output.creditRisk` to `MED` (short for medium) when the credit score is in the range of 4 – 7.



- \_\_ k. Click the **Add a New Condition Value** icon to add a column to the table. If the icon is not active, select the `<8` table cell first.



- \_\_ l. Click the **Enter Value** link in the first row, fourth column, and type the expression: `<12`  
\_\_ m. Click the **Enter Value** link in the second row, fourth column.

- \_\_\_ n. Select **String**, type **LOW** in the **Type a string** field, and press Enter. This rule sets **Output.creditRisk** to **LOW** when the credit score is in the range of 8 – 11.

| Conditions         |    |    |     | Action                      |
|--------------------|----|----|-----|-----------------------------|
| Output.creditScore | <4 | <8 | <12 | "HIGH"                      |
| Output.creditScore |    |    | <12 | "MED"                       |
|                    |    |    |     | <a href="#">Enter Value</a> |

Decision Table Example

String  
Type a string: LOW

- \_\_\_ o. Place your cursor in the **<12** cell (first row, fourth column), right-click the cell, and select **Add Condition Otherwise** from the menu.
- \_\_\_ p. Click the **Enter Value** link in the **Otherwise** column.
- \_\_\_ q. Select **String**, type **HIGH** in the **Type a string** field, and press Enter. This rule sets **Output.creditRisk** to **HIGH** if the credit score is outside the normal range of 1 to 11.

| Conditions         |        |       |       | Action                      |
|--------------------|--------|-------|-------|-----------------------------|
| Output.creditScore | <4     | <8    | <12   | Otherwise                   |
| Output.creditScore | "HIGH" | "MED" | "LOW" | <a href="#">Enter Value</a> |
|                    |        |       |       |                             |

Decision Table Example

String  
Type a string: HIGH

- \_\_\_ r. The completed **CreditRiskAssessmentDT** decision table resembles the following image.

The screenshot shows a decision table editor window titled "Table". The table has two rows of conditions and one row of actions. The first condition row has four columns: <4, <8, <12, and Otherwise. The second condition row has four columns: "HIGH", "MED", "LOW", and "HIGH". The action row contains a single column labeled "Actions".

| Conditions         |        |       |       |           |
|--------------------|--------|-------|-------|-----------|
| Output.creditScore | <4     | <8    | <12   | Otherwise |
| Output.creditRisk  | "HIGH" | "MED" | "LOW" | "HIGH"    |
| Actions            |        |       |       |           |

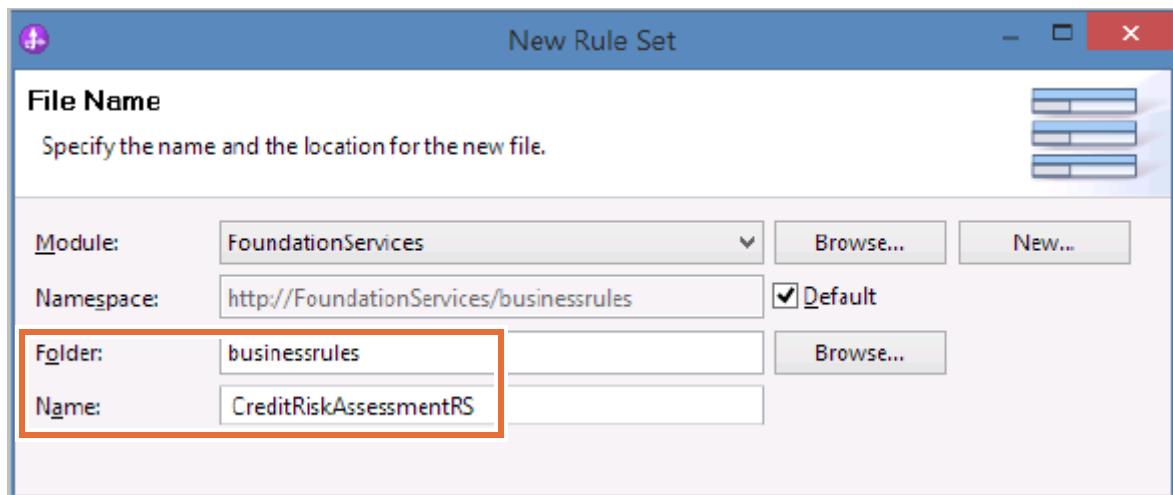
- \_\_\_ 5. Save your changes and close the decision table editor by closing the tab. Continue to ignore any errors in the **Problems** view.

## Create the rule set for the rule group

- \_\_\_ 1. Create a rule set that is named `CreditRiskAssessmentRS` in the `FoundationServices` module, in the `businessrules/` folder, that uses the `CreditRiskAssessment` interface.

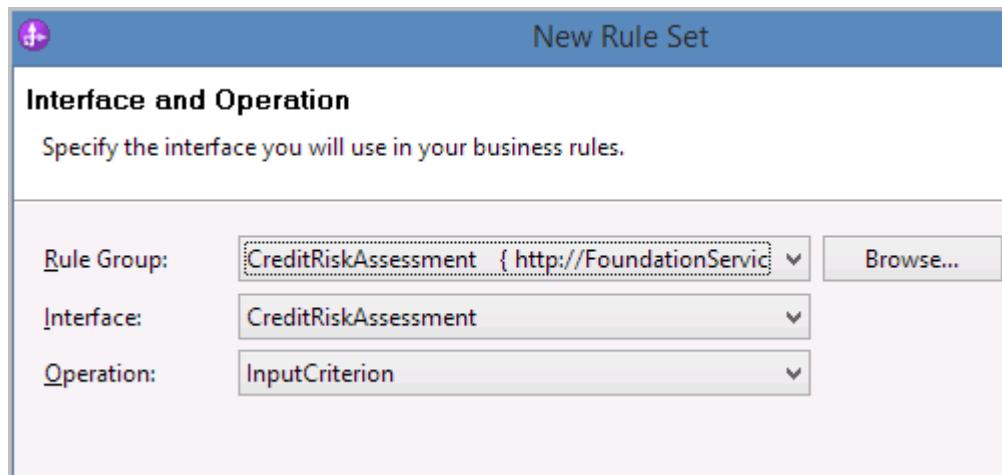
You are creating both a rule set and a decision table for educational purposes. Only the rule set is used in your application.

- \_\_\_ a. Expand **FoundationServices > Integration Logic**.
- \_\_\_ b. Right-click **Rule Logic** and click **New > Rule Set** from the menu.
- \_\_\_ c. In the File Name window, follow these instructions:
  - Click **Browse** beside the **Folder** field, select the **businessrules** folder, and click **OK**.
  - Type `CreditRiskAssessmentRS` in the **Name** field.



- \_\_\_ d. Accept the remaining default options and click **Next**.

- \_\_\_ e. In the “Interface and Operation” window, enter the following information.
- In the **Rule Group** field, select **CreditRiskAssessment**.
  - Verify that the **Interface** is set to `CreditRiskAssessment` and the **Operation** is set to `InputCriterion`.



- \_\_\_ f. Click **Finish**.
- \_\_\_ 2. Create an if-then rule template that is named `CreditRiskTemplate` with three parameters: `lowerLimit` (of type **int**), `upperLimit` (of type **int**), and `decision` (of type **string**). A rule set template defines the implementation and parameters for an if-then or action rule. When defined, the template can be used to create instances of the same rule by using different parameters.

The template logic is:

If all of the following are true

- `Output.creditScore > lowerLimit`
- `Output.creditScore < upperLimit`

Then `Output.creditRisk = decision`

The presentation of the template is:

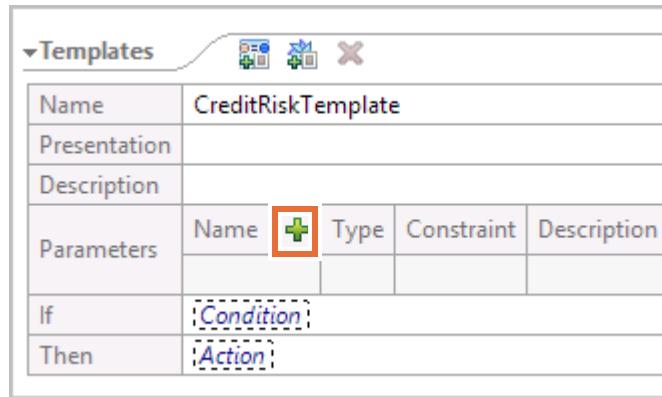
If the customer credit score is greater than `lowerLimit` and less than `upperLimit` then the credit risk is `decision`

- \_\_\_ a. In the **Templates** section of the rule set editor, click the **Add If-Then Template** icon.



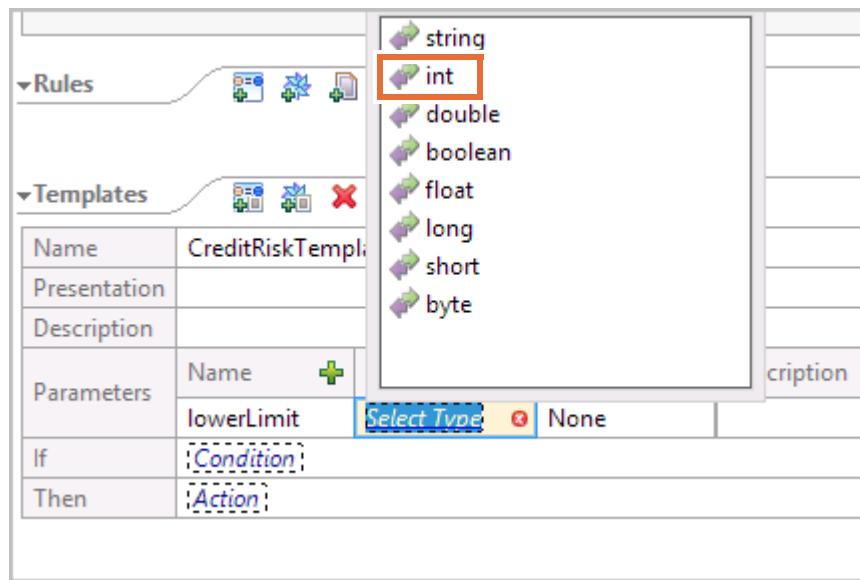
- \_\_\_ b. In the **Name** field, replace `Template 1` with: `CreditRiskTemplate`

- \_\_\_ c. In the **Parameters** row, click the plus sign (+) icon beside **Name** to add a parameter.



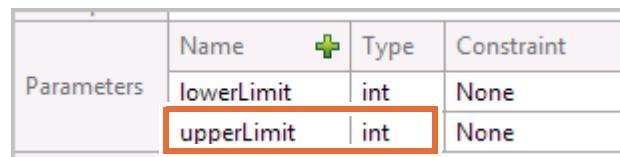
| Parameters |  | Name             | Type | Constraint | Description |
|------------|--|------------------|------|------------|-------------|
| If         |  | <i>Condition</i> |      |            |             |
| Then       |  | <i>Action</i>    |      |            |             |

- \_\_\_ d. In the **Name** column, change the parameter name from `param1` to: `lowerLimit`  
 \_\_\_ e. In the **Type** column, click the **Select Type** link and select **int**.



| Parameters |  | Name             | Type               | Constraint  | Description |
|------------|--|------------------|--------------------|-------------|-------------|
| lowerLimit |  | <i>Condition</i> | <i>Select Type</i> | <i>None</i> |             |
| If         |  | <i>Action</i>    |                    |             |             |

- \_\_\_ f. In the **Parameters** row, click the plus sign (+) icon beside **Name** to add a parameter.  
 \_\_\_ g. In the **Name** column, change the parameter name from `param1` to: `upperLimit`  
 \_\_\_ h. In the **Type** column, click the **Select Type** link and select **int**.



| Parameters |  | Name | Type | Constraint |
|------------|--|------|------|------------|
| lowerLimit |  | int  | None |            |
| upperLimit |  | int  | None |            |

- \_\_\_ i. In the **Parameters** row, click the plus sign (+) icon beside **Name** to add a parameter.  
 \_\_\_ j. In the **Name** column, change the parameter name from `param1` to: `decision`

- \_\_\_ k. In the **Type** column, click the **Select Type** link and select **string**.

|            | Name       | Type   | Constraint |
|------------|------------|--------|------------|
| Parameters | lowerLimit | int    | None       |
|            | upperLimit | int    | None       |
|            | decision   | string | None       |

- \_\_\_ l. In the **If** row, click the **Condition** link and select **All of the following are true** from the list.

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Templates | <ul style="list-style-type: none"> <li><input type="checkbox"/> var1 is equal to var2</li> <li><input type="checkbox"/> var1 is greater than var2</li> <li><input checked="" type="checkbox"/> All of the following are true</li> <li><input type="checkbox"/> Any or the following is true</li> <li><input type="radio"/> Input : CustomerApplication</li> <li><input type="radio"/> Output : CustomerApplication</li> <li><input type="radio"/> lowerLimit : int</li> <li><input type="radio"/> upperLimit : int</li> <li><input type="radio"/> decision : string</li> </ul> |
| If        | <b>Condition</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Then      | <b>Action</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

- \_\_\_ m. Under "all of the following are true," click the first **Condition** link.  
 \_\_\_ n. In the expression builder list, expand **Output : CustomerApplication** and select **creditScore**.  
 \_\_\_ o. Select the greater than (>) operator.  
 \_\_\_ p. Select **lowerLimit : int**. The completed condition resembles the following image:

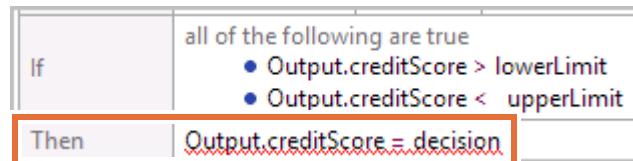
|    |                                                                                                               |
|----|---------------------------------------------------------------------------------------------------------------|
| If | all of the following are true                                                                                 |
|    | <ul style="list-style-type: none"> <li>• Output.creditScore &gt; lowerLimit</li> <li>• ;Condition;</li> </ul> |

- \_\_\_ q. Click the second **Condition** link under "all of the following are true."  
 \_\_\_ r. Expand **Output : CustomerApplication** and select **creditScore** from the list.  
 \_\_\_ s. Select the less than (<) operator.  
 \_\_\_ t. Select **upperLimit : int**. The completed "If" section of the template resembles the following image:

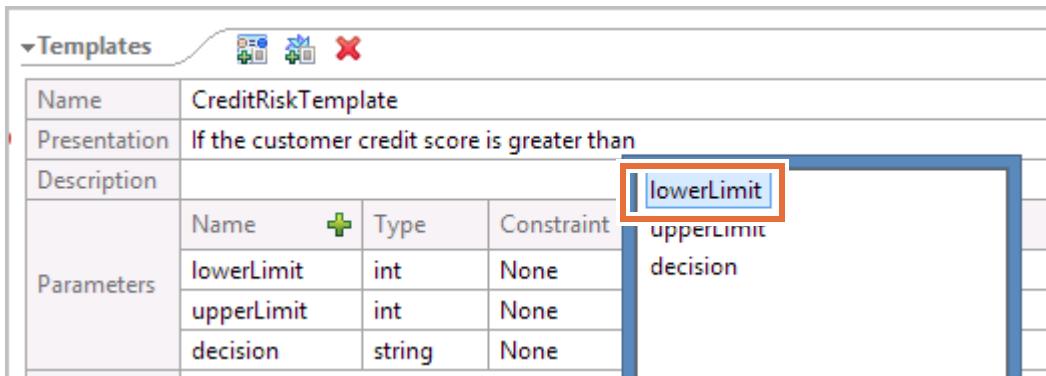
|    |                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------|
| If | all of the following are true                                                                                                        |
|    | <ul style="list-style-type: none"> <li>• Output.creditScore &gt; lowerLimit</li> <li>• Outout.creditScore &lt; upperLimit</li> </ul> |

- \_\_\_ u. In the **Then** row, click the **Action** link.

- \_\_ v. Expand **Output : CustomerApplication** and select **creditRisk** from the list.
- \_\_ w. Select the equal sign (=) operator.
- \_\_ x. Select **decision : string**. The completed “Then” portion of the template resembles the following image:



- \_\_ y. In the **Presentation** field, type: If the customer credit score is greater than (insert a space at the end of the text).
- \_\_ z. Click the **Presentation** field after the space at the end of the word **than** to reveal the expression builder.
- \_\_ aa. Select **lowerLimit** from the list.



- \_\_ ab. After **lowerLimit**, type a space followed by the text **and less than** followed by another space.
- \_\_ ac. Click the **Presentation** field after the space at the end of the word **than** to reveal the expression builder.
- \_\_ ad. Select **upperLimit** from the list.
- \_\_ ae. After **upperLimit**, type a space followed by the text **then the credit risk is** followed by another space.
- \_\_ af. Click the **Presentation** field after the space at the end of the word **is** to reveal the expression builder.
- \_\_ ag. Select **decision** from the list.

\_\_ ah. Save your changes. The completed template resembles the following figure:

| Templates    |                                                                                                                                                          |      |            |             |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|-------------|
| Name         | CreditRiskTemplate                                                                                                                                       |      |            |             |
| Presentation | If the customer credit score is greater than <code>lowerLimit</code> and less than <code>upperLimit</code> then the credit risk is <code>decision</code> |      |            |             |
| Description  |                                                                                                                                                          |      |            |             |
| Parameters   | Name                                                                                                                                                     | Type | Constraint | Description |
|              | lowerLimit                                                                                                                                               | int  | None       |             |
|              | upperLimit                                                                                                                                               | int  | None       |             |
| If           | all of the following are true                                                                                                                            |      |            |             |
|              | <ul style="list-style-type: none"> <li>• Output.creditScore &gt; lowerLimit</li> <li>• Output.creditScore &lt; upperLimit</li> </ul>                     |      |            |             |
| Then         | Output.creditRisk = decision                                                                                                                             |      |            |             |

- \_\_ 3. Create an initialization action rule that is named `MapOutput` that copies the data from the Input business object and assigns it to the Output business object. The rule logic is implemented by using the following expression: `Output = copyBO(Input)`

\_\_ a. In the **Rules** section of the editor, click the **Add Action Rule** icon.



- \_\_ b. Change the **Name** from `Rule1` to: `MapOutput`  
 \_\_ c. Select the **Action** link and select **Output : CustomerApplication** from the list.  
 \_\_ d. Select the equal sign (=) operator.  
 \_\_ e. Select **Copy BO**.  
 \_\_ f. Select **Input : CustomerApplication**.  
 \_\_ g. The completed “Action” section of the rule resembles the following figure:

| Rules        |                                       |
|--------------|---------------------------------------|
| Name         | MapOutput                             |
| Presentation |                                       |
| Action       | <code>Output = copyBO( Input )</code> |

4. Implement the logic of the rule set by creating the following business rules. Use the **CreditRiskTemplate** you created previously to generate the rules.

- If the **creditScore** value is less than 4, then the **creditRisk** is **HIGH**.
- If the **creditScore** value is in the range of 4 – 7, then the **creditRisk** is **MED**.
- If the **creditScore** value is in the range of 8 – 11, then the **creditRisk** is **LOW**.
- Otherwise, set the **creditRisk** to **HIGH**.

- a. In the **Rules** section of the editor, click the **Add Template Rule** icon.



- b. Select **CreditRiskTemplate** from the list.  
c. In the **Name** field, replace **Rule1** with: **RiskHIGH**  
d. In the **Presentation** field, click the **Enter Value** link in the first text box.  
e. Type **0** in the box.  
f. In the **Presentation** field, click the **Enter Value** link in the second text box.  
g. Type **4** in the box.  
h. In the **Presentation** field, click the **Enter Value** link in the third text box.  
i. Type **HIGH** in the box. The completed rule resembles the following figure:

|              |                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------|
| Name         | RiskHIGH                                                                                                         |
| Template     | CreditRiskTemplate                                                                                               |
| Presentation | If the customer credit score is greater than <b>0</b> and less than <b>4</b> then the credit risk is <b>HIGH</b> |

- j. Click the **Add Template Rule** icon to add another rule to the rule set.



- k. Select **CreditRiskTemplate** from the list.  
l. In the **Name** field, replace **Rule1** with: **RiskMED**  
m. In the **Presentation** field, click the **Enter Value** link in the first text box.  
n. Type **3** in the box.  
o. In the **Presentation** field, click the **Enter Value** link in the second text box.  
p. Type **8** in the box.  
q. In the **Presentation** field, click the **Enter Value** link in the third text box.

- \_\_\_ r. Type MED in the box. The completed rule resembles the following figure:

|              |                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------|
| Name         | RiskMED                                                                                                         |
| Template     | CreditRiskTemplate                                                                                              |
| Presentation | If the customer credit score is greater than <b>3</b> and less than <b>8</b> then the credit risk is <b>MED</b> |

- \_\_\_ s. In the **Rules** section of the editor, click the **Add Template Rule** icon.



- \_\_\_ t. Select **CreditRiskTemplate** from the list.  
 \_\_\_ u. In the **Name** field, replace **Rule1** with: **RiskLOW**  
 \_\_\_ v. In the **Presentation** field, click the **Enter Value** link in the first text box.  
 \_\_\_ w. Type **7** in the box.  
 \_\_\_ x. In the **Presentation** field, click the **Enter Value** link in the second text box.  
 \_\_\_ y. Type **12** in the box.  
 \_\_\_ z. In the **Presentation** field, click the **Enter Value** link in the third text box.  
 \_\_\_ aa. Type **LOW** in the box. The completed rule resembles the following figure:

|              |                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------|
| Name         | RiskLOW                                                                                                          |
| Template     | CreditRiskTemplate                                                                                               |
| Presentation | If the customer credit score is greater than <b>7</b> and less than <b>12</b> then the credit risk is <b>LOW</b> |

- \_\_\_ 5. Save your changes. The completed rule set resembles the following figure:

|              |                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------|
| <b>Rules</b> |                                                                                                                  |
|              |                                                                                                                  |
| Name         | <b>MapOutput</b>                                                                                                 |
| Presentation |                                                                                                                  |
| Action       | <b>Output =copyBO( Input)</b>                                                                                    |
| Name         | <b>RiskHIGH</b>                                                                                                  |
| Template     | CreditRiskTemplate                                                                                               |
| Presentation | If the customer credit score is greater than <b>0</b> and less than <b>4</b> then the credit risk is <b>HIGH</b> |
| Name         | <b>RiskMED</b>                                                                                                   |
| Template     | CreditRiskTemplate                                                                                               |
| Presentation | If the customer credit score is greater than <b>3</b> and less than <b>8</b> then the credit risk is <b>MED</b>  |
| Name         | <b>RiskLOW</b>                                                                                                   |
| Template     | CreditRiskTemplate                                                                                               |
| Presentation | If the customer credit score is greater than <b>7</b> and less than <b>12</b> then the credit risk is <b>LOW</b> |

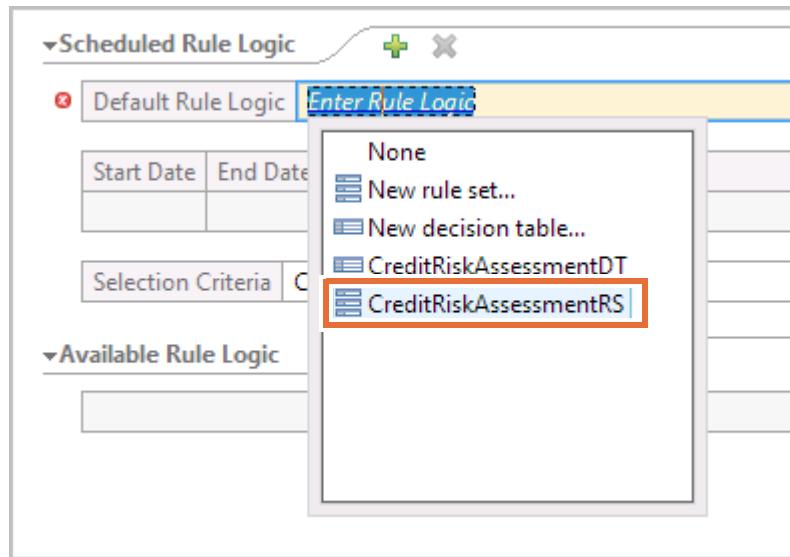
- \_\_\_ 6. Close the rule set editor. Continue to ignore any errors in the **Problems** view.
- \_\_\_ 7. Add CreditRiskAssessmentRS to the CreditRiskAssessment rule group as the Default Rule Logic, and add CreditRiskAssessmentDT to the Available Rule Logic.
  - \_\_\_ a. In the Business Integration view, expand **FoundationServices > Integration Logic > Rule Groups > businessrules**.
  - \_\_\_ b. Double-click **CreditRiskAssessment** to open the rule group editor.

**Note**

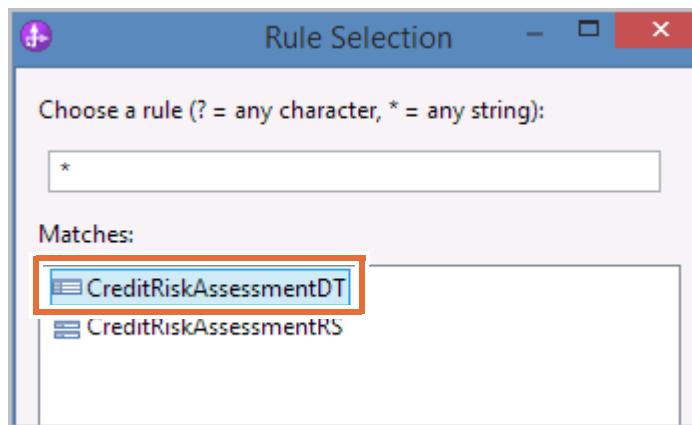
If the `businessrules` folder is not displayed in the view, click the “Show/Hide Folders” icon.



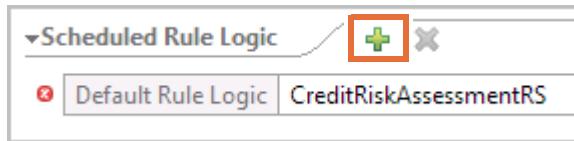
- \_\_\_ c. Click the **Enter Rule Logic** link in the **Default Rule Logic** section.
- \_\_\_ d. Select **CreditRiskAssessmentRS** from the list.



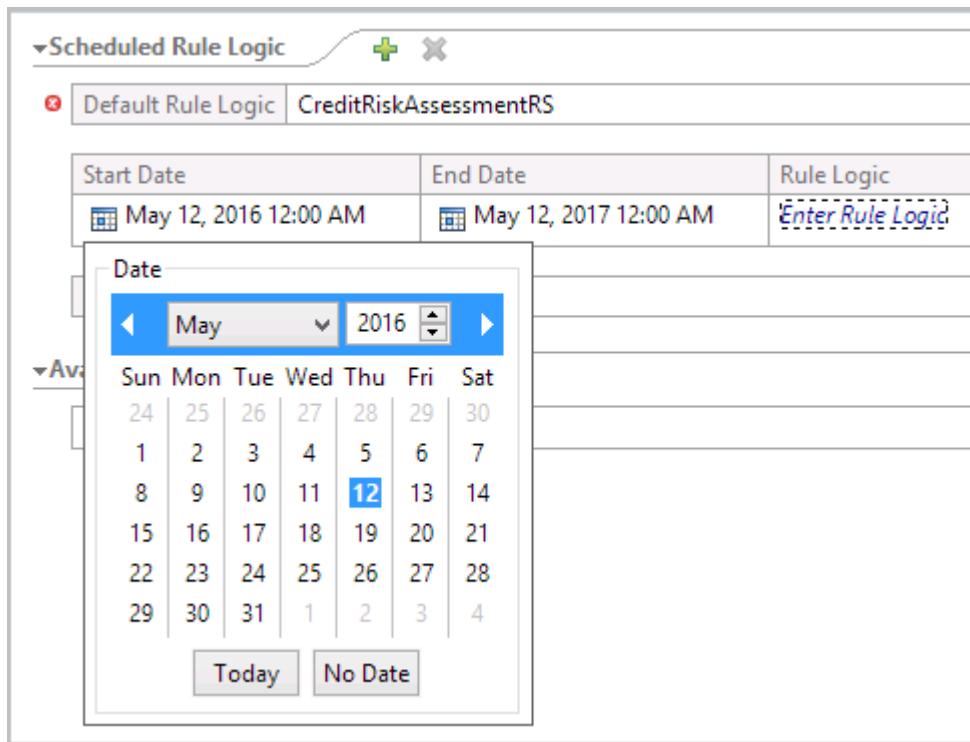
- \_\_\_ e. In the **Available Rule Logic** section of the editor, click the plus sign (+) icon.
- \_\_\_ f. Select **CreditRiskAssessmentDT** in the Rule Selection dialog box.



- \_\_\_ g. Click **OK** to add the decision table to the Available Rule Logic section.
- \_\_\_ 8. Complete the **Start Date**, **End Date**, and **Rule Logic** entries in the selection table so that the **CreditRiskAssessmentRS** rule set is invoked when the **Current date** is between **12 AM** today and **11:59 PM** one year from now.
  - \_\_\_ a. Click the **Add Date Selection Entry** icon (the green plus icon) to set the **Start Date** and **End Date** for the target rule logic.



- \_\_\_ b. Click the **Start Date** calendar icon and if the date is not set automatically, select the current date. The dates that you choose are not going to be the same dates that are shown here.



The time is set to **12 AM** automatically.

- \_\_\_ c. Click the **End Date** calendar icon and select a date one year from today.  
 \_\_\_ d. For the End Date value, place your cursor inside **12:00 AM** and manually change the time to: **11:59 PM**  
 \_\_\_ e. Click the **Enter Rule Logic** link in the **Rule Logic** column of the table.  
 \_\_\_ f. Select **CreditRiskAssessmentRS** from the list. The completed Scheduled Rule Logic section of the rule group resembles the following figure:

| Scheduled Rule Logic                                   |                        |                        |
|--------------------------------------------------------|------------------------|------------------------|
| <input checked="" type="checkbox"/> Default Rule Logic | CreditRiskAssessmentRS |                        |
| Start Date                                             | End Date               | Rule Logic             |
| May 12, 2016 12:00 AM                                  | May 12, 2017 11:59 PM  | CreditRiskAssessmentRS |
| Selection Criteria                                     |                        | Current date           |

| Available Rule Logic     |                        |
|--------------------------|------------------------|
| <input type="checkbox"/> | CreditRiskAssessmentRS |
| <input type="checkbox"/> | CreditRiskAssessmentDT |

**Information**

For this exercise, the provided examples use the date range that starts May 12, 2016 and ends May 12, 2017. The dates that you choose are not going to be the same dates.

- 9. Save your changes and close the rule group editor. Verify that the **Problems** view shows no errors. You can ignore any warnings.

## **Part 2: Incorporate a rule group component into an assembly diagram**

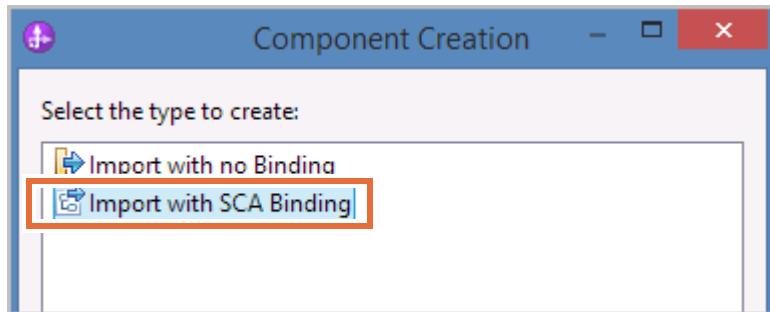
In this portion of the exercise, you add the CreditRiskAssessment rule group to the FoundationServices assembly diagram, and you generate an export component for the rule group with an SCA binding. This action allows other IBM Process Server SCA applications to call FoundationServices in the most efficient manner possible.

You then create an import component on the FoundationModule assembly diagram and wire it to the CreditRiskAssessmentPartner reference of the AccountVerification business process. When the Credit Risk Assessment invoke activity is reached, the process uses the import component that is wired to the CreditRiskAssessmentPartner reference to call the export component in FoundationServices.

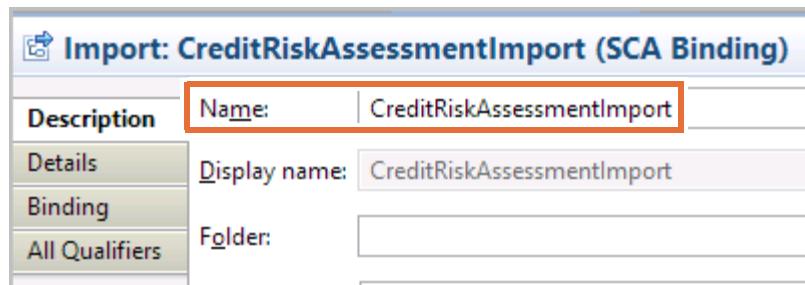
To assemble the application and test the business rules:

- 1. Add the CreditRiskAssessment rule group component to the FoundationServices assembly diagram.
  - a. If the assembly diagram is not already open, in the Business Integration view, expand **FoundationServices** and double-click **Assembly Diagram**.
  - b. In the Business Integration view, expand **FoundationServices > Integration Logic > Rule Groups > businessrules**.
  - c. Drag the **CreditRiskAssessment** rule group onto the assembly diagram.
- 2. Generate an export component for the CreditRiskAssessment rule group with an SCA binding.
  - a. On the FoundationServices assembly diagram, right-click **CreditRiskAssessment** and click **Generate Export > SCA binding** from the menu.
  - b. Accept the default export name: **CreditRiskAssessmentExport**
  - c. Save the assembly diagram.
- 3. Create an import component on the FoundationModule assembly diagram that is used to invoke the CreditRiskAssessmentExport in FoundationServices.
  - a. In the Business Integration view, expand **FoundationModule** and double-click **Assembly Diagram**.
  - b. In the Business Integration view, expand **FoundationServices > Assembly Diagram**.
  - c. Drag **CreditRiskAssessmentExport** onto the **FoundationModule** assembly diagram.

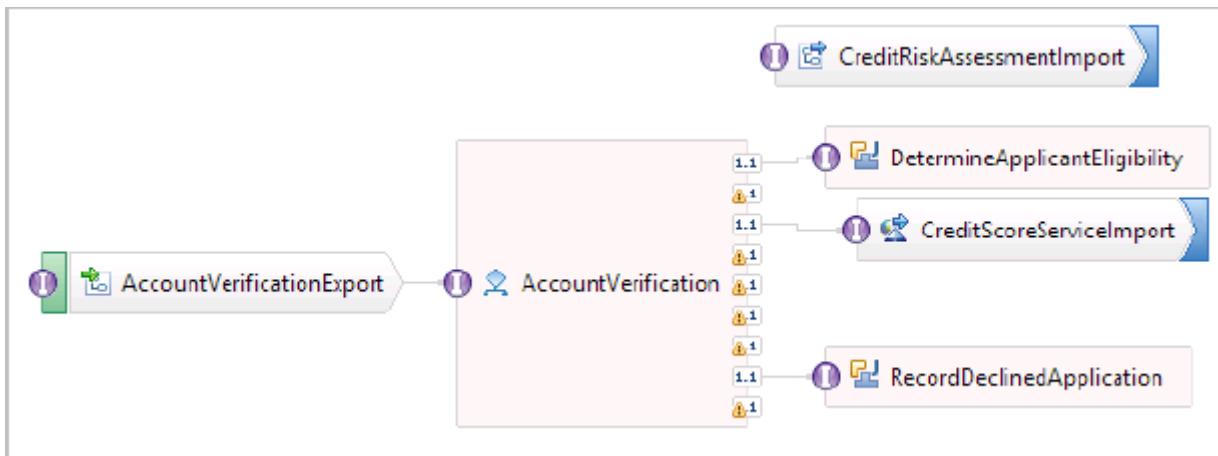
- \_\_ d. In the Component Creation dialog box, select **Import with SCA Binding**.



- \_\_ e. Click **OK**.
- \_\_ f. Select the newly created **Import1** and switch to the **Description** tab in the **Properties** view.
- \_\_ g. Change the **Name** from **Import1** to: **CreditRiskAssessmentImport**

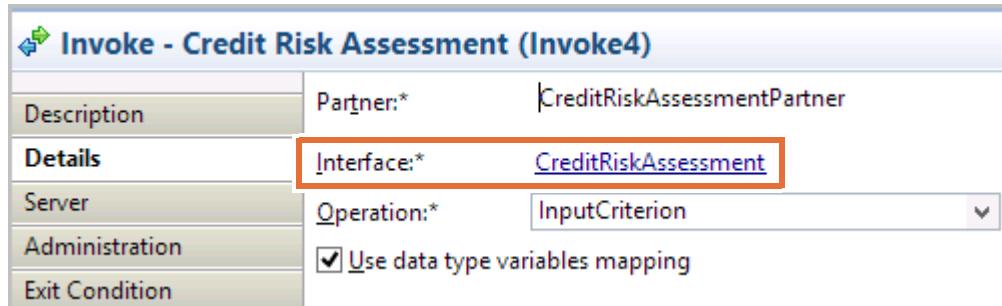


- \_\_ h. Save your changes to the assembly diagram. **AccountVerification** has three references that are wired to target services. The **CreditRiskAssessmentImport** component is going to be wired soon.

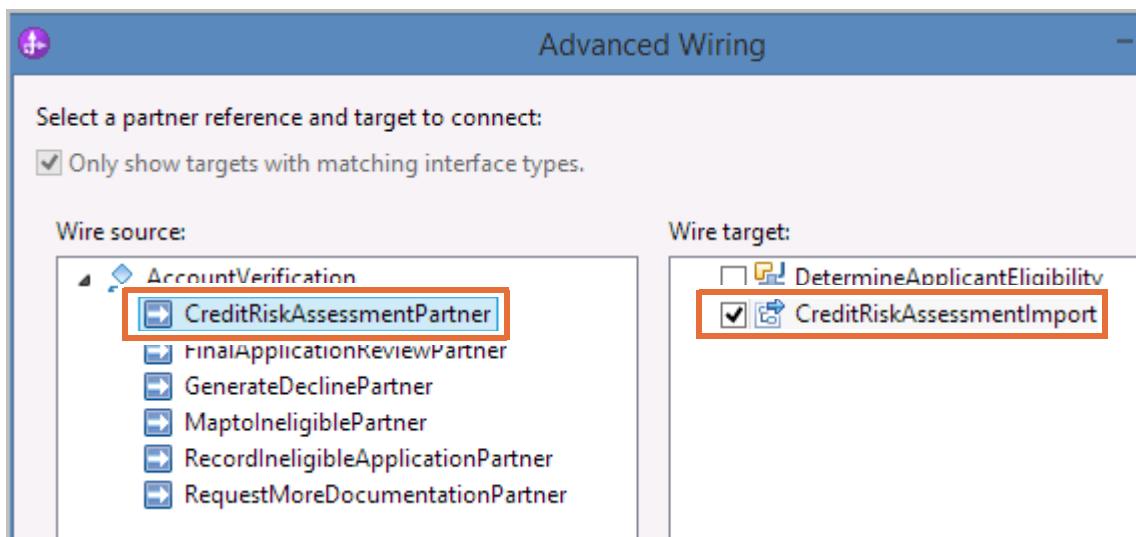


- \_\_ 4. Examine the process activity that invokes the CreditRiskAssessment rule group.
- \_\_ a. Double-click the **AccountVerification** component in the assembly diagram to open the process implementation in the business process editor.
- \_\_ b. Select the **CreditRiskAssessment** invoke activity in the business process.

- c. Switch to the **Details** tab in the **Properties** view. The activity invokes the **CreditRiskAssessment** interface of the **CreditRiskAssessment** rule group.

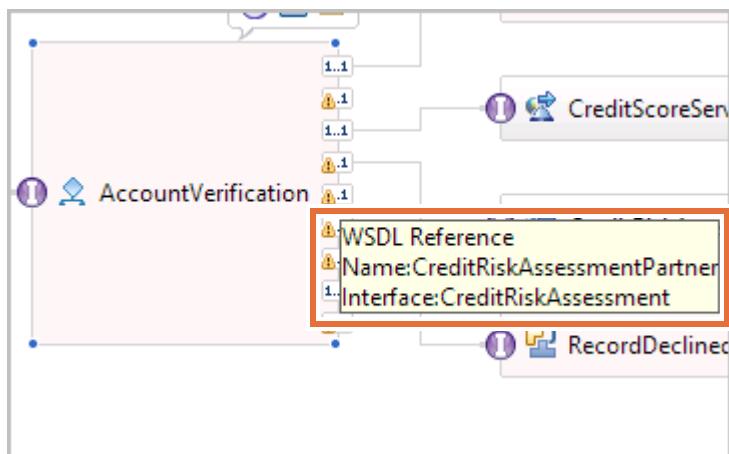


- d. Close the business process editor and return to the **FoundationModule** assembly diagram.
- 5. Wire the **CreditRiskAssessmentImport** component to the **CreditRiskAssessmentPartner** reference on the **AccountVerification** process component.
- a. Right-click the **AccountVerification** process component and click **Wire (Advanced)** from the menu.
  - b. In the Advanced Wiring dialog box, in the “Wire source” window, select **CreditRiskAssessmentPartner**.
  - c. In the “Wire target” window, select **CreditRiskAssessmentImport**.

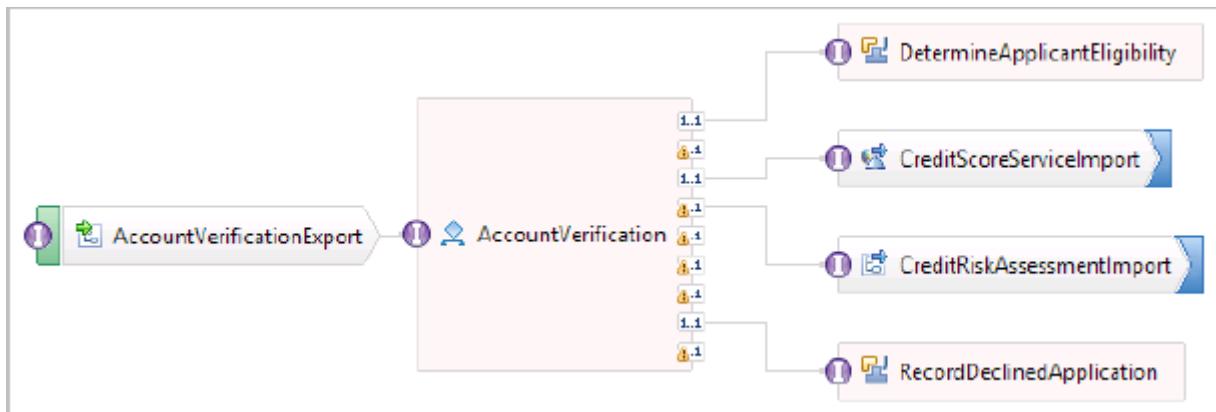


- d. Click **OK**.

- \_\_\_ e. Save your changes. You can verify the wiring by hovering over the reference that is wired to **CreditRiskAssessmentImport**. The dialog box indicates that the component is wired to **CreditRiskAssessmentPartner**.



The assembly diagram resembles the following figure:



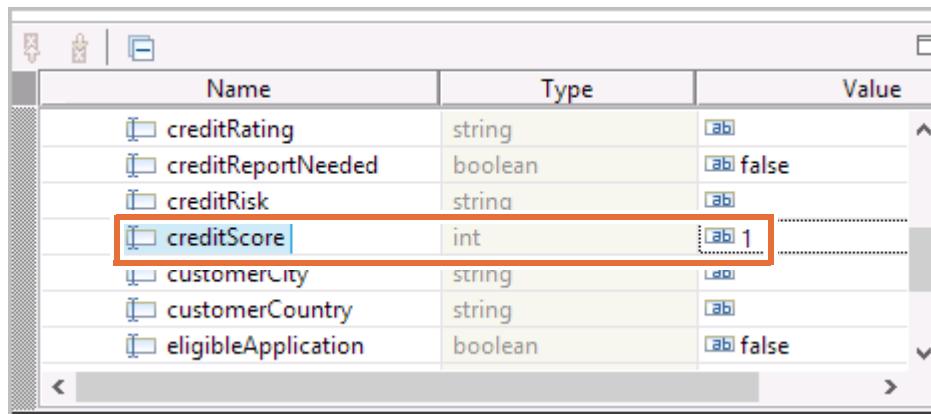
### **Part 3: Test a business rule group in the integrated test client**

After assembling the application, you test the CreditRiskAssessment business rules by invoking the CreditRiskAssessmentExport component.

To test the business rules:

- \_\_\_ 1. Start the server (if it is not already running) and deploy **FoundationServicesApp**.
  - \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Wait for the startup process to complete before continuing.  
The server is started when the message `Server server1 open for e-business` is displayed in the **Server Logs** view. The server status also changes to **Started** in the **Servers** view.
  - \_\_\_ b. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
  - \_\_\_ c. Select **FoundationServicesApp** in the **Available projects** window and click **Add**.
  - \_\_\_ d. Click **Finish**.

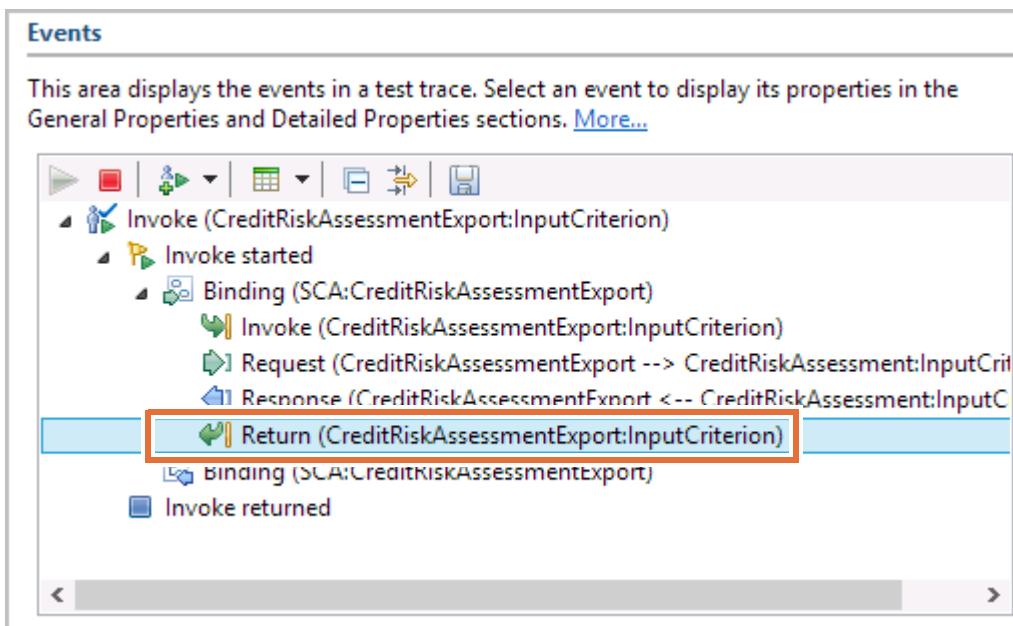
- \_\_\_ e. Wait until the module is published and started. The process is complete when the message Application started: FoundationServicesApp is displayed in the **Server Logs** view.
- \_\_\_ 2. Use several different creditScore values to test the CreditRiskAssessmentExport in the FoundationServices module. The business rules return correct creditRisk values.
  - \_\_\_ a. If the assembly diagram is not open, in the Business Integration view, expand **FoundationServices** and double-click **Assembly Diagram**.
  - \_\_\_ b. In the FoundationServices assembly diagram, right-click **CreditRiskAssessmentExport** and click **Test Component**.
  - \_\_\_ c. In the **Initial request parameters** section, change the **creditScore** value to 1. It is not necessary to populate any of the remaining fields.



| Name                | Type    | Value         |
|---------------------|---------|---------------|
| creditRating        | string  | [ab]          |
| creditReportNeeded  | boolean | [ab] false    |
| creditRisk          | string  | [ab]          |
| <b>creditScore</b>  | int     | [ab] <b>1</b> |
| customerCity        | string  | [ab]          |
| customerCountry     | string  | [ab]          |
| eligibleApplication | boolean | [ab] false    |

- \_\_\_ d. Click **Continue** to run the test.
- \_\_\_ e. When the **Select a Deployment Location** dialog box is displayed, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_\_ f. When the User Login dialog box is displayed, accept the default values for the **User ID** and **Password** fields. Click **OK**.

- \_\_\_ g. If it is not already selected, select the **Return(CreditRiskAssessmentExport:InputCriterion)** event in the **Events** window.



- \_\_\_ h. Review the output value for creditRisk in the **Return parameters** section. According to the CreditRiskAssessmentRS definition, a credit score of less than 4 evaluates to a credit risk of **HIGH**.

| Name                | Type    | Value                                        |
|---------------------|---------|----------------------------------------------|
| contactFirstName    | string  | Lab                                          |
| contactLastName     | string  | Lab                                          |
| contactPhoneNumber  | string  | Lab                                          |
| creditRating        | string  | Lab                                          |
| creditReportNeeded  | boolean | Lab false                                    |
| <b>creditRisk</b>   | string  | <b>Lab HIGH</b> (highlighted with a red box) |
| creditScore         | int     | Lab 1                                        |
| customerCity        | string  | Lab                                          |
| customerCountry     | string  | Lab                                          |
| eligibleApplication | boolean | Lab false                                    |

- \_\_\_ 3. Rerun the test with credit score of 6, which evaluates to a creditRisk of **MED**.

- \_\_\_ a. Click the **Invoke** icon to create another test invocation thread.



- \_\_\_ b. In the **Initial request parameters** section, change the **creditScore** value to **6**.

- \_\_\_ c. Click **Continue** to run the test.

- \_\_\_ d. Review the **Return parameters** for the **Return(CreditRiskAssessmentExport:InputCriterion)** event and verify that the **creditRisk** is **MED**.

| Name                | Type    | Value      |
|---------------------|---------|------------|
| contactFirstName    | string  | [ab]       |
| contactLastName     | string  | [ab]       |
| contactPhoneNum1    | string  | [ab]       |
| creditRating        | string  | [ab]       |
| creditReportNeeded  | boolean | [ab] false |
| creditRisk          | string  | [ab] MED   |
| creditScore         | int     | [ab] 6     |
| customerCity        | string  | [ab]       |
| customerCountry     | string  | [ab]       |
| eligibleApplication | boolean | [ab] false |
| ineligibleReason    | string  | [ab]       |

- \_\_\_ 4. Repeat the steps to test a **creditScore** value of **9**, which evaluates to a **creditRisk** of **LOW**.

| Name                | Type    | Value      |
|---------------------|---------|------------|
| contactFirstName    | string  | [ab]       |
| contactLastName     | string  | [ab]       |
| contactPhoneNum1    | string  | [ab]       |
| creditRating        | string  | [ab]       |
| creditReportNeeded  | boolean | [ab] false |
| creditRisk          | string  | [ab] LOW   |
| creditScore         | int     | [ab] 9     |
| customerCity        | string  | [ab]       |
| customerCountry     | string  | [ab]       |
| eligibleApplication | boolean | [ab] false |
| ineligibleReason    | string  | [ab]       |

- \_\_\_ 5. When you finish, close the **FoundationServices\_test** tab.  
 \_\_\_ 6. Click **No** when you are prompted to save the test trace.  
 \_\_\_ 7. Leave the server in the running state with **FoundationServicesApp** deployed.

#### Part 4: Use the **Business Rule Manager web client** to interact with business rules at run time

In this portion of the exercise, you use the Business Rules Manager web client to view the properties of your rule group and to do the necessary steps to create a rule from a template. Administrators and business users can use the Business Rules Manager web client to change rule parameters and add rules from templates, among other things. Here you learn how to make but not save the changes.



## Note

**For IBM BPM on Cloud users**

Click **Launch** on the Business Rules Manager tile to access the Business Rules Manager client.

To view your rules in the Business Rules Manager:

- 1. Start the Business Rules Manager web client.
  - a. Start Firefox and enter `http://localhost:9080/br` in the URL field.
  - b. If the **This Connection is Untrusted** window is displayed, then expand **I Understand the Risks** and click **Add Exception**.
  - c. In the Add Security Exception window, click **Confirm Security Exception**.
  - d. In the login page for the Business Rules Manager, enter `admin` in the **User ID** field and `web1sphere` in the **Password** field, and click **Login**.
- 2. Examine the contents of the **CreditRiskAssessment** rule group in the Business Rules Manager web client.
  - a. Expand **CreditRiskAssessment > InputCriterion** to view the available rules.

The screenshot shows a hierarchical tree structure under 'Business Rules Resources'. At the top level is 'CreditScoreRG'. Below it is 'CreditRiskAssessment', which is expanded to show its sub-folders: 'InputCriterion', 'CreditRiskAssessmentDT', and 'CreditRiskAssessmentRS'. The 'CreditRiskAssessmentDT' and 'CreditRiskAssessmentRS' nodes are highlighted with a red rectangular border.

- \_\_\_ b. Click the **InputCriterion** link and examine the content. This page shows the properties of the rule group (the Scheduled Rule Logic and Default Rule Logic).

**InputCriterion - Rule Schedule**

[Back](#) [Edit](#)

**General Information**

|                |                                 |        |          |
|----------------|---------------------------------|--------|----------|
| Last Published | May 12, 2016 23:56 (Local Time) | Status | Original |
| Description    |                                 |        |          |

**Scheduled Rule Logic**

Local Time ▾

| Start Date/Time                                           | End Date/Time      | Effective Rule Logic                   |
|-----------------------------------------------------------|--------------------|----------------------------------------|
| May 12, 2016 00:00                                        | May 12, 2017 23:59 | <a href="#">CreditRiskAssessmentRS</a> |
| Default Rule Logic (If no other rule logic is applicable) |                    | <a href="#">CreditRiskAssessmentRS</a> |

- \_\_\_ c. Click **Back** at the top of the page to return to the previous page. Make sure that you do not click back for the browser because that throws an error. You must click **Back** on the page.
- \_\_\_ d. Click the **CreditRiskAssessmentDT** link and examine the content.

**CreditRiskAssessmentDT - Decision Table**

[Back](#) [Edit](#) [Copy](#)

**General Information**

|                |                                 |                  |                                         |
|----------------|---------------------------------|------------------|-----------------------------------------|
| Name           | CreditRiskAssessmentDT          | Target Namespace | http://FoundationServices/businessrules |
| Last Published | May 12, 2016 23:56 (Local Time) | Status           | Original                                |
| Description    |                                 |                  |                                         |

**Decision Table**

|                                                                                                          |        |       |       |           |
|----------------------------------------------------------------------------------------------------------|--------|-------|-------|-----------|
| Output.creditScore  ➔ | <4     | <8    | <12   | Otherwise |
| Output.creditRisk ➔                                                                                      | "HIGH" | "MED" | "LOW" | "HIGH"    |

- \_\_\_ e. Click the **Back** button to return to the previous page. Make sure that you click **Back** on the page and not the back on the browser because that does not work for the business rules client.

- \_\_ f. Click the **CreditRiskAssessmentRS** link and examine the content.

**CreditRiskAssessmentRS - Rule Set**

[Back](#) [Edit](#) [Copy](#)

**General Information**

|                |                                 |                  |                                         |
|----------------|---------------------------------|------------------|-----------------------------------------|
| Name           | CreditRiskAssessmentRS          | Target Namespace | http://FoundationServices/businessrules |
| Last Published | May 12, 2016 23:56 (Local Time) | Status           | Original                                |
| Description    |                                 |                  |                                         |

**Rules**

| Display Name | Rule                                                                                        | Description |
|--------------|---------------------------------------------------------------------------------------------|-------------|
| RiskHIGH     | If the customer credit score is greater than 0 and less than 4 then the credit risk is HIGH |             |
| RiskMED      | If the customer credit score is greater than 3 and less than 8 then the credit risk is MED  |             |
| RiskLOW      | If the customer credit score is greater than 7 and less than 12 then the credit risk is LOW |             |

- \_\_ g. Click **Edit**. In edit mode, you can change descriptions, change individual rule parameters, change rule names, delete rules, or change the execution order by moving the rules up or down the list.
- \_\_ h. Click **New Rule from Template** to create a rule. In the Business Rules Manager, you can create rules only from templates. If the developer does not implement rule templates in the application, new rules cannot be added and rule parameters cannot be changed at run time.

**Rules**

**New Rule from Template**

| Name     | Display Name | Rule                                                                                        | Description | Action                                                                                     |
|----------|--------------|---------------------------------------------------------------------------------------------|-------------|--------------------------------------------------------------------------------------------|
| RiskHIGH | RiskHIGH     | If the customer credit score is greater than 0 and less than 4 then the credit risk is HIGH |             | <input type="button" value="Delete"/> <input checked="" type="checkbox"/> Synchronize Name |

- \_\_ i. You can create a rule by entering a **Name**, **Display Name**, **Rule parameters**, and a **Description** value.

| Templates<br>To create a new rule, fill in data in a template and click "Add" button. |                      |                      |                                                                                                                                                   |                      |                      |                                    |
|---------------------------------------------------------------------------------------|----------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|----------------------|------------------------------------|
| Template Name                                                                         | Name                 | Display Name         | Rule                                                                                                                                              | Description          | Action               |                                    |
| CreditRiskTemplate                                                                    | <input type="text"/> | <input type="text"/> | If the customer credit score is greater than <input type="text"/> and less than <input type="text"/> then the credit risk is <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="button" value="Add"/> |

- \_\_ j. Click **Close Template List** and click **Cancel**. This action prevents the rule from taking effect in the runtime environment. To apply the new rule immediately, you save the rule and publish it to the runtime environment.

Changes that you make in the Business Rules Manager are not synchronized with IBM Integration Designer. To bring runtime changes into a project, you must use the administrative console to export the rules from the runtime environment.

The following screen capture depicts the appropriate view in the administrative console. You are not required to do this task.

[Application servers > server1 > Business rules](#)

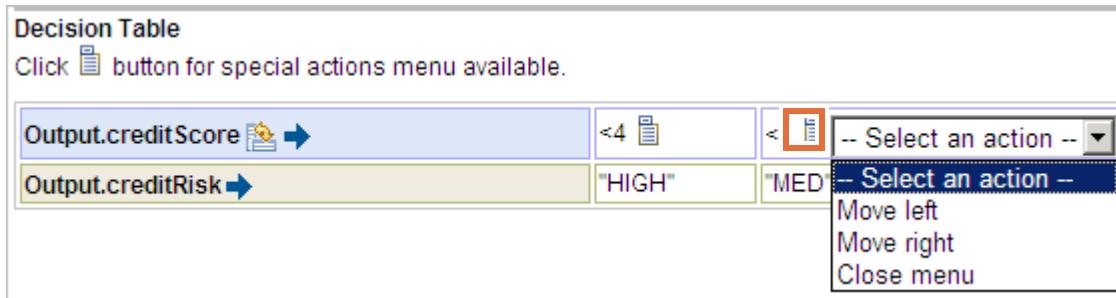
Select one or more business rules then click "Export" to export them. Click "Import" to import one or more business rules.

[+] Preferences

|                                             |                                       |                                         |
|---------------------------------------------|---------------------------------------|-----------------------------------------|
| <input type="button" value="Export"/>       | <input type="button" value="Import"/> | <input type="button" value="Cancel"/>   |
|                                             |                                       |                                         |
| Select                                      | Name                                  | Namespace                               |
| Description                                 |                                       |                                         |
| You can administer the following resources: |                                       |                                         |
| <input checked="" type="checkbox"/>         | CreditRiskAssessment                  | http://FoundationServices/businessrules |
| Total 1                                     |                                       |                                         |

- \_\_ k. Click **Back** to return to the previous page.  
 \_\_ l. Click the **CreditRiskAssessmentDT** link.  
 \_\_ m. Click **Edit**.

- \_\_ n. Click one of the **Open the actions menu** icons to view the choices available.



- \_\_ o. Click **Cancel** to return to the previous page.  
\_\_ p. Feel free to examine the application.  
\_\_ q. Click **Logout** to exit the Business Rules Manager web browser and close the browser.
- \_\_ 3. Remove the projects and (optionally) stop the server.
- \_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.  
\_\_ b. Click **Remove All** and click **Finish**.  
\_\_ c. (Optional) Stop the server.
- \_\_ 4. Close IBM Integration Designer.

## End of exercise

## **Exercise review and wrap-up**

In this exercise, you created rule sets and decision tables that contain business rules.



# Exercise 10. Implementing WebSphere (JCA) adapters

## What this exercise is about

In this exercise, you configure a WebSphere adapter as part of a business process solution.

## What you should be able to do

After completing this exercise, you should be able to:

- Configure the WebSphere Adapter for Flat Files
- Use the external service tool to generate artifacts that are used in an application
- Incorporate adapter-related SCA artifacts in an assembly diagram
- Test an adapter in the IBM Integration Designer test environment

## Introduction

IBM WebSphere Adapters make it possible for Java Platform, Enterprise Edition (Java EE) components, such as applications, to communicate with enterprise information system (EIS) resources. An EIS is the information infrastructure for an enterprise (for example, an enterprise resource planning system). A WebSphere adapter acts as an intermediary between the Java EE component and the EIS. This way, it is not necessary for the Java EE component to understand the low-level API or data structures of the EIS.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

## Exercise instructions

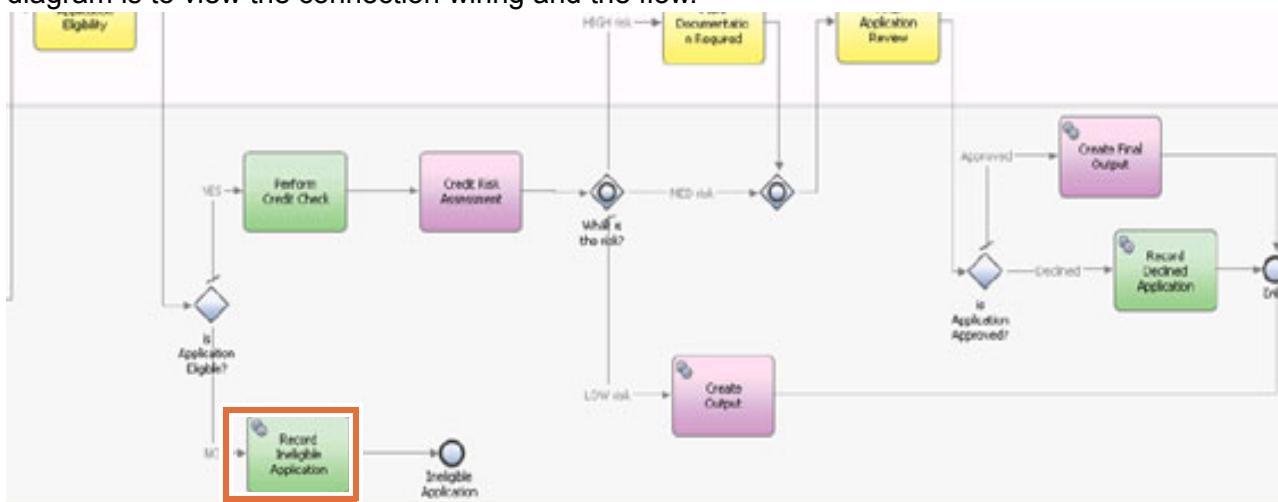
The IBM WebSphere Adapter for Flat Files connects Java Platform, Enterprise Edition components that are running on WebSphere products (such as IBM Process Server or WebSphere Enterprise Service Bus) with file systems. For example, the Java EE component, when configured to work with the adapter, can create a file with specified contents on the file system. The file can then be accessed from the file system by another application.

The IBM WebSphere Adapter for Flat Files facilitates the exchange of business data in the form of delimited records between file systems and Java EE applications. The adapter supports inbound and outbound operations and the use of business objects, business components, and business services.

In this exercise, you implement a service to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files. The AccountVerification process invokes the service when the eligibleApplication attribute is set to `false`.

See the process application model that was created previously. In this exercise, you implement the Record Ineligible Application activity. The purpose of this activity is to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files.

Do not be concerned about reading the small text in this diagram. The purpose of the solution diagram is to view the connection wiring and the flow.



### Part 1: Configure the WebSphere Adapter for Flat Files

In this portion of the exercise, you use the external service wizard to configure the flat file adapter for outbound processing. Though your application needs only outbound, the adapter supports both inbound and outbound communication with the file system.

To configure the WebSphere Adapter for Flat Files:

- \_\_ 1. Open the Exercise 10 workspace.
- \_\_ a. On your desktop, open the **Exercise Shortcuts** folder.

- \_\_\_ b. Double-click the **Exercise 10** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - \_\_\_ c. If you get a message that the server is already set to publish, then click **OK**. If the server is already running from the previous exercise, you get this message.
  - \_\_\_ d. Close the **Getting Started** tab.
- \_\_\_ 2. Create two file system directories for the flat file adapter. The adapter requires a staging directory, `C:\IneligibleAppArchive\staging`, and an output directory, `C:\IneligibleAppArchive\outdir`. When configured, the adapter writes the contents of the `IneligibleApplication` business object into the `C:\IneligibleAppArchive\outdir` directory.
- \_\_\_ a. Open **Windows Explorer** and browse to the `C:\` directory.
  - \_\_\_ b. Create a directory that is named: `IneligibleAppArchive`
  - \_\_\_ c. Create two subdirectories, named: `outdir` and `staging`



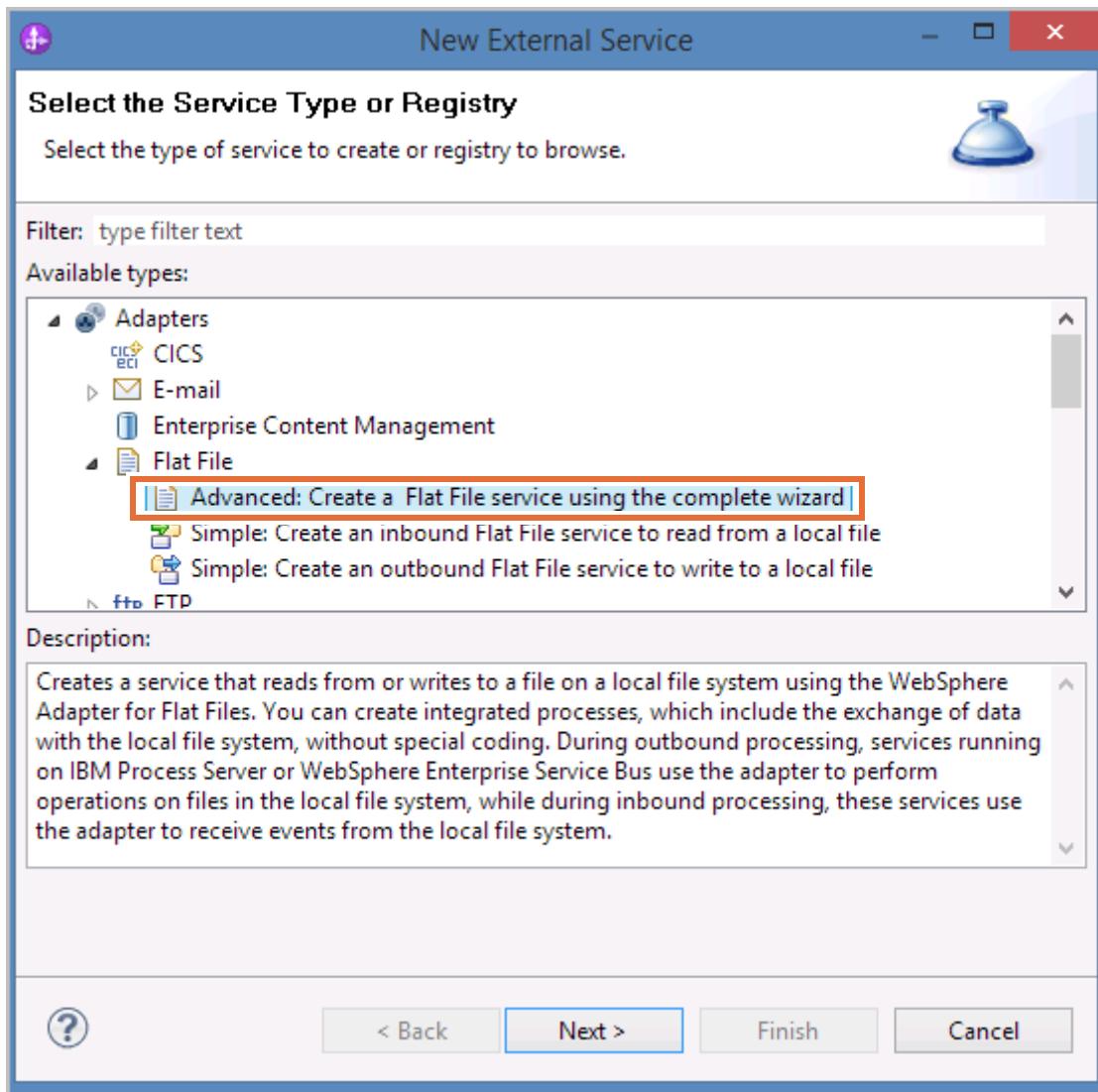
The `\staging` directory is used to temporarily store files to avoid file system write conflicts. The `\outdir` directory is the final output directory for the serialized `IneligibleApplication` business objects.

- \_\_\_ d. Close Windows Explorer.

## **Part 2: Use the external service tool to generate artifacts that are used in an application**

- \_\_\_ 1. Run the External Service wizard to configure the WebSphere Adapter for Flat Files. The adapter uses the directories that you examined previously. The adapter code is deployed inside the FoundationServices module, and a `FlatFileOutboundImport` component is created on the FoundationServices assembly diagram. The `FlatFileOutboundImport` component invokes the `recordIneligibleApplication` operation to write an `IneligibleApplication.txt` file to `C:\IneligibleAppArchive\outdir`.
  - \_\_\_ a. In the Business Integration view, right-click the **FoundationServices** module, and click **New > External Service** from the menu.

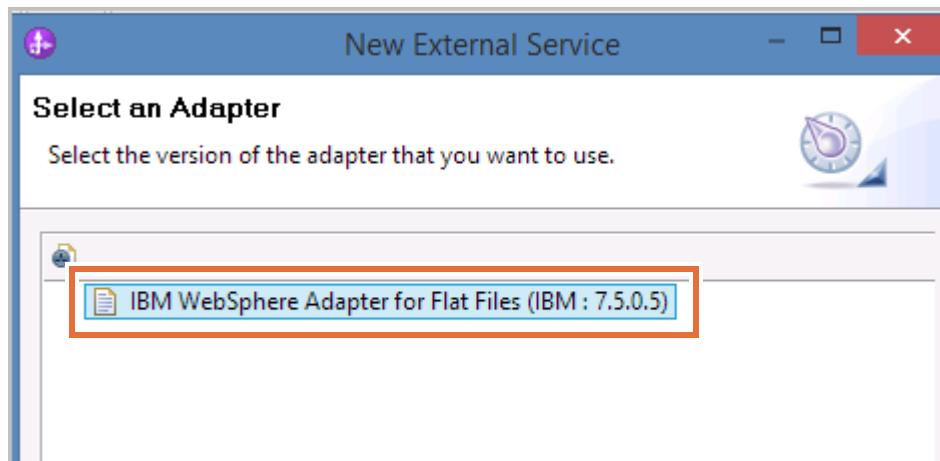
- \_\_\_ b. In the “Select the Service Type or Registry” window, in the Available types section, expand **Adapters > Flat File** and select **Advanced: Create a Flat File service using the complete wizard**.



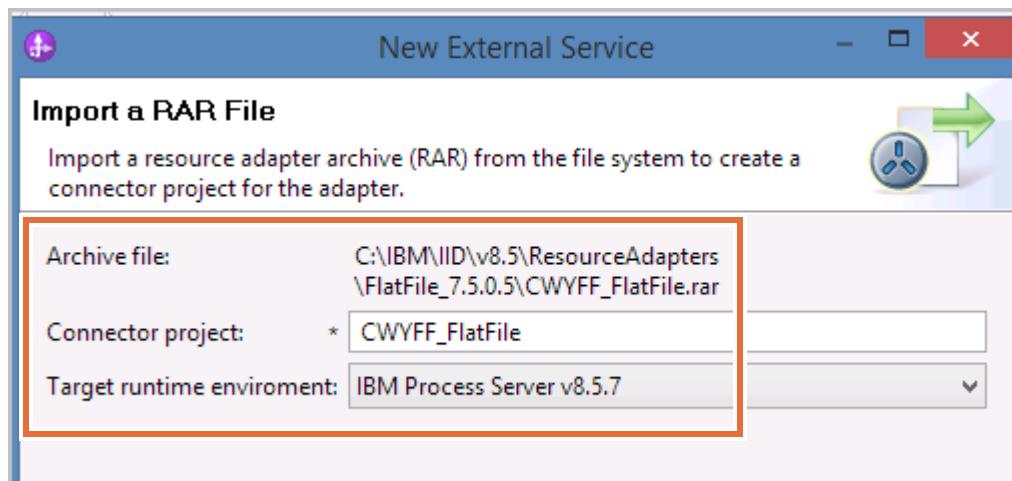
The “Simple” options use sample patterns to generate adapter services. For more information about patterns in IBM Integration Designer, see the product documentation.

- \_\_\_ c. Click **Next**.

- \_\_ d. In the “Select an Adapter” window, select **IBM WebSphere Adapter for Flat Files (IBM: 7.5.0.5)**.



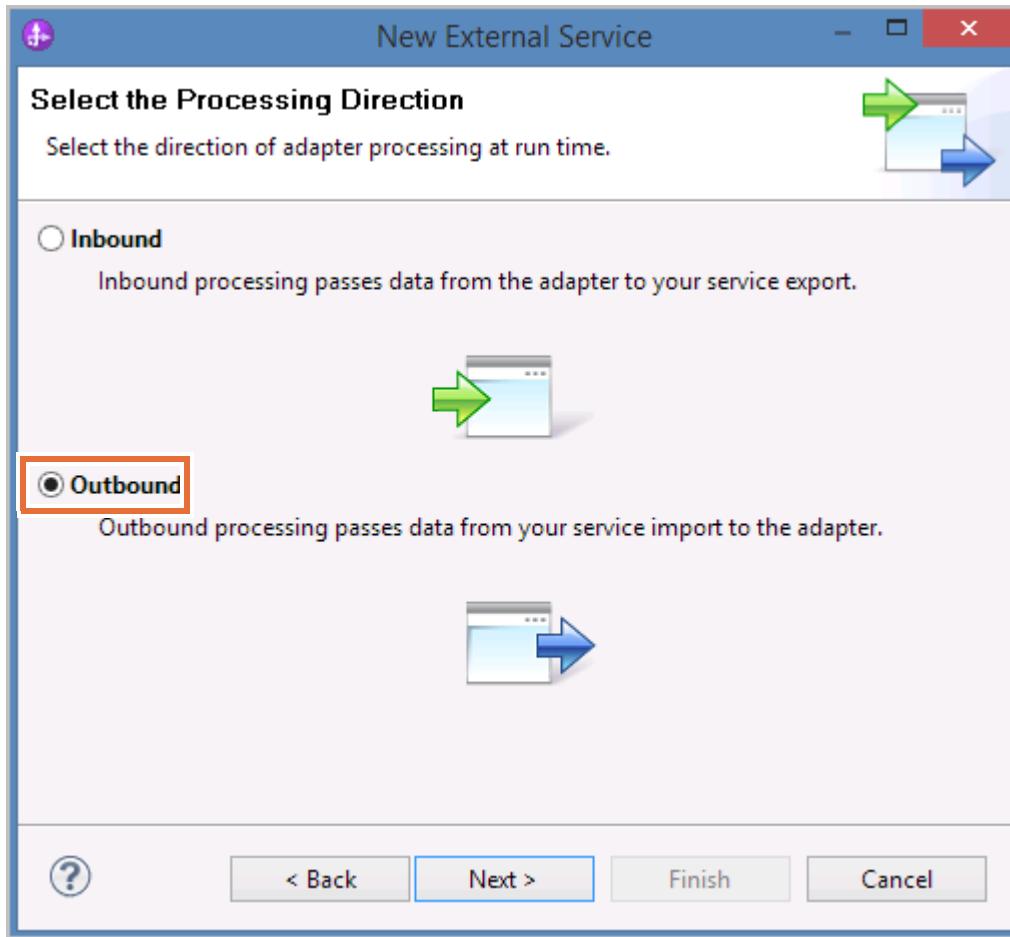
- \_\_ e. Click **Next**.
- \_\_ f. In the “Import a RAR file” window, accept the default options. The **Connector project** field is set to `CWYFF_FlatFile`, and the **Target runtime environment** field is set to **IBM Process Server v8.5.7**.



A RAR file is a Java resource adapter archive file that is used to package a resource adapter for the Java 2 Connector (J2C) architecture. The RAR file for an adapter can be deployed inside an application or at the server level for use by multiple applications. For information about deploying adapters independently, see the product documentation.

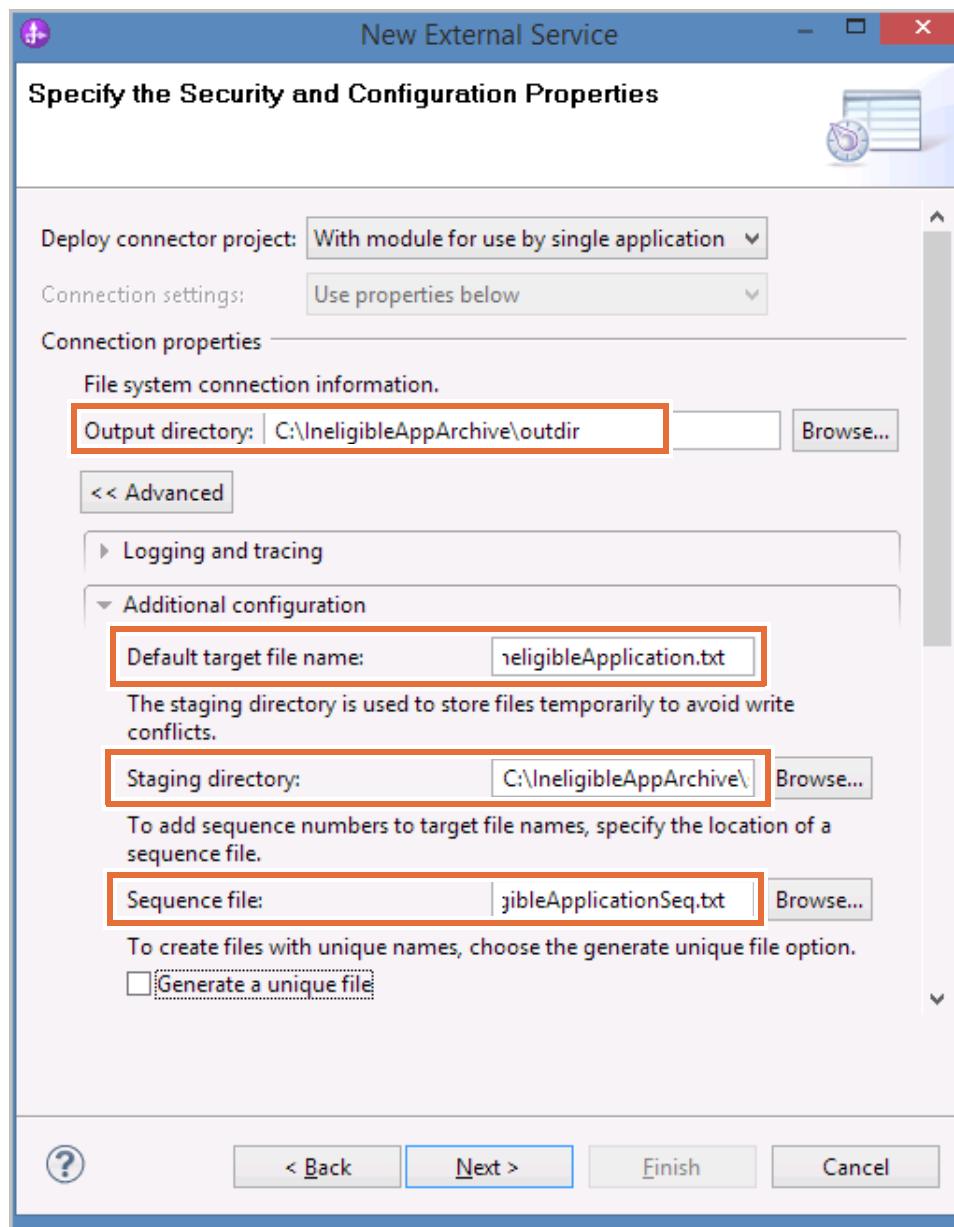
- \_\_ g. Click **Next**.

- \_\_ h. In the “Select the Processing Direction” window, select **Outbound**.

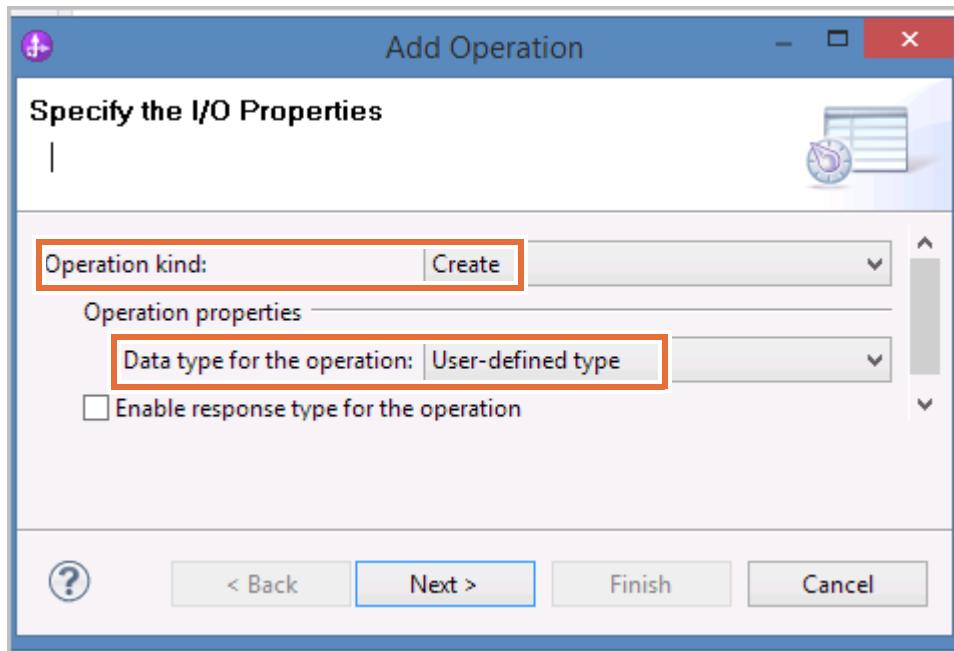


- \_\_ i. Click **Next**.

- \_\_ j. In the “Specify the Security and Configuration Properties” window, enter the following information.
- Verify that **Deploy connector project** is set to: With module for use by single application
  - For **Output directory**, click **Browse**, go to C:\IneligibleAppArchive\outdir, and click **OK**.
  - Click **Advanced** to open the additional connection properties, expand **Additional Configuration**, and enter the following information:
    - In the **Default target file name** field, type: IneligibleApplication.txt
    - Click **Browse** for **Staging directory**, go to C:\IneligibleAppArchive\staging, and click **OK**.
    - In the **Sequence file** field, type the following path and file name: C:\IneligibleAppArchive\outdir\IneligibleApplicationSeq.txt

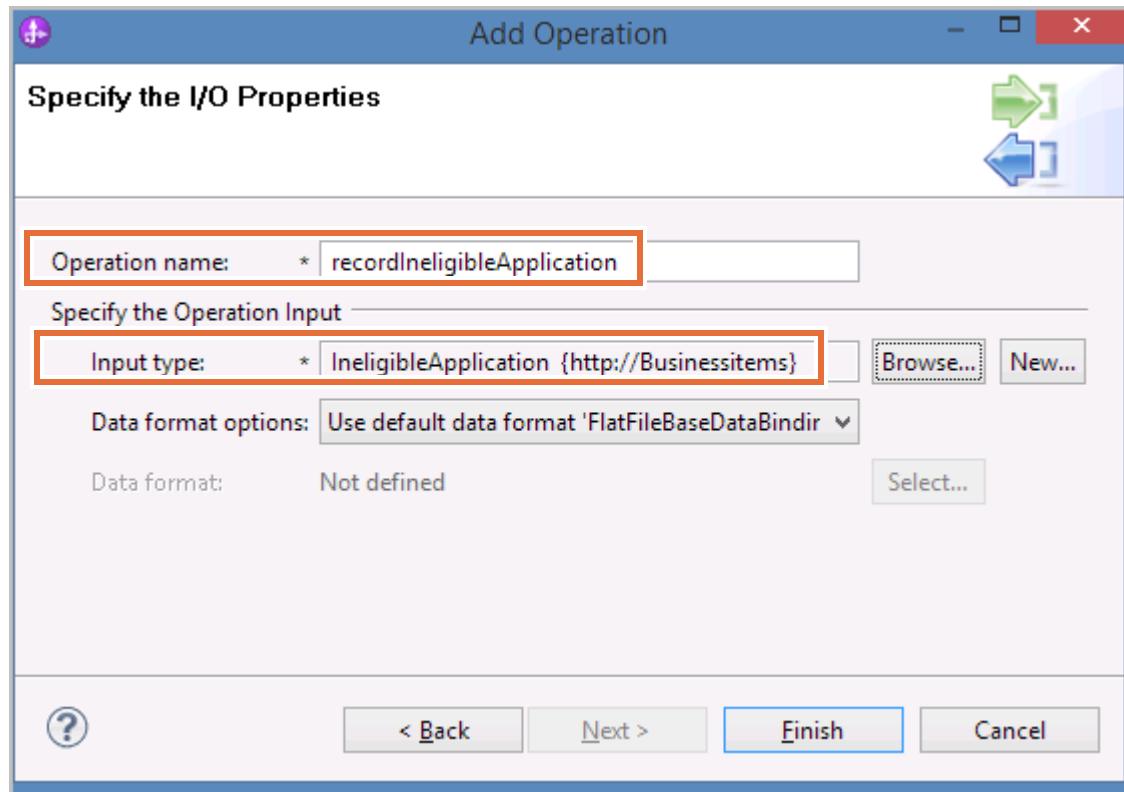


- \_\_ k. Accept the remaining default options and click **Next**.
- \_\_ l. In the “Add, Edit or Remove Operations” window, click **Add** to add an operation.
- \_\_ m. In the “Specify the I/O Properties” window, verify that **Operation kind** is set to **Create** and verify that **Data type for the operation** is set to: **User-defined type**



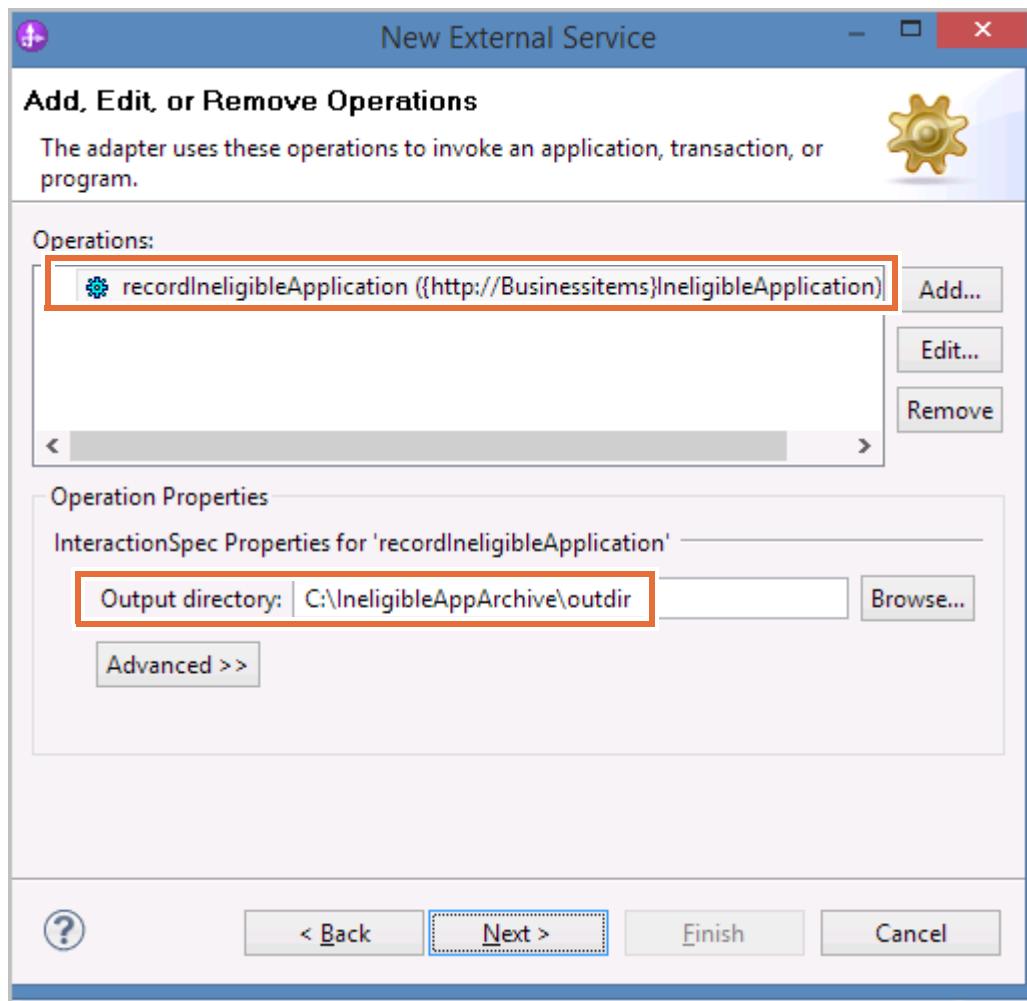
- \_\_ n. Accept the remaining default options and click **Next**.

- \_\_ o. In the second “Specify the I/O Properties” window, enter the following information.
- In the **Operation name** field, type: recordIneligibleApplication
  - Click **Browse** beside the **Input type** field, select the **IneligibleApplication** business object, and click **OK**.



- \_\_ p. Accept the remaining default options and click **Finish**.

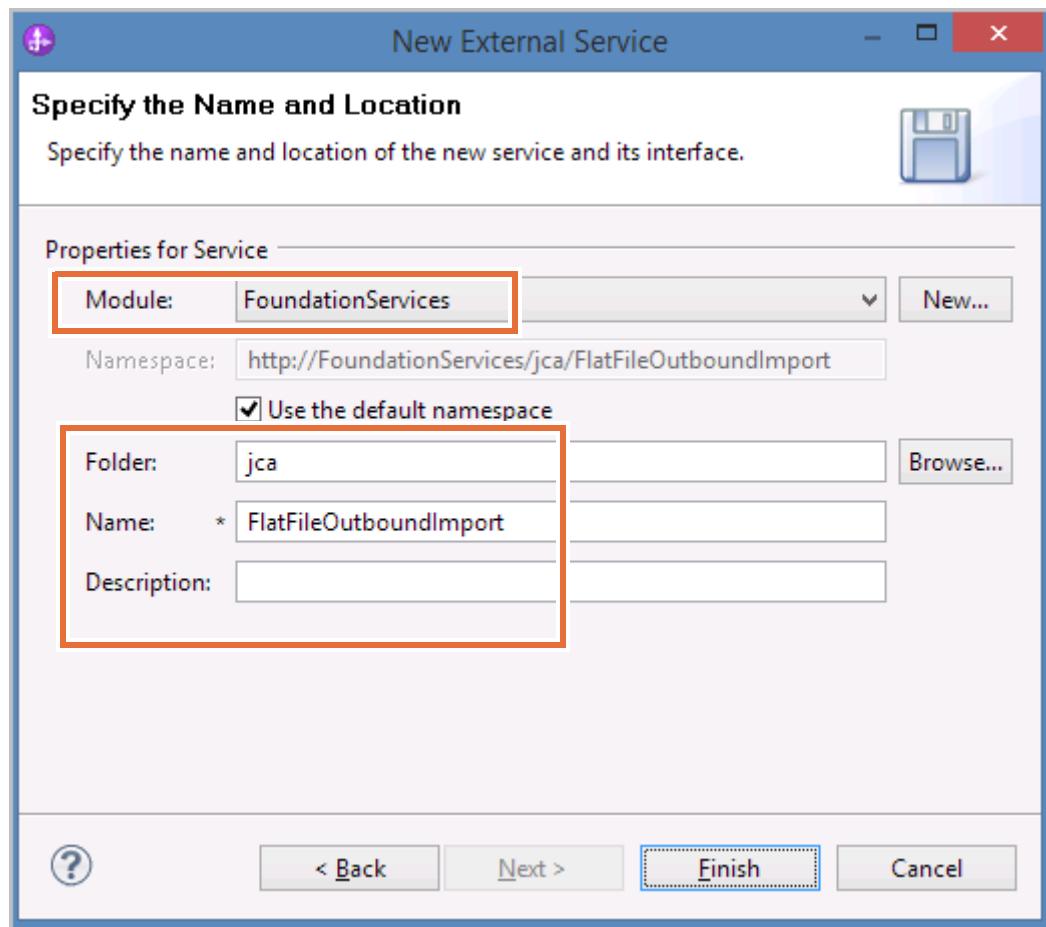
- \_\_\_ q. When you are returned to the “Add, Edit or Remove Operations” window, provide the following information:
- Verify that recordIneligibleApplication is added to the Operations window.
  - In the **Output directory** field, type `C:\IneligibleAppArchive\outdir` or use **Browse** to locate the directory.



- \_\_\_ r. Accept the remaining default options and click **Next**.

\_\_\_ s. In the “Specify the Name and Location” window, enter the following information.

- Verify that the **Module** field is set to: FoundationServices
- In the **Folder** field, type: jca
- In the **Name** field, change the name of the import component to: FlatFileOutboundImport

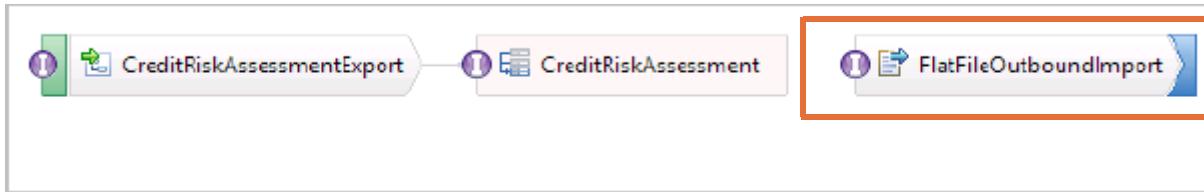


\_\_\_ t. Accept the remaining default options and click **Finish**. Wait for the workspace to build.

### **Part 3: Incorporate adapter-related SCA artifacts into an assembly diagram**

- \_\_\_ 1. Examine the artifacts that the External Service wizard generates.
- \_\_\_ a. In the Business Integration view, expand **FoundationServices** and double-click **Assembly Diagram**.

- \_\_ b. Verify that the **FlatFileOutboundImport** component was added to the assembly diagram of the **FoundationServices** module.



- \_\_ c. In the Business Integration view, expand **FoundationServices > Interfaces > jca** and double-click **FlatFileOutboundImport**.

The interface has a one-way operation that is named `recordIneligibleApplication`. The operation is one-way because you only chose outbound for the processing direction.

**Interface**

**Configuration**

|               |                                              |                                                                      |
|---------------|----------------------------------------------|----------------------------------------------------------------------|
| Name          | FlatFileImport                               | <a href="#">Refactor name</a>                                        |
| Namespace     | http://FoundationServices/jca/FlatFileImport | <a href="#">Refactor namespace</a>                                   |
| Binding Style | document literal wrapped                     | <a href="#">Change binding style to document literal non-wrapped</a> |

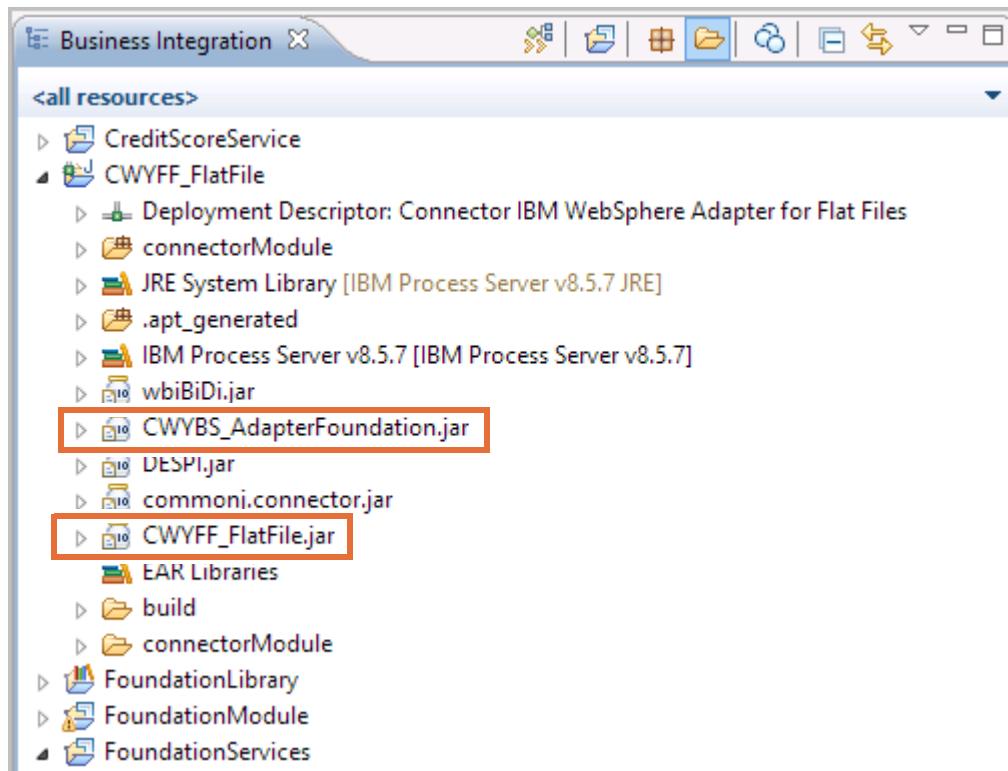
**Operations**

Operations and their parameters

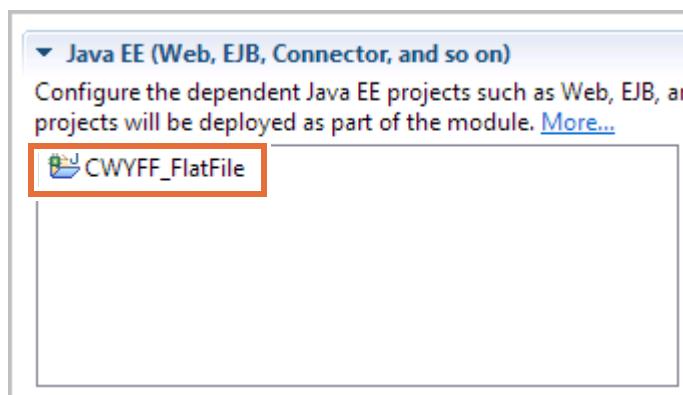
|                                    |                                  |
|------------------------------------|----------------------------------|
| Name                               |                                  |
| <b>recordIneligibleApplication</b> |                                  |
| <a href="#">Inputs</a>             | recordIneligibleApplicationInput |

- \_\_ d. Close the interface editor.  
\_\_ e. In the Business Integration view, expand the **CWYFF\_FlatFile** project.

- \_\_\_ f. Note the contents of the adapter project. In particular, note the `CWYFF_FlatFile.jar` file that contains the FlatFile adapter code and the `CWYBS_AdapterFoundation.jar` file that contains the adapter foundation classes.



- \_\_\_ g. In the Business Integration view, expand **FoundationServices** and double-click **Dependencies**.  
 \_\_\_ h. The adapter project (`CWYFF_FlatFile`) is added to the Java EE dependencies list.



- \_\_\_ i. Close the dependency editor.

## Part 4: Test an adapter in the IBM Integration Designer test environment

In this portion of the exercise, you use the newly created FlatFileOutboundImport component to test the WebSphere Adapter for Flat Files. After confirming the configuration of the adapter, you create the remaining components that are needed to process ineligible applications.

To test the flat file adapter:

- \_\_\_ 1. Start the server (if it is not running) and deploy **FoundationServicesApp**.
  - \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Alternatively, click the **Start the server** icon in the toolbar to start the server.

The server is started when the message `Server server1 open for e-business` is displayed in the **Server Logs** view. The server status also changes to **Started** in the **Servers** view.
  - \_\_\_ b. If you get a message that the server is already set to publish, then click **OK**.
  - \_\_\_ c. When the server is successfully started, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
  - \_\_\_ d. Double-click **FoundationServicesApp** in the **Available** list to add the project to the **Configured projects** list.
  - \_\_\_ e. Click **Finish**. Wait a few minutes for the application to deploy.
  - \_\_\_ f. If needed, expand the server and check the status of the newly added module. If the **FoundationServicesApp** module has a **Stopped** status in the **Servers** view, then right-click the module and click **Restart** from the menu. If prompted to do so, republish the module. Continue to the next step when the status changes to **Started**.
  - \_\_\_ g. When the application is deployed and started, the message `Application started: FoundationServicesApp` is displayed in the **Server Logs** view.
- \_\_\_ 2. Test the FlatFileOutboundImport component on the FoundationServices assembly diagram.
  - \_\_\_ a. In the **FoundationServices** assembly diagram, right-click **FlatFileOutboundImport** and click **Test Component** from the menu.

- \_\_\_ b. Enter the following test data in the **Initial request parameters** section.

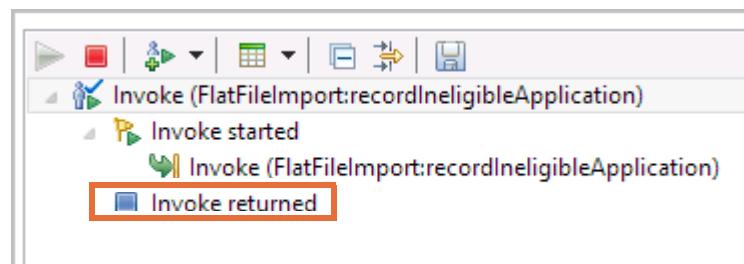
- applicationDate: 6/22/2016
- companyName: AbcCo
- requestAccountAmount: 10000
- comments: None
- ineligibleReason: Bad Credit

Initial request parameters:

Value editor  XML editor

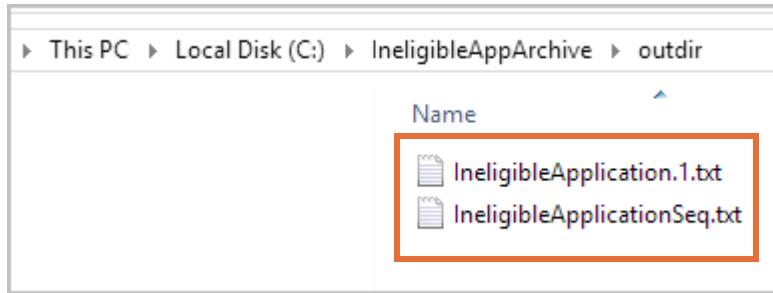
| Name                             | Type                  | Value      |
|----------------------------------|-----------------------|------------|
| recordIneligibleApplicationInput | IneligibleApplication |            |
| applicationDate                  | string                | 6/22/2016  |
| companyName                      | string                | AbcCo      |
| requestAccountAmount             | int                   | 10000      |
| comments                         | string                | None       |
| ineligibleReason                 | string                | Bad Credit |

- \_\_\_ c. Click the **Continue** icon on the Events toolbar to run the test.
- \_\_\_ d. In the “Select a Deployment Location” dialog box, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_\_ e. In the User Login dialog box, accept the default entries for **User ID** and **Password** and click **OK**.
- \_\_\_ f. The test run is complete when the blue, square **Invoke returned** icon is displayed.



- \_\_\_ g. Open Windows Explorer and browse to `C:\IneligibleAppArchive\outdir`.

- \_\_\_ h. If the test was successful, a sequence file that is named `IneligibleApplicationSeq.txt` and an output file that is named `IneligibleApplication.1.txt` are listed.



If you did another test, a file that is named `IneligibleApplication.2.txt` would be generated.

- \_\_\_ i. Open `IneligibleApplication.1.txt` in a text editor such as Notepad.  
\_\_\_ j. The file contains the sample data that you entered in the test client.

```

<?xml version="1.0" encoding="UTF-8"?>
<p:IneligibleApplication xsi:type="p:IneligibleApplication"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 |xmlns:p="http://Businessitems">
 <applicationDate>6/22/2016</applicationDate>
 <companyName>AbcCo</companyName>
 <requestAccountAmount>10000</requestAccountAmount>
 <comments>None</comments>
 <ineligibleReason>Bad Credit</ineligibleReason>
</p:IneligibleApplication>

```

- \_\_\_ k. Close `IneligibleApplication.1.txt` and close Windows Explorer.  
\_\_\_ l. Close the test client tab and click **No** when you are prompted to save the test trace.
3. Remove FoundationServicesApp from the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
- \_\_\_ b. Click **Remove All** and click **Finish**.
- \_\_\_ c. Do not stop the server.

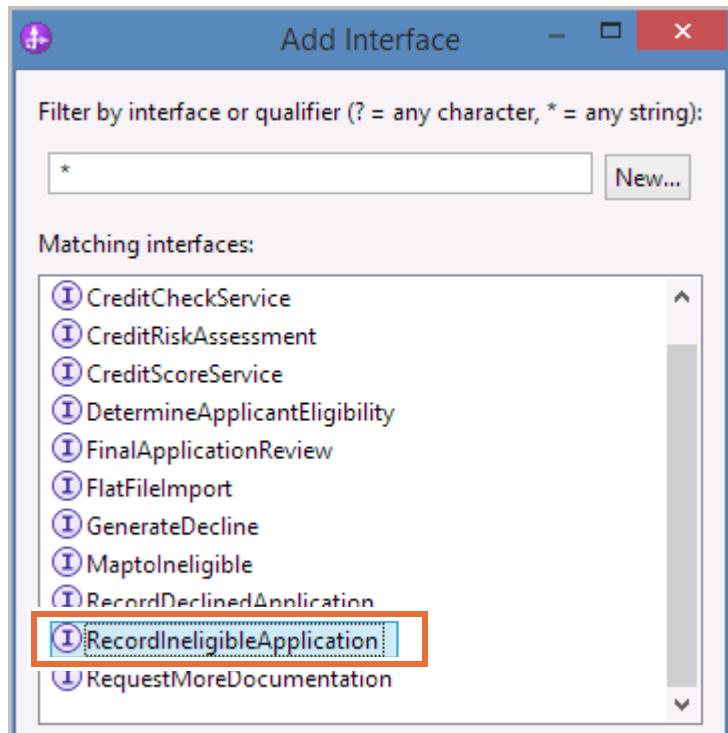
## **Part 5: Create the RecordIneligibleApplication Java component**

In this portion of the lab, you create a RecordIneligibleApplication Java component. This Java component writes informational messages to the JVM log (`SystemOut.log`) so you can track application progress. The component also creates a Message business object that contains the

rejection message: Account verification recorded this application as ineligible for the customer <company name>, and it invokes the FlatFileOutboundImport component. The application is then passed to the flat file adapter for serialization to the file system.

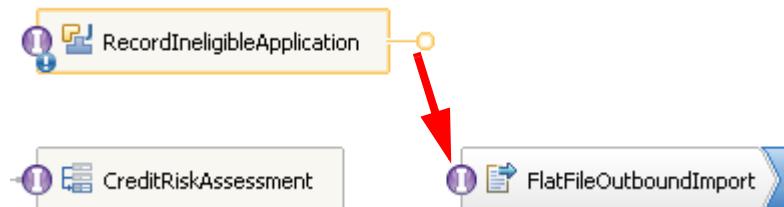
To create the RecordIneligibleApplication component:

- 1. Add a Java component that is named `RecordIneligibleApplication` to the **FoundationServices** assembly diagram.
- a. Right-click any blank space on the **FoundationServices** assembly diagram, and click **Add > Java** from the menu.
- b. Switch to the **Description** tab in the **Properties** view.
- c. Change the **Name** from `Component1` to: `RecordIneligibleApplication`
- d. Save your changes.
- 2. Add the `RecordIneligibleApplication` interface to the `RecordIneligibleApplication` Java component.
- a. Right-click the **RecordIneligibleApplication** Java component, and click **Add > Interface** from the menu.
- b. In the Add Interface dialog box, select **RecordIneligibleApplication**.

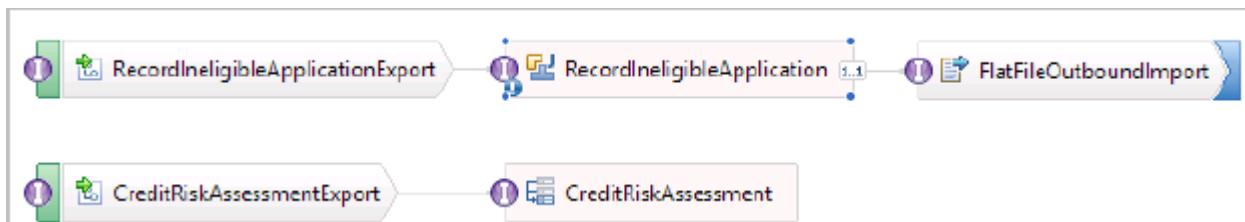


- c. Click **OK**.
- 3. Wire the **RecordIneligibleApplication** component to the **FlatFileOutboundImport** component and create a matching reference on the Java component.
- a. Hover the mouse pointer over **RecordIneligibleApplication**.

- \_\_\_ b. Drag the orange, circular handle of **RecordIneligibleApplication** to the interface of the **FlatFileOutboundImport** component.



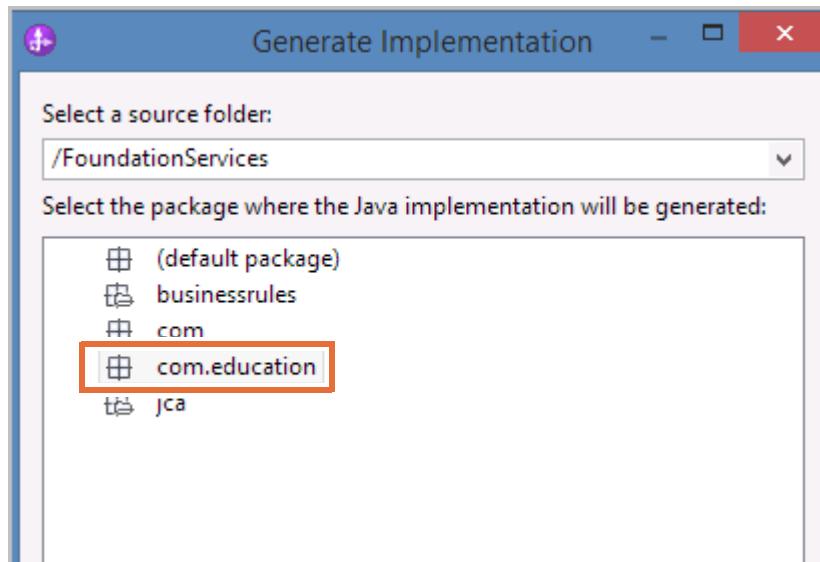
- \_\_\_ c. Click **OK** in the Add Wire dialog box to create a matching reference on the **RecordIneligibleApplication** component.  
 \_\_\_ d. Save your changes.
- \_\_\_ 4. Generate an export that is named `RecordIneligibleApplicationExport` for the **RecordIneligibleApplication** Java component. Generate the export with an SCA binding.
- \_\_\_ a. Right-click the **RecordIneligibleApplication** Java component and click **Generate Export > SCA Binding** from the menu.  
 \_\_\_ b. Accept the default export name: `RecordIneligibleApplicationExport`  
 \_\_\_ c. Save your changes. Your assembly diagram resembles the following figure:



- \_\_\_ 5. Use the snippet code in  
`C:\labfiles\Support_Files\Ex10\JavaMethod_InputCriterion.txt` to generate the implementation for the **RecordIneligibleApplication** Java component.
- \_\_\_ a. Right-click the **RecordIneligibleApplication** Java component and click **Generate Implementation** from the menu.

\_\_\_ b. In the Generate Implementation dialog box, take the following actions.

- Click **New Package**.
- Type `com.education` in the **Package name** field and click **OK**.
- When you return to the **Generate Implementation** dialog box, select `com.education`.



\_\_\_ c. Click **OK**. The `RecordIneligibleApplicationImpl.java` file opens in the Java editor.

\_\_\_ d. At the beginning of the file, click the plus symbol (+) to expand the import section.

\_\_\_ e. Add the following import statement:

```
import com.ibm.websphere.bo.BOFactory;
```

```
import com.ibm.websphere.sca.Service;
import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.websphere.bo.BOFactory;
```

\_\_\_ f. Scroll to the `public DataObject InputCriterion` method at the end of the file.

\_\_\_ g. In Windows Explorer, browse to `C:\labfiles\Support Files\Ex10`.

\_\_\_ h. Open the `JavaMethod_InputCriterion.txt` file in a text editor such as Notepad.

- \_\_ i. Copy the text in JavaMethod\_InputCriterion.txt and paste it over the green comment lines (that begin with //) in the public DataObject InputCriterion method. Be sure to remove the final return null; from the method.

```
public DataObject InputCriterion(DataObject input) {
 System.out.println("[Java] Record Ineligible Application - begins");
 System.out.println("Account verification failed for customer : "
 + input.getString("companyName"));

 storeFlatFile(input);

 // Construct the Message SDO for return to the calling process
 ServiceManager serviceManager = new ServiceManager();
 BOFactory bof = (BOFactory) serviceManager
 .locateService("com/ibm/websphere/bo/BOFactory");

 DataObject msg = bof.create("http://FoundationLibrary/businessitems", "I"
 String msgText = "Account Verification recorded this application as inel"
 + input.getString("companyName");
 msg.setString("message", msgText);
 System.out.println("[Java] Record Ineligible Application - ends");
 return msg;
}
```

If you see an error marker in the Java editor next to storeFlatFile(input), you can ignore it. The storeFlatFile method is implemented shortly.

- \_\_ j. Alternatively, enter the following code:

```
System.out.println("[Java] Record Ineligible Application - begins");
System.out.println("Account verification failed for customer : " +
 input.getString("companyName"));

storeFlatFile(input);

// Construct the Message SDO for return to the calling process
ServiceManager serviceManager = new ServiceManager();
BOFactory bof = (BOFactory) serviceManager.
locateService("com/ibm/websphere/bo/BOFactory");

DataObject msg = bof.create("http://FoundationLibrary/businessitems",
 "Message");
String msgText = "Account Verification recorded this application as
ineligible for the customer: "
+ input.getString("companyName");
msg.setString("message", msgText);
System.out.println("[Java] Record Ineligible Application - ends");
return msg;
```

- \_\_ k. Close the JavaMethod\_InputCriterion.txt file but leave Windows Explorer open.

- \_\_ l. Switch to Windows Explorer and open C:\labfiles\Support Files\Ex10\JavaMethod\_storeFlatFile.txt in a text editor such as Notepad.
  - \_\_ m. Copy the text in JavaMethod\_storeFlatFile.txt.
  - \_\_ n. Switch back to the Java editor.
  - \_\_ o. Place the cursor just above the last } marker in RecordIneligibleApplicationImpl.java.

```
 return msg;
}
```

- \_ p. Paste the code from `JavaMethod_storeFlatFile.txt` to create a method `storeFlatFile` in `RecordIneligibleApplicationImpl.java`.

```
public void storeFlatFile(DataObject input) {
 // Set up the Ineligible File BO for passing to the Flat File Outbound
 // Interface
 ServiceManager serviceManager = new ServiceManager();
 BOFactory bof = (BOFactory) serviceManager
 .locateService("com/ibm/websphere/bo/BOFactory");

 try {
 // Invoke the Flat File Outbound Interface to append the entry to
 // the file
 System.out.println(">>> Invoking the Flat File Outbound service ...");
 locateService_FlatFileOutboundImportPartner().invoke(
 "recordIneligibleApplication", input);
 System.out.println("<<< Flat File Outbound service invoked OK! ...");
 }
 catch (Throwable t) {
 t.printStackTrace();
 }
}
```

- \_\_\_ q. Alternatively, enter the following code:

```
public void storeFlatFile(DataObject input) {

 ServiceManager serviceManager = new ServiceManager();
 BOFactory bof = (BOFactory) serviceManager
 .locateService("com/ibm/websphere/bo/BOFactory");

 try {
 System.out.println(">>> Invoking the Flat File Outbound service ...");
 locateService_FlatFileOutboundImportPartner().invoke(
 "recordIneligibleApplication", input);
 System.out.println("<<< Flat File Outbound service invoked OK! ...");
 }
 catch (Throwable t) {
 t.printStackTrace();
 }
}
```

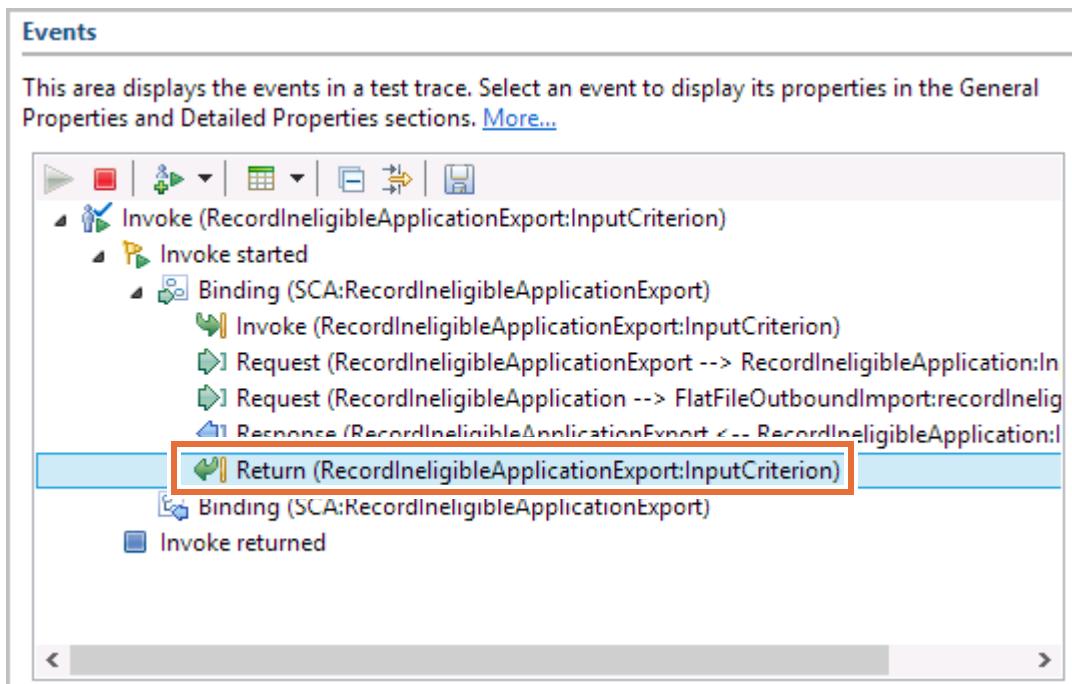
- \_\_\_ 6. Press Ctrl+S to save your changes. No error markers are showing in the left margin of the Java editor. If you receive errors, then you can replace the entire text in the implementation with the Java code that is provided in the RecordIneligibleApplicationImpl.java file under C:\Support Files\Ex10.
- \_\_\_ 7. Close the Java editor.
- \_\_\_ 8. Save the changes to the **FoundationServices** assembly diagram.
- \_\_\_ 9. Close `JavaMethod_storeFlatFile.txt` but leave Windows Explorer open.

## **Part 6: Test the RecordIneligibleApplication Java component**

In this portion of the exercise, you test the RecordIneligibleApplicationExport component to verify that both the FlatFileOutboundImport component and the RecordIneligibleApplication component are implemented correctly.

- \_\_\_ 1. Start the server (if it is not running) and deploy **FoundationServicesApp**.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Alternatively, click the **Start the server** icon in the toolbar to start the server. Wait for the startup process to complete before continuing.
- The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
- \_\_\_ b. When the server is successfully started, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
- \_\_\_ c. Double-click **FoundationServicesApp** in the **Available** list to add the project to the **Configured projects** list.
- \_\_\_ d. Click **Finish**.

- \_\_\_ e. When the application is deployed and started, the message Application started: FoundationServicesApp is displayed in the **Server Logs** view.
- \_\_\_ 2. Test the **RecordIneligibleApplicationExport** component.
  - \_\_\_ a. On the **FoundationServices** assembly diagram, right-click **RecordIneligibleApplicationExport**, and click **Test Component** from the menu.
  - \_\_\_ b. Enter the following test data in the **Initial request parameters** section.
    - applicationDate: 6/22/2016
    - companyName: AbcCo
    - requestAccountAmount: 10000
    - comments: None
    - ineligibleReason: Bad credit
  - \_\_\_ c. Click the **Continue** icon on the Events toolbar to run the test.
  - \_\_\_ d. In the “Select a Deployment Location” dialog box, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
  - \_\_\_ e. In the User Login dialog box, accept the default entries for **User ID** and **Password** and click **OK**.
  - \_\_\_ f. When the test completes, select the **Return (RecordIneligibleApplicationExport:InputCriterion)** event.



- \_\_\_ g. In the **Return parameters** section, the value in the **message** field is: Account Verification recorded this application as ineligible for the customer: AbcCo

| Value Editor |         |         | XML Source                                                                           |
|--------------|---------|---------|--------------------------------------------------------------------------------------|
|              | Name    | Type    | Value                                                                                |
|              | Output  | Message | ab                                                                                   |
|              | message | string  | Account Verification recorded this application as ineligible for the customer: AbcCo |

Messages that the **RecordIneligibleApplication** component returned are displayed in the Server Logs view.

| Type        | Time                | Thread ID | Contents                                              |
|-------------|---------------------|-----------|-------------------------------------------------------|
| Log message | May 13, 2016 21:... | 00000054  | WSVR0221I: Application started: FoundationServicesApp |
| Log message | May 13, 2016 21:... | 000001ec  | [Java] Record Ineligible Application - begins         |
| Log message | May 13, 2016 21:... | 000001ec  | Account verification failed for customer : AbcCo      |
| Log message | May 13, 2016 21:... | 000001ec  | >>> Invoking the Flat File Outbound service ...       |
| Log message | May 13, 2016 21:... | 000001ec  | <<< Flat File Outbound service invoked OK! ...        |
| Log message | May 13, 2016 21:... | 000001ec  | [Java] Record Ineligible Application - ends           |

- \_\_\_ h. In Windows Explorer, browse to C:\IneligibleAppArchive\outdir.
  - \_\_\_ i. Verify that a new `IneligibleApplication.txt` file was generated in C:\IneligibleAppArchive\outdir. The file is named `IneligibleApplication.2.txt`.
  - \_\_\_ j. Close Windows Explorer.
- \_\_\_ 3. Close the test client tab and click **No** when you are prompted to save the test trace.
- \_\_\_ 4. Remove **FoundationServicesApp** from the server and stop the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7** and click **Add and Remove** from the menu.
  - \_\_\_ b. Click **Remove All** and click **Finish**.
  - \_\_\_ c. (Optional) Stop the server.

## Wire the service to the business process

The `AccountVerification` process in `FoundationModule` invokes the `RecordIneligibleApplication` service to archive ineligible applications to the file system (by using the WebSphere Adapter for Flat Files). In this portion of the exercise, you create an import component on the `FoundationModule`

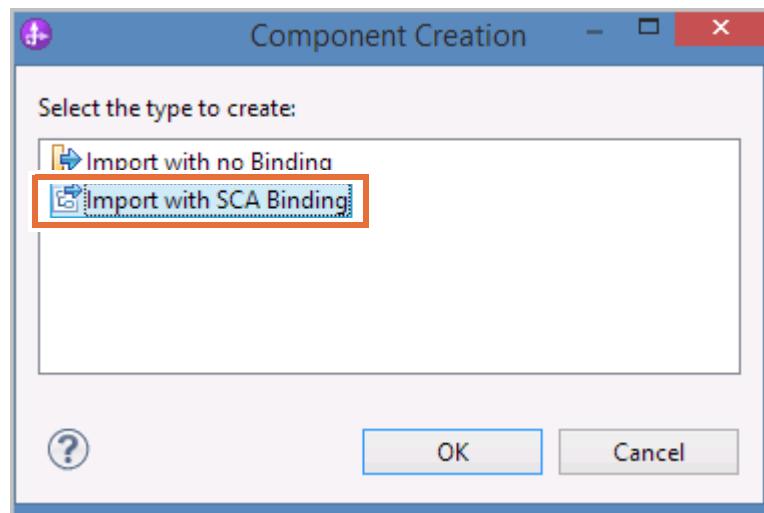
assembly diagram that calls the RecordIneligibleApplicationExport component in FoundationServices. You then wire the import component to the AccountVerification process.

To wire RecordIneligibleApplication to the AccountVerification process:

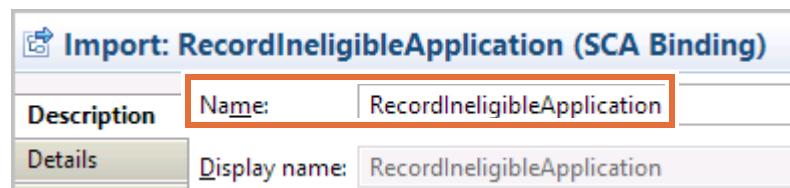
- 1. Open the **FoundationModule** assembly diagram.

  - a. In the Business Integration view, expand **FoundationModule**.
  - b. Double-click **Assembly Diagram**.

- 2. Create an import component on the **FoundationModule** assembly diagram, named: RecordIneligibleApplication
  - a. In the Business Integration view, expand **FoundationServices > Assembly Diagram**.
  - b. Drag **RecordIneligibleApplicationExport** onto the **FoundationModule** assembly diagram.
  - c. In the Component Creation dialog box, select **Import with SCA Binding**.

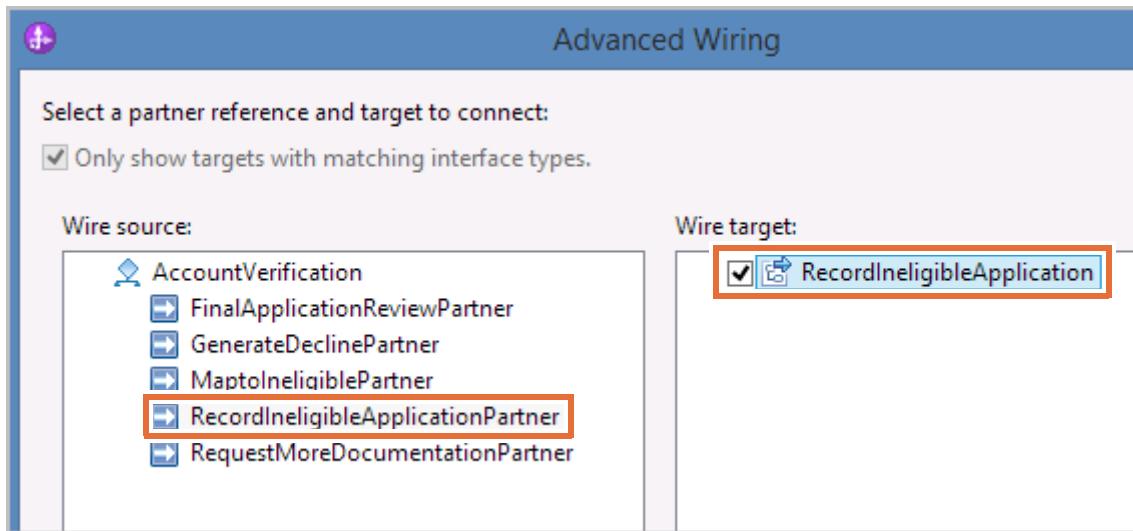


- d. Click **OK**.
- e. Switch to the **Description** tab in the **Properties** view.
- f. Change the **Name** of the import component to: RecordIneligibleApplication

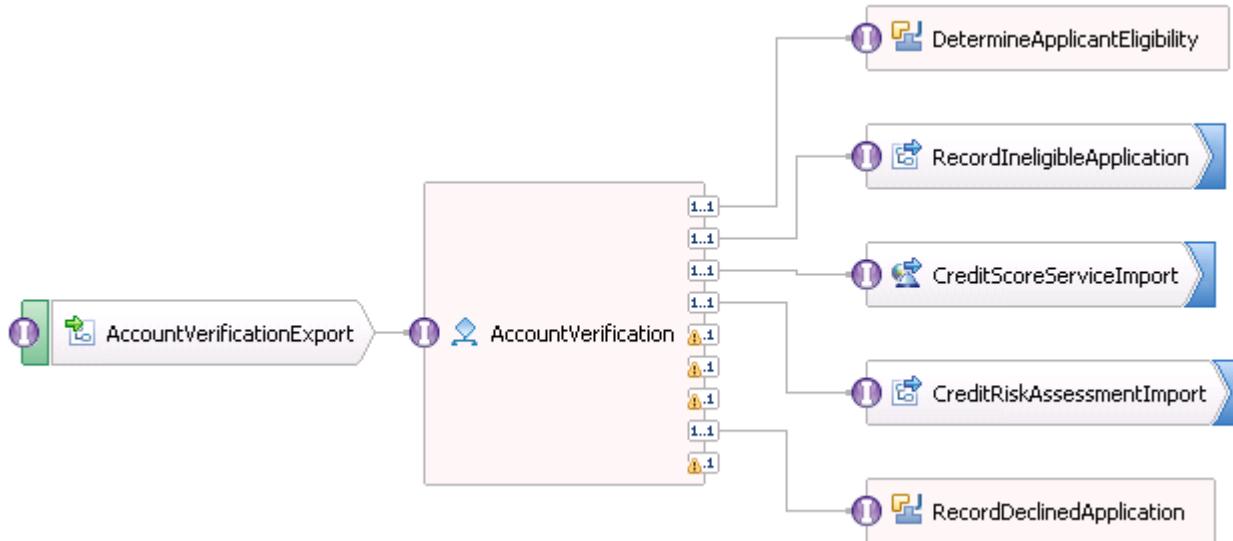


- g. Save your changes.
- 3. Wire the **RecordIneligibleApplicationPartner** reference of the **AccountVerification** process to the **RecordIneligibleApplication** import.
  - a. Right-click the **AccountVerification** process component and click **Wire (Advanced)** from the menu.

- \_\_\_ b. In the Advanced Wiring dialog box, in the “Wire source” window, select **RecordIneligibleApplicationPartner**.
- \_\_\_ c. In the “Wire target” window, select **RecordIneligibleApplication**.



- \_\_\_ d. Click **OK**.
- \_\_\_ e. Save your changes. The **FoundationModule** assembly diagram resembles the following figure:



- \_\_\_ 4. Close IBM Integration Designer.

## End of exercise

## Exercise review and wrap-up

In this exercise, you configured the WebSphere Adapter for Flat Files and tested the adapter in the IBM Integration test environment.



# Exercise 11.Creating mediation services, part I

## What this exercise is about

In this exercise, you create a mediation module and build a simple XML data map.

## What you should be able to do

After completing this exercise, you should be able to:

- Create a mediation module that contains a Mapping primitive
- Define an XML data map
- Test a mediation module that contains a Mapping primitive

## Introduction

Mediation flows intercept and modify messages that are passed between services (providers) and clients (requesters) that want to use those services. Implement mediation flows between services to process the messages that are being passed between them. Mediation flows provide the logic that processes the messages.

For example, mediation flows can be used to find services with specific characteristics that a requester is seeking. It can also be used to resolve interface differences between requesters and providers. For complex interactions, mediation primitives can be linked sequentially. Typical mediations include:

- Transforming a message from one format to another so that the receiving service can accept the message
- Conditionally routing a message to one or more target services; the routing is based on the contents of the message
- Augmenting a message by adding data from a data source

Mediation service applications are assembled and deployed as one or more mediation modules. Mediation modules are deployable units that contain mediation flows.

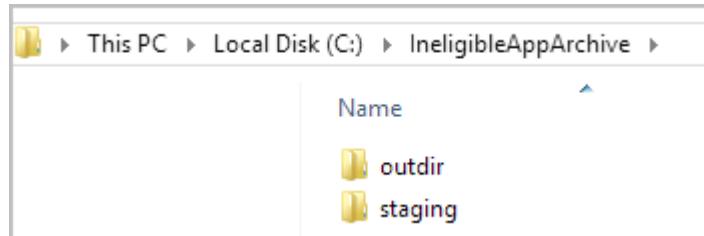
A mediation module can have the following parts:

- Mediation flow components
- Imports that identify service providers and their interfaces
- Exports that expose the mediation module to service requesters
- Java components

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

This exercise stores data in a local directory. They were created in a previous exercise. You must make sure that the directories  
C:\IneligibleAppArchive\outdir and  
C:\IneligibleAppArchive\staging are present. If they are not, you must create them.



## Exercise instructions

In this exercise, you implement a simple mediation module that is named `IneligibleMediationService`, which transforms messages between the `AccountVerification` process and the `RecordIneligibleApplication` service (which archives ineligible applications). The data input for the `RecordIneligibleApplication` service differs from the `CustomerApplication` input that the `AccountVerification` process uses. As such, you must transform the messages that are passed between the services.

You implement a Mapping primitive inside a mediation flow to transform the messages by using a data map. After implementing the transformation in the mediation flow, you assemble the application by wiring the mediation service to the `AccountVerification` process.

### ***Part 1: Create a mediation module that contains a Mapping primitive***

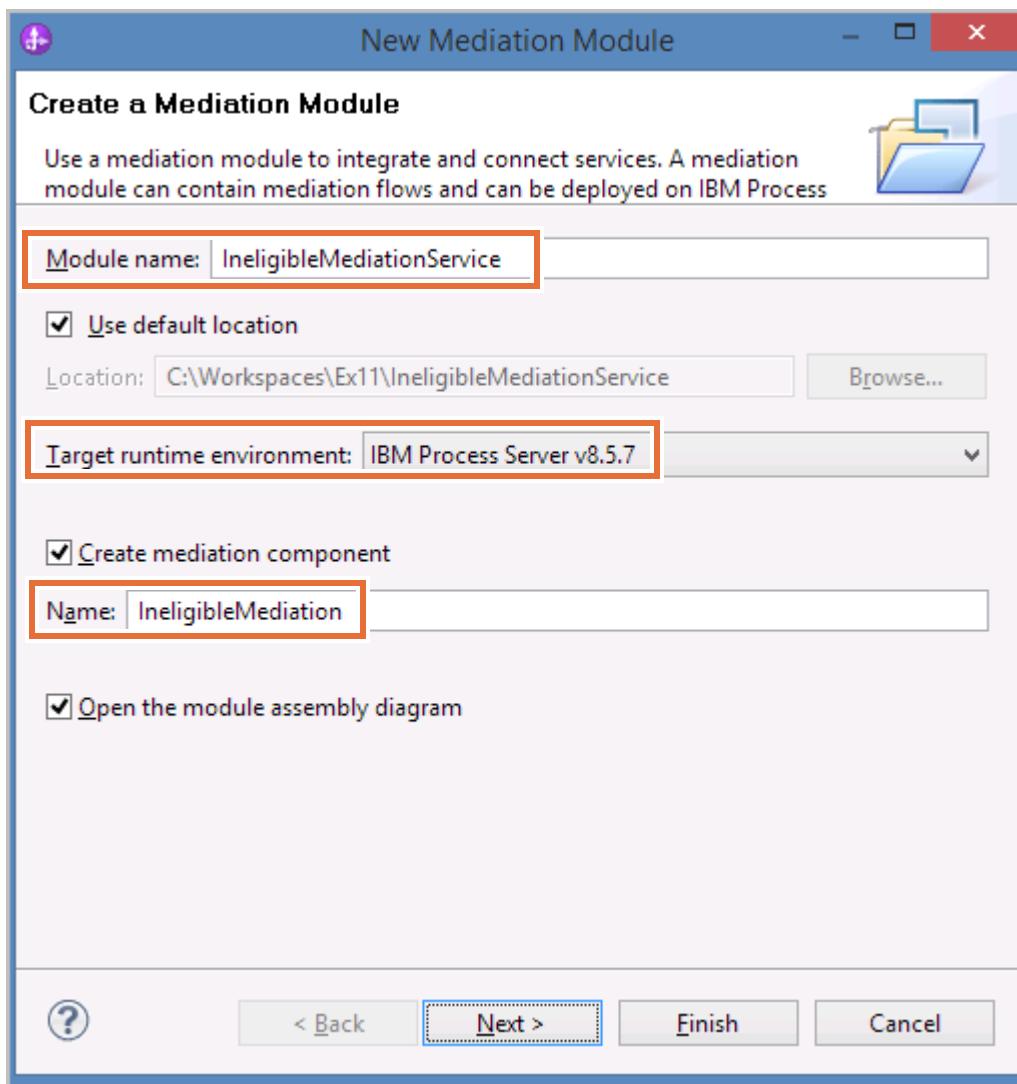
In this portion of the exercise, you create a mediation module that is named `IneligibleMediationService`. This module transforms a `CustomerApplication` into an `IneligibleApplication` before invoking the `RecordIneligibleApplicationExport` component to archive the application. Using a Mapping primitive, you implement an XML data map to transform the data in the body of the service message objects that are passed between the services.

To create the `IneligibleMediationService`:

- 1. Open the Exercise 11 workspace.
  - a. On your desktop, open the **Exercise Shortcuts** folder.
  - b. Double-click the **Exercise 11** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - c. If you get a message that the server is already set to publish, then click **OK**.
  - d. Close the **Getting Started** tab.
- 2. Create a mediation module that is named `IneligibleMediationService`. The target runtime environment for the module is **IBM Process Server v8.5.7**, and the module has a dependency on `FoundationLibrary`. The mediation component in the module is named `IneligibleMediation`.
  - a. Click **File > New > Mediation Module** from the menu options.

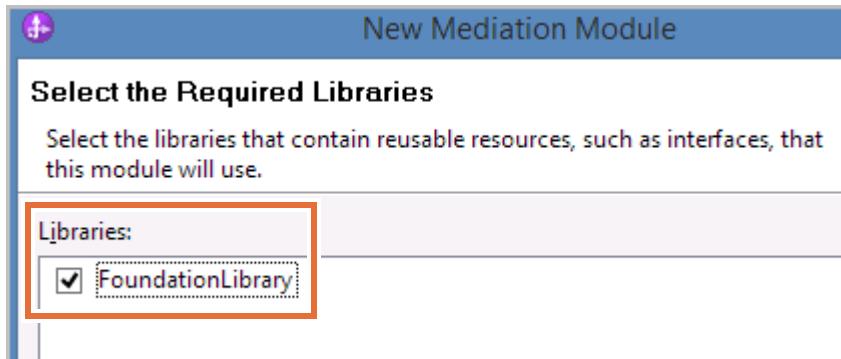
\_\_ b. At the **Create a Mediation Module** pane, enter the following information.

- Type `IneligibleMediationService` in the **Module name** field. This action automatically places `IneligibleMediationService` in the mediation component Name field.
- Change the **Name** of the mediation component from `IneligibleMediationService` to: `IneligibleMediation`
- Select **IBM Process Server v8.5.7** from the **Target runtime environment**.

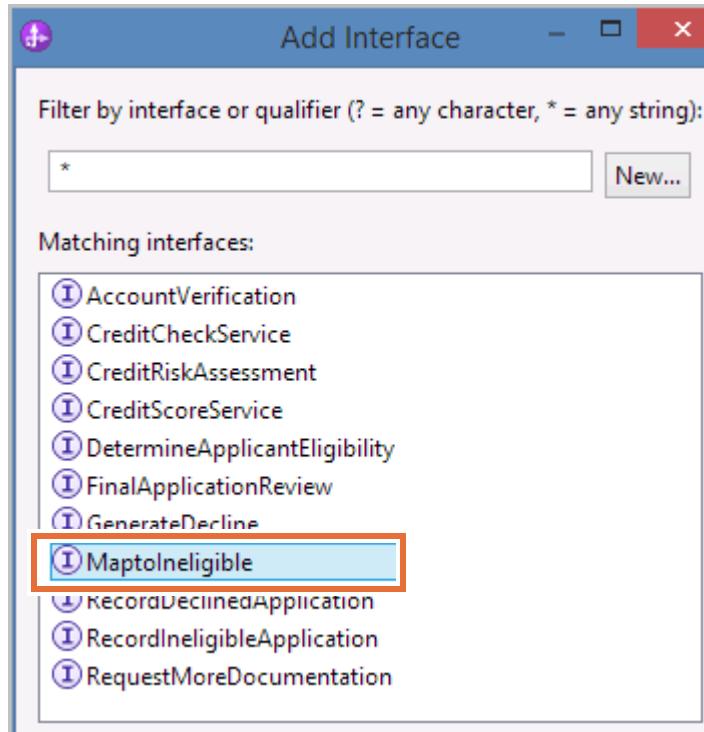


\_\_ c. Accept the remaining default options and click **Next**.

- \_\_\_ d. Select **FoundationLibrary** check box to include it as a dependency.

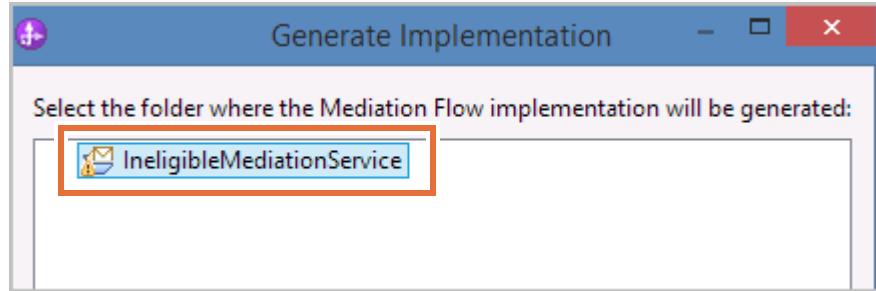


- \_\_\_ e. Click **Finish**.
- \_\_\_ 3. Add the **MaptoIneligible** interface to the **IneligibleMediation** mediation flow component.
- \_\_\_ a. Right-click the **IneligibleMediation** component and click **Add > Interface** from the menu.
- \_\_\_ b. In the Add Interface dialog box, select **MaptoIneligible**.

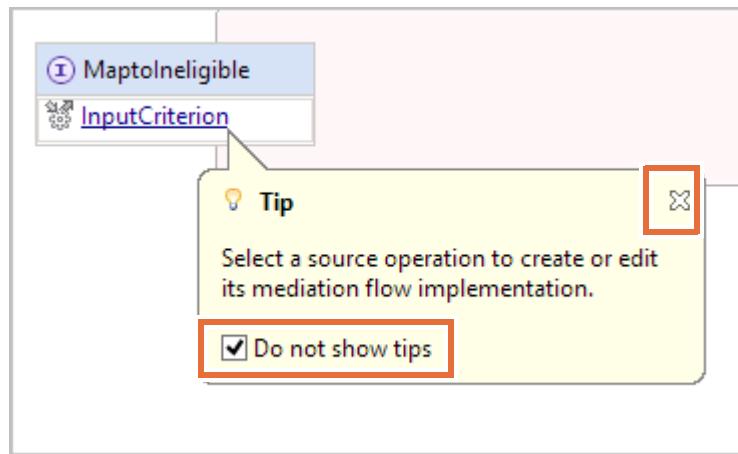


- \_\_\_ c. Click **OK**.
- \_\_\_ 4. Use the **Blank Mediation Flow** template to generate the implementation for the **IneligibleMediation** mediation flow component.
- \_\_\_ a. Right-click the **IneligibleMediation** component and click **Generate Implementation** from the menu.

- \_\_\_ b. In the Generate Implementation dialog box, select the **IneligibleMediationService** folder.



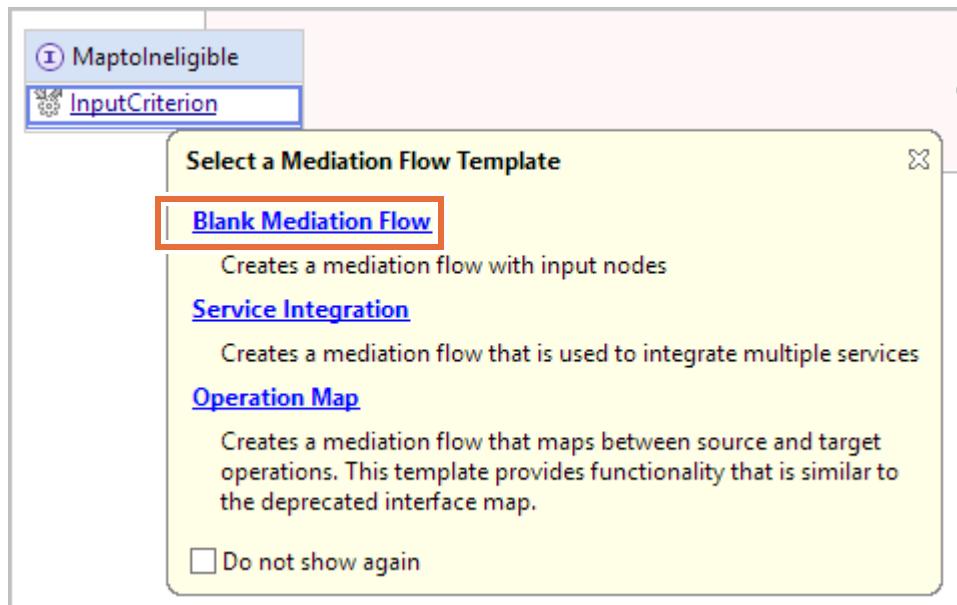
- \_\_\_ c. Click **OK**. The mediation flow editor opens.  
\_\_\_ d. If a Tip dialog box opens in the mediation flow editor, select **Do not show tips** in the dialog box and close it.



## Part 2: Define an XML data map

- \_\_\_ 1. Create an XML data map to transform the input parameters into the output.  
\_\_\_ a. In the Operation connections section of the editor, click the **InputCriterion** operation in the **MaptoIneligible** interface.

- \_\_\_ b. In the “Select a Mediation Flow Template” dialog box, click the **Blank Mediation Flow** link.



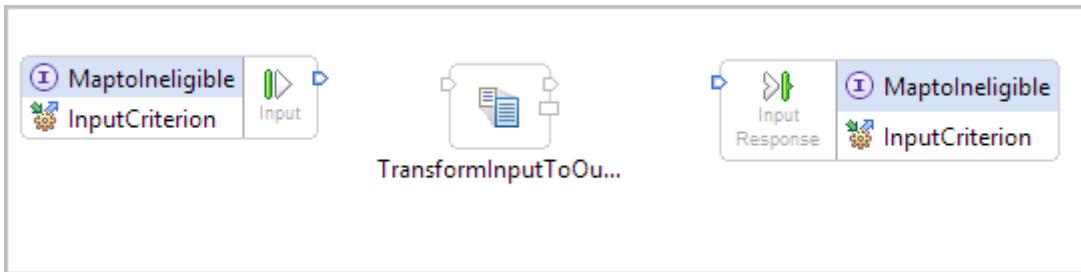
- \_\_\_ c. The generated mediation request flow consists of an **Input** node and an **Input Response** node. You might find it necessary to scroll to the right to see the **Input Response** node.



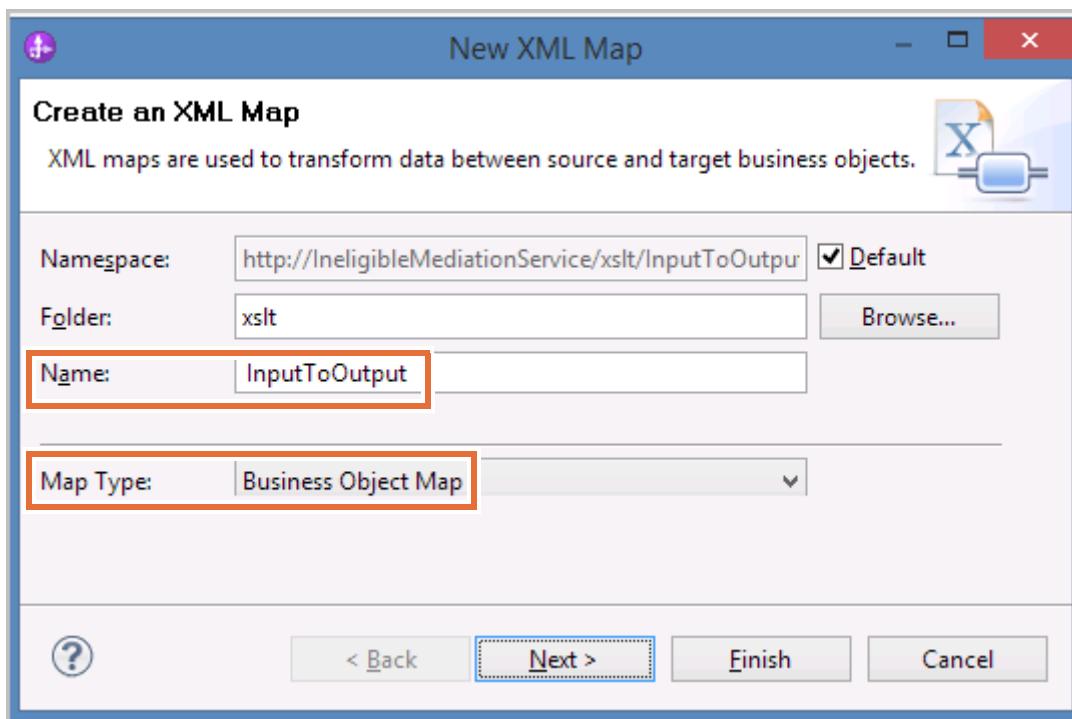
- \_\_\_ 2. Add a Mapping primitive to the request flow that is named: `TransformInputToOutput`
- \_\_\_ a. In the palette, expand **Transformation** and select **Mapping**.
- \_\_\_ b. Click the blank space on the canvas between the Input node and the Input Response node to add the primitive to the diagram.
- \_\_\_ c. Switch to the **Description** tab in the **Properties** view.
- \_\_\_ d. Change the **Display Name** to: `TransformInputToOutput`

| Mapping : TransformInputToOutput |                                                                             |
|----------------------------------|-----------------------------------------------------------------------------|
| <b>Description</b>               | <b>Display name:</b> <code>TransformInputToOutput</code>                    |
| <b>Terminal</b>                  | <b>Name:</b> <code>TransformInputToOutput</code>                            |
| <b>Details</b>                   | <b>Description:</b> Transforms message formats, or changes message content, |
| <b>Promotable Properties</b>     |                                                                             |

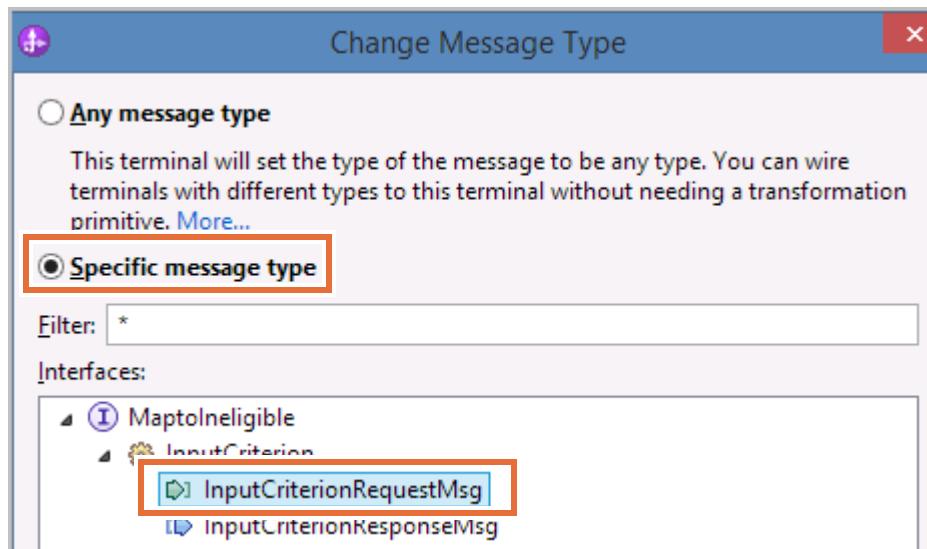
- \_\_ e. Save your changes. The request flow resembles the following figure:



- \_\_ 3. Create the XML data map that the Mapping primitive uses to transform the body of the CustomerApplication input to the body of an IneligibleApplication output. The input message for the map is `InputCriterionRequestMsg`. The output message for the map is `InputCriterionResponseMsg`.
- \_\_ a. With **TransformInputToOutput** selected, switch to the **Details** tab in the **Properties** view.
  - \_\_ b. For the **Mapping File** field, click **New**.
  - \_\_ c. In the “Create an XML Map” pane, enter the following information.
    - Verify that `xslt` is the value in the **Folder** field.
    - Type `InputToOutput` in the **Name** field.
    - Select **Business Object Map** in the **Map Type** field.

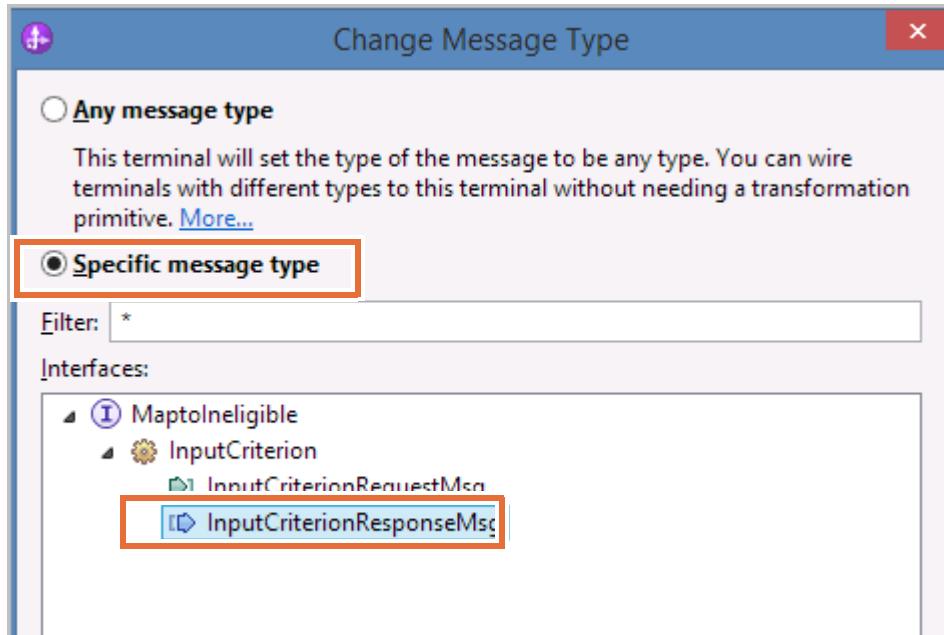


- \_\_\_ d. Before you click **Next**, view the different options available in the **Map Type** field. They are XSLT 1.0 map, XSLT 2.0 map, and Business Object Map. You can use the Mapping mediation primitive to switch between XSL stylesheets and business object maps. When performance is important, you can set the map generation type to generate a business object map instead of an XSLT. Alternatively, you can also generate a business object map instead of an XSLT when the advanced functions that XSLT offers are not required.
- \_\_\_ e. Click **Next**.
- \_\_\_ f. In the Specify Message Types dialog box, take the following actions.
- For **Message Root**, select **/body**.
  - Click **Browse by Input Message Body**, select **Specific message type**, and select **InputCriterionRequestMsg**.

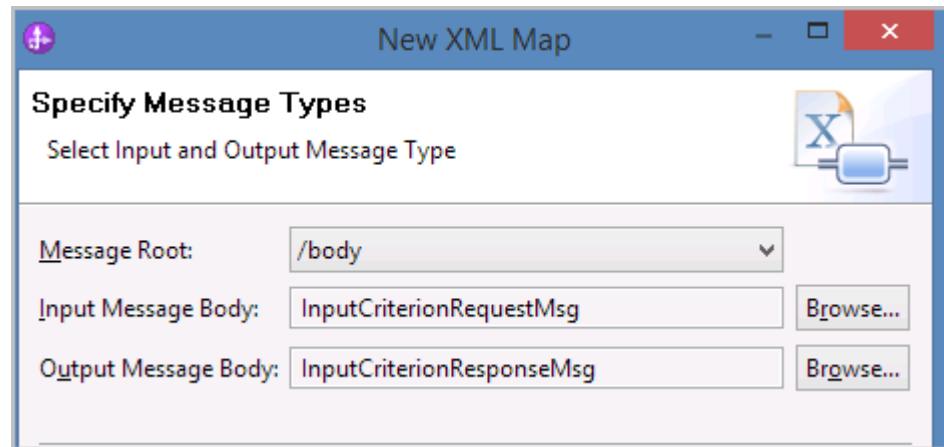


- Click **OK**.

- Click **Browse by Output Message Body**, select **Specific message type**, and select **InputCriterionResponseMsg**.



- Click **OK**.
- The Specify Message Types pane resembles the following figure:



- Ignore any warnings at the bottom of the dialog box and click **Finish** to open the **InputToOutput** map in the mapping editor.

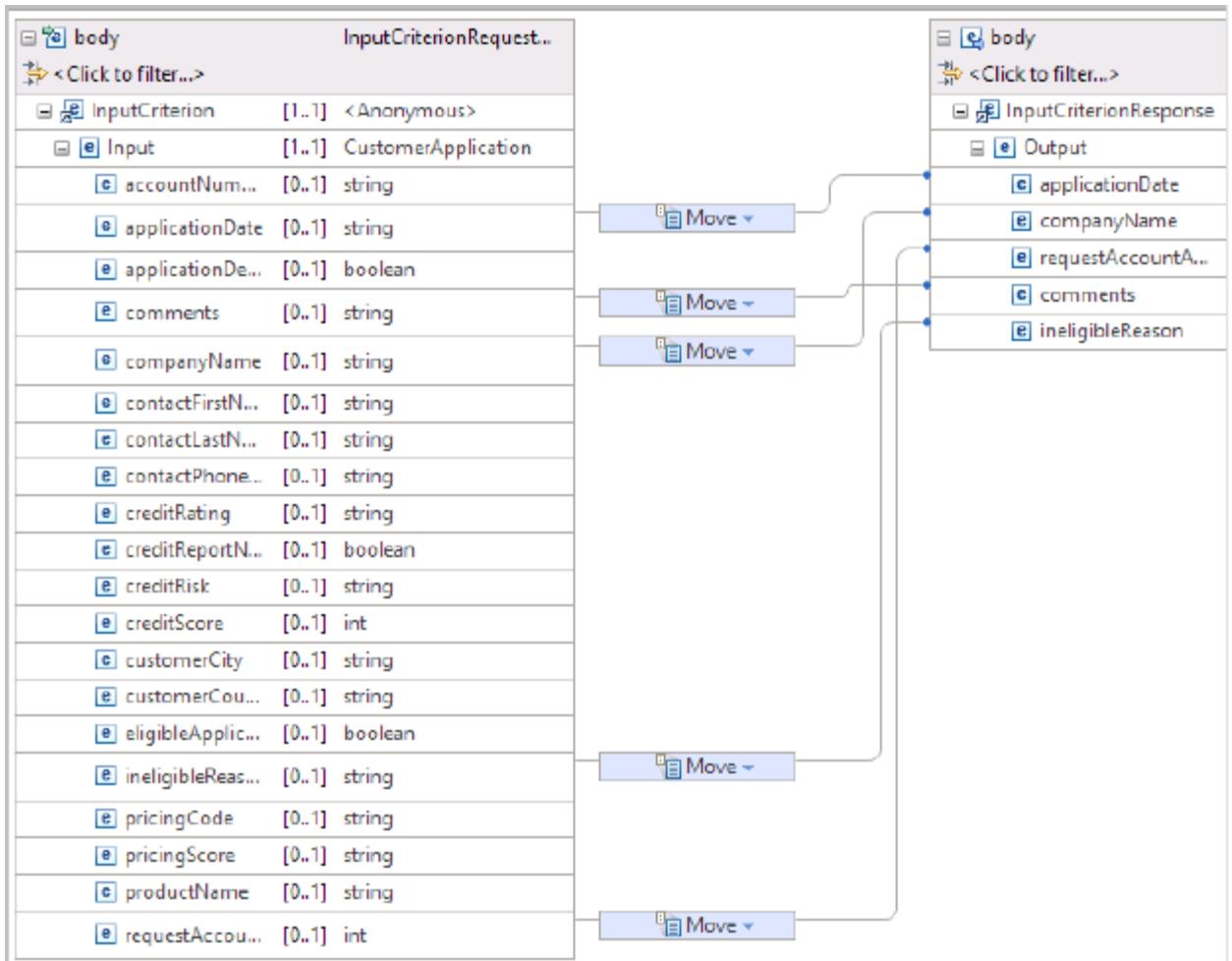
**Note**

The Message Root field identifies the part of the message that you want to map in the XML mapping editor.

- / (root) transforms the entire message and is used to map SOAP attachments
- /body transforms the message body
- /headers transforms the message headers
- /context transforms the message context

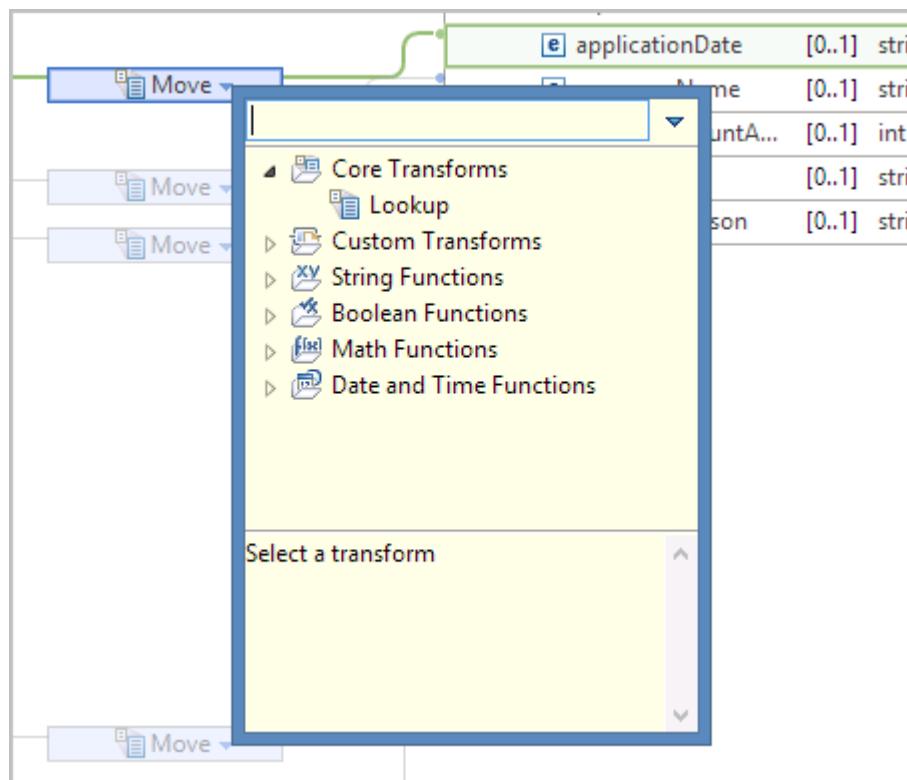
- 4. Implement the map definition. The `applicationDate`, `companyName`, `requestAccountAmount`, `comments`, and `ineligibleReason` elements in the input service message object (SMO) are mapped to the corresponding target elements in the output SMO. All transformations are **Move** transformations.
  - a. In the mapping editor, expand **InputCriterion > Input** in the input SMO by clicking the plus (+) symbols.
  - b. In the mapping editor, expand **InputCriterionResponse > Output** in the output SMO by clicking the plus (+) symbols.
  - c. In the body section of the input SMO, right-click **applicationDate** and click **Create Connection** from the menu.
  - d. Click **applicationDate** in the body section of the output SMO to create the **Move** transformation.
  - e. In the body section of the input SMO, right-click **companyName** and click **Create Connection** from the menu.
  - f. Click **companyName** in the body section of the output SMO to create the **Move** transformation.
  - g. In the body section of the input SMO, right-click **requestAccountAmount** and click **Create Connection** from the menu.
  - h. Click **requestAccountAmount** in the body section of the output SMO to create the **Move** transformation.
  - i. In the body section of the input SMO, right-click **comments** and click **Create Connection** from the menu.
  - j. Click **comments** in the body section of the output SMO to create the **Move** transformation.
  - k. In the body section of the input SMO, right-click **ineligibleReason** and click **Create Connection** from the menu.

- I. Click **ineligibleReason** in the body section of the output SMO to create the **Move** transformation. The completed XML map resembles the following figure:

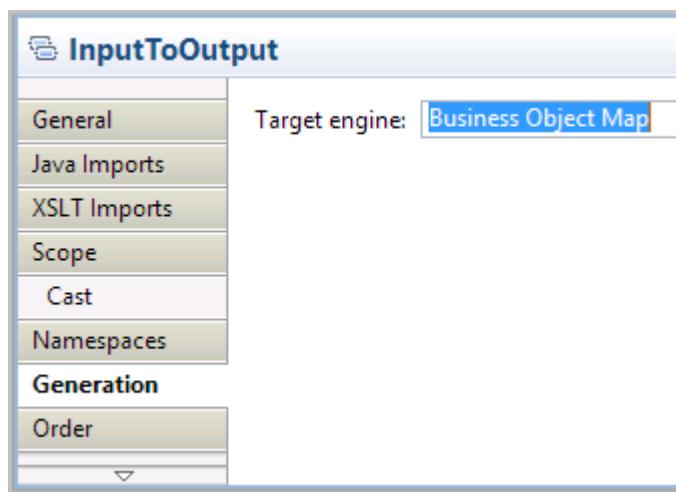


- 5. Click **File > Save All** from the menu options to save your changes to the map, the mediation flow, and the assembly diagram.
- 6. Explore the supported transforms.
- a. Verify that you saved your changes before you do the next few steps, or you might get incorrect results.

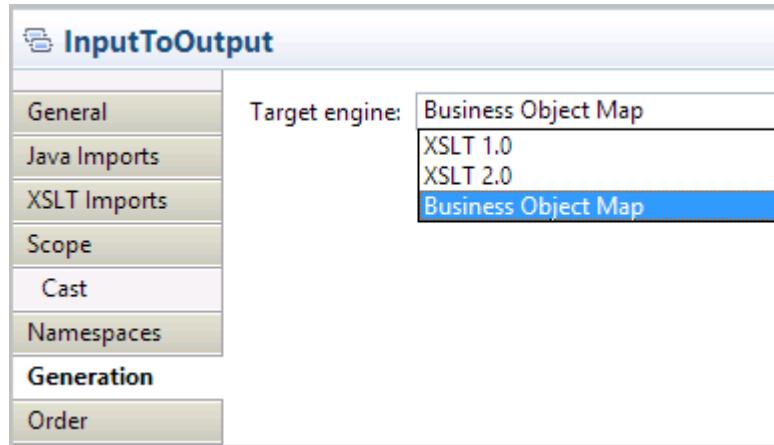
- \_\_\_ b. Click the down arrow in a Move transform to view the available transforms for the business object map. Recall that you set the **map type** to be generated as a **bo map** when you created the Mapping primitive.



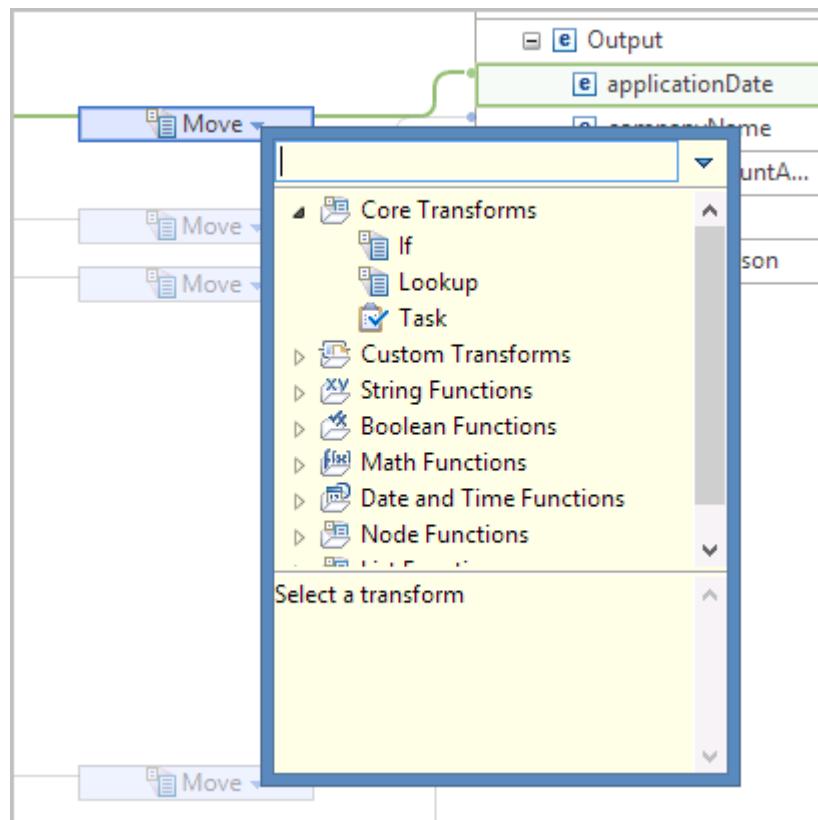
- \_\_\_ c. Click anywhere in the empty space in the XML map editor and verify that **InputToOutput** is listed in the **Properties** view.  
 \_\_\_ d. Click the **Generation** tab. **Business Object Map** is listed as the **Target engine**.



- \_\_ e. Click the **Target engine** list to view the choices available. It offers an option to switch to **XSLT 1.0** or **XSLT 2.0** as the map type.

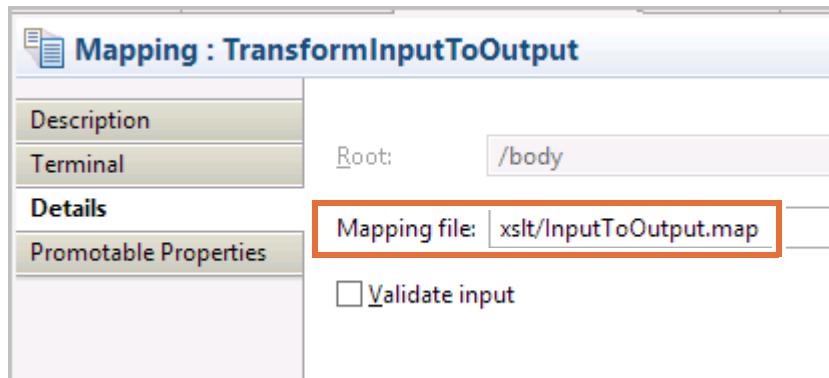


- \_\_ f. Select **XSLT 1.0** as the **Target engine** type.
- \_\_ g. Click the down arrow in a **Move** transform to view the available transforms. This time, the list of the transforms changes to support the XSLT map generation. Also, notice that the transforms that a business object map supports are a subset of transforms that XSLT supports. The tool allows an easy switch between the business object map and the XSLT target engine when mapping requirements change. Validations are done for unsupported transforms during the switch.

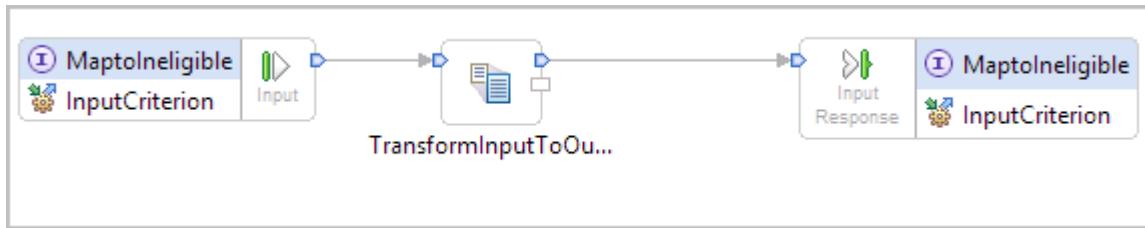


- \_\_ 7. Close the map editor and do not save your changes.

8. Verify the definition of the Mapping primitive.
- \_\_ a. With **TransformInputToOutput** selected in the request flow editor, switch to the **Details** tab in the **Properties** view.
  - \_\_ b. Verify that the **Mapping file** is set to: `xslt/InputToOutput.map`

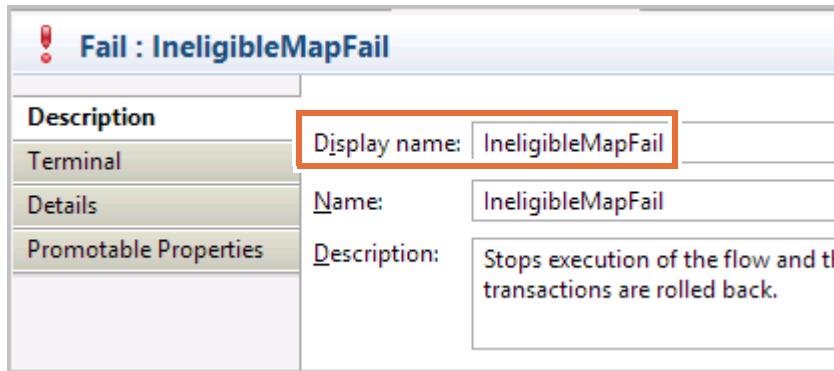


9. Assemble the request flow by wiring the Input and Input Response nodes to the Mapping primitive. Wire the **out** terminal of the **Input** node to the **in** terminal of the **TransformInputToOutput** primitive. Wire the **out** terminal of the **TransformInputToOutput** primitive to the **in** terminal of the **Input Response** node.
- \_\_ a. Right-click the **out** terminal of the **InputCriterion** input node and click **Add Connection** from the menu. If you find it difficult to right-click the **out** terminal, you can hover over the terminal and grab the orange, circular handle.
  - \_\_ b. Click the **in** terminal of the **TransformInputToOutput** primitive to add the connection.
  - \_\_ c. Right-click the **out** terminal of the **TransformInputToOutput** primitive and click **Add Connection** from the menu.
  - \_\_ d. Click the **in** terminal of the **InputCriterion** input response node to add the connection.
  - \_\_ e. Save the **IneligibleMediation** flow. Your diagram resembles the following figure:

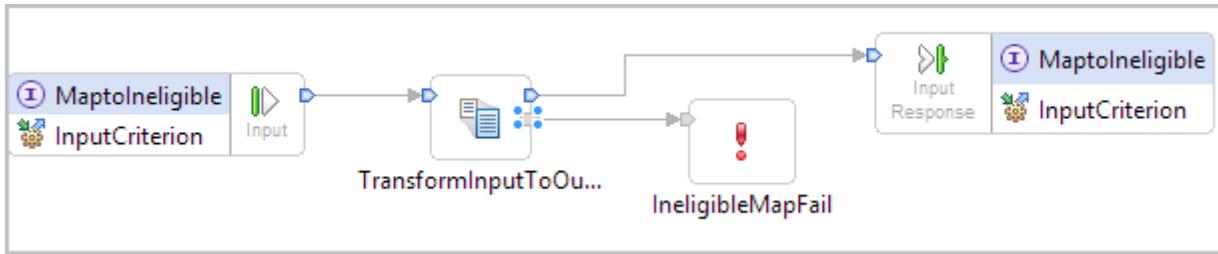


10. Add a **Fail** primitive to the request flow named: `IneligibleMapFail`  
Wire the **fail** terminal of the **TransformInputToOutput** primitive to the **in** terminal of the **Fail** primitive. The fail primitive stops flow execution and throws an exception when a failure is encountered in the map.
- \_\_ a. In the palette, expand **Error Handling** and select **Fail**.
  - \_\_ b. Click an empty portion of the diagram to the right of the **TransformInputToOutput** primitive to add the fail primitive to the flow.
  - \_\_ c. With the **Fail** primitive selected, switch to the **Description** tab in the **Properties** view.

- \_\_ d. Change the **Display name** to: IneligibleMapFail



- \_\_ e. Save your changes.
- \_\_ f. Right-click the **fail** terminal on the **TransformInputToOutput** primitive and click **Add Connection** from the menu.
- \_\_ g. Click the **in** terminal on the **IneligibleMapFail** primitive to add the connection.
- \_\_ h. Save your changes. The request flow diagram resembles the following figure:



- \_\_ i. Close the **IneligibleMediation** tab to return to the assembly diagram. You do not implement a response flow for this mediation because the flat file adapter processes communications in only one direction: outbound.
- \_\_ 11. Create an export with SCA binding for the **IneligibleMediation** component.
- \_\_ a. In the assembly diagram, right-click **IneligibleMediation** and click **Generate Export > SCA Binding** from the menu.
- \_\_ b. Accept the default export name: **IneligibleMediationExport**
- \_\_ c. Save your changes. Your assembly diagram resembles the following figure:



- \_\_ 12. Close the assembly diagram and make sure that you have no errors in the Problems view. You can ignore any warnings.

### Part 3: Test a mediation module that contains a Mapping primitive

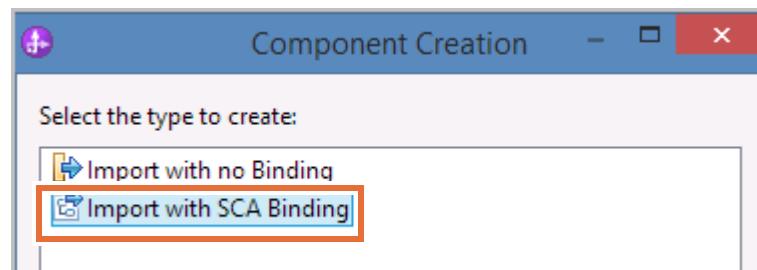
In this portion of the exercise, you wire the IneligibleMediationService to the AccountVerification business process on the FoundationModule assembly diagram. The RecordIneligibleApplication service can be called to archive an ineligible application by using the flat file adapter. However, before that call is made, the CustomerApplication data object used by the AccountVerification business process must be transformed.

The IneligibleMediationService transforms a CustomerApplication data object into an IneligibleApplication data object that is used by the RecordIneligibleApplication service. The IneligibleMediationService mediation module uses a Mapping primitive (and an XML data map) to transform the data.

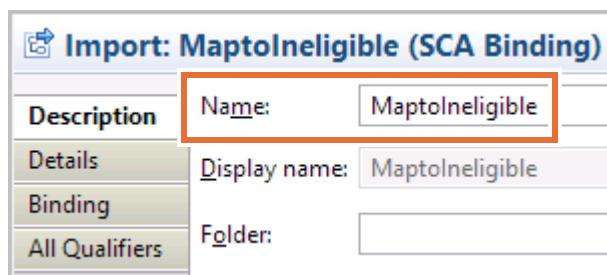
### Assembling the applications

To wire the IneligibleMediationService to the AccountVerification business process:

- 1. Create an import component that is named `MaptoIneligible` on the **FoundationModule** assembly diagram. The `MaptoIneligible` import invokes the `IneligibleMediationExport` component in the `IneligibleMediationService` mediation module.
  - a. In the Business Integration view, expand **FoundationModule** and double-click **Assembly Diagram**.
  - b. In the Business Integration view, expand **IneligibleMediationService > Assembly Diagram**.
  - c. Drag **IneligibleMediationExport** onto the **FoundationModule** assembly diagram.
  - d. Select **Import with SCA Binding** and click **OK**.

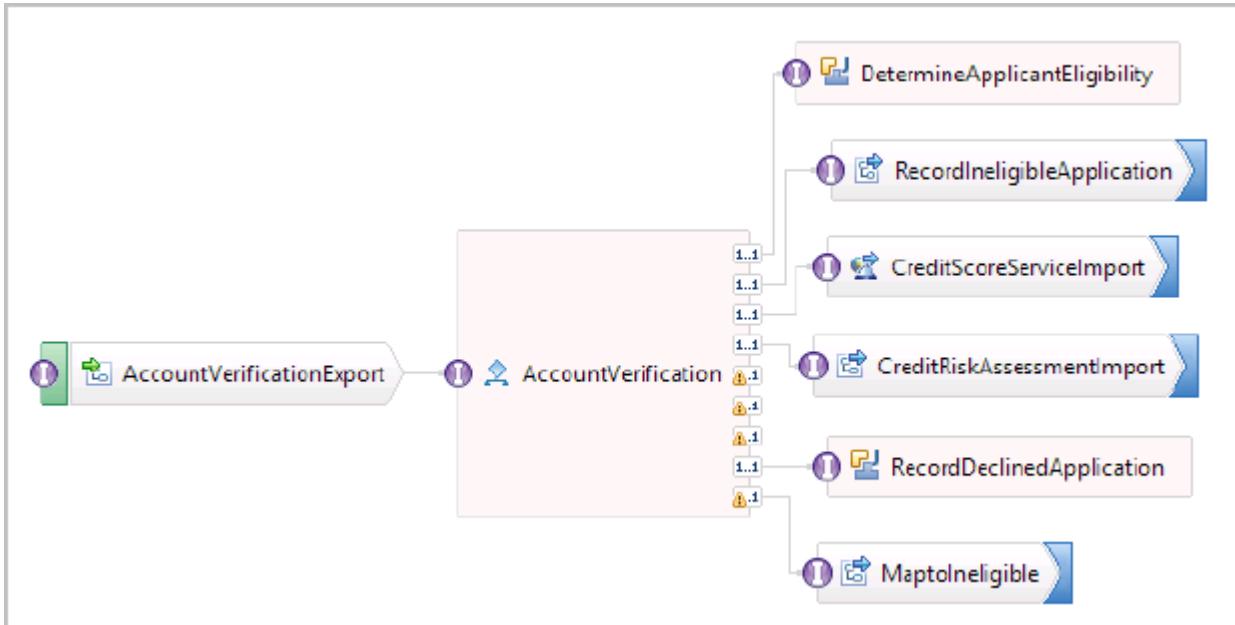


- e. Select the import and switch to the **Description** tab in the **Properties** view.
- f. Change the Name to: `MaptoIneligible`



- g. Save your changes.

2. Wire the **MaptoIneligiblePartner** reference on the AccountVerification process component to the newly created **MaptoIneligible** import.
- Right-click the **MaptoIneligible** import component and click **Wire to Existing** from the menu.
  - Save your changes.
  - Verify the wiring by hovering over the reference that is wired to the **MaptoIneligible** import. The dialog box displays the **MaptoIneligiblePartner** reference.
  - Your **FoundationModule** assembly diagram resembles the following visual:



## Testing the applications

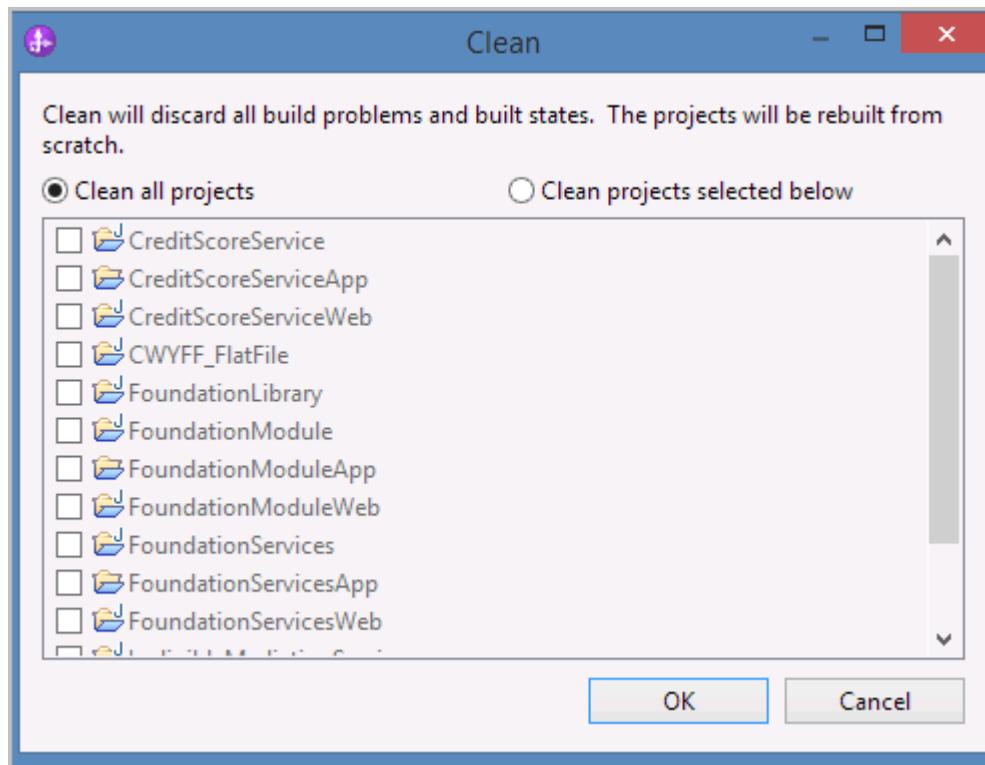
Now that you completed the IneligibleMediationService mediation module implementation, the “ineligible” path through the AccountVerification business process is complete. If an application is determined to be ineligible for an account, the eligibleApplication field is set to `false`.

When an application is ineligible, the IneligibleMediationService is called to transform the CustomerApplication into an IneligibleApplication by using the data map you implemented earlier. The map transforms the data in the body of the service message objects when they are passed between the services. After transformation, the IneligibleApplication is sent to the RecordIneligibleApplication service for archiving by the flat file adapter.

To test the **IneligibleMediationService**:

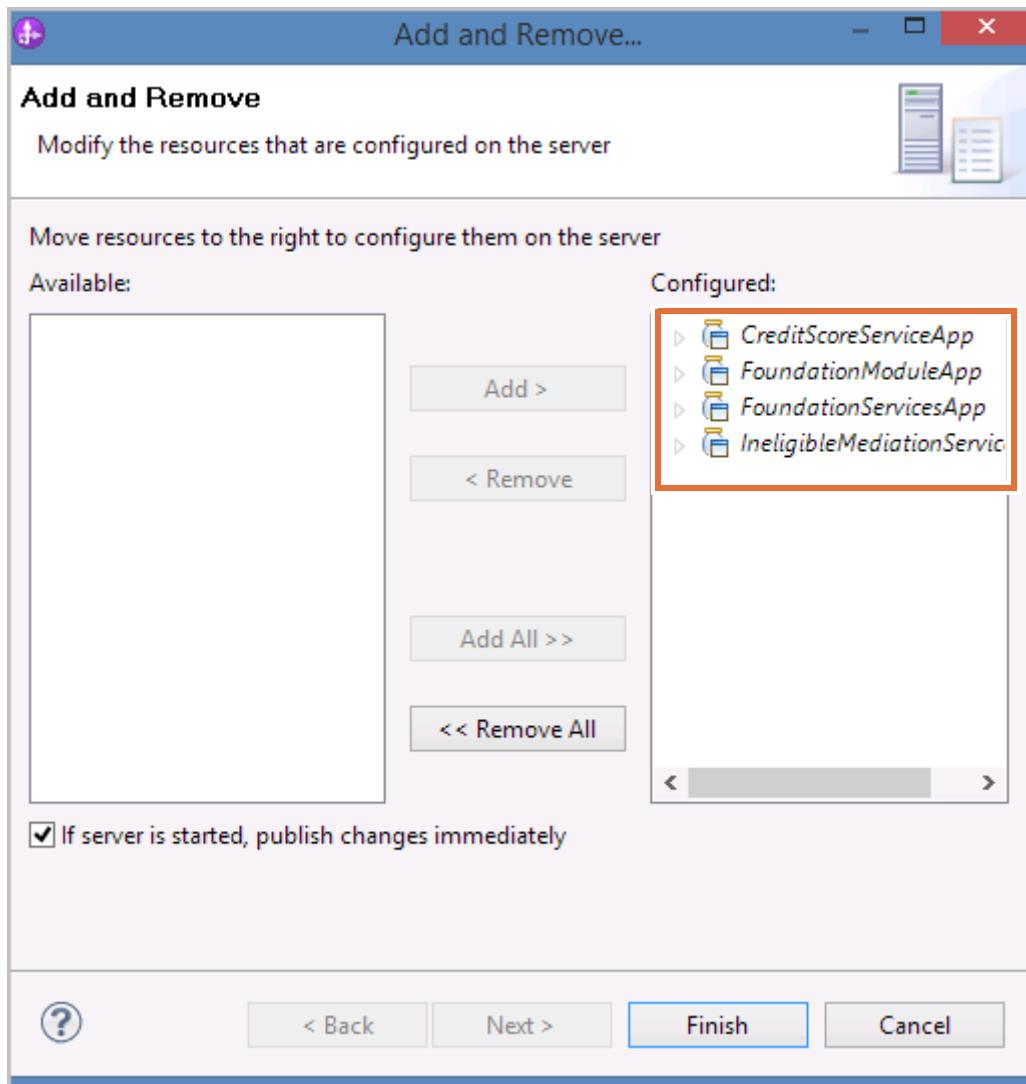
- Clean the project. Before testing, it is a good idea to clean the project to make sure that no errors remain.
  - Click **Project > Clean** from the menu.

- \_\_\_ b. Verify that **Clean all projects** is selected and click **OK**. Wait a few minutes until the workspace is built before continuing to the next step.



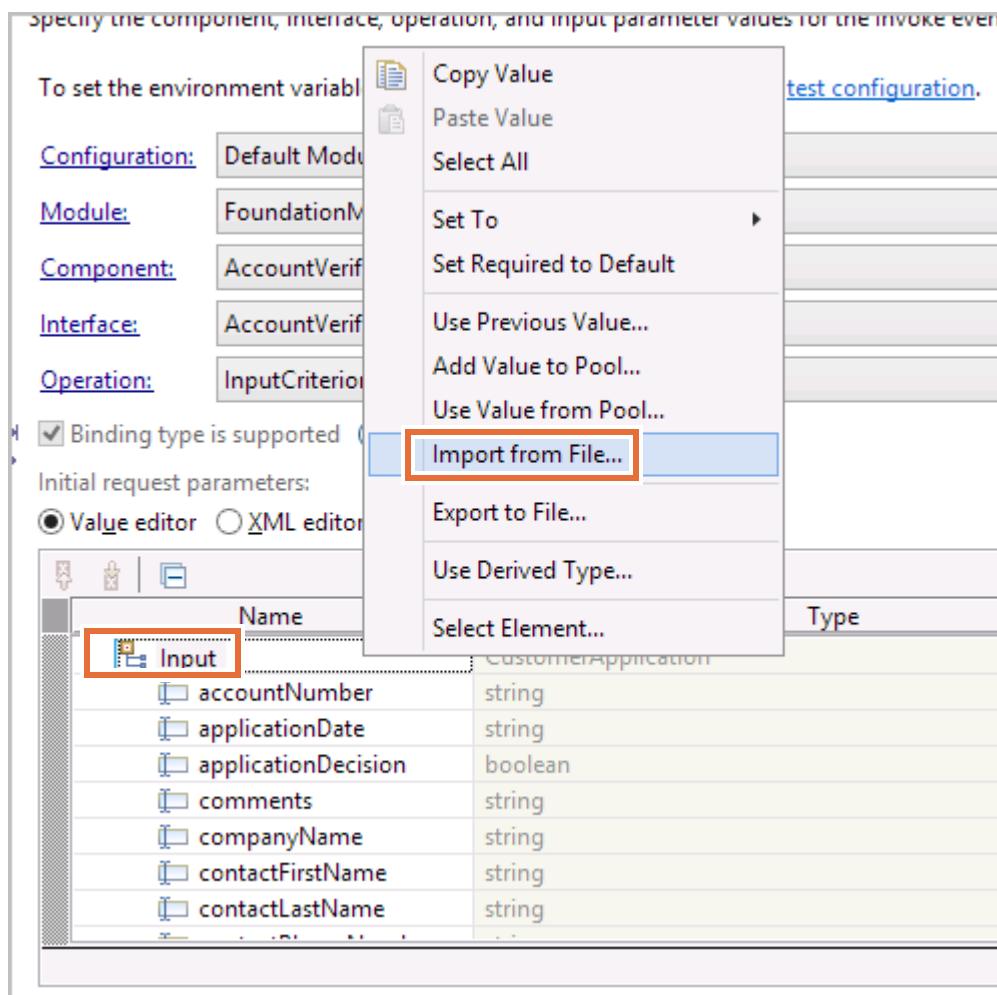
- \_\_\_ 2. Start the server (if it is not already running) and deploy IneligibleMediationServiceApp, FoundationServicesApp, CreditScoreServiceApp, and FoundationModuleApp.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** (if necessary). Wait for the startup process to complete before continuing.  
The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
- \_\_\_ b. In the Servers view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.

- c. Click **Add All** to add the projects to the **Configured** list. You can also double-click projects individually to add them to the **Configured** projects list.



- d. Click **Finish**. It takes a few minutes for the applications to publish and start. If you receive an error in publishing, then try restarting the individual applications.
- e. When the applications are deployed and started, the messages Application started: CreditScoreServiceApp, Application started: IneligibleMediationServiceApp, Application started: FoundationServicesApp, and Application started: FoundationModuleApp are displayed in the **Server Logs** view.
- f. If any module has a **Stopped** status, then right-click the module and click **Restart** from the menu. If prompted to do so, republish the module. Wait for the server status to change to **Started, Synchronized**. If the server has a status of **Started, Publishing**, then clicking the server refreshes the status to **Started, Synchronized**.
- 3. On the FoundationModule assembly diagram, right-click the **AccountVerificationExport** component and click **Test Component** from the menu.

4. Use C:\labfiles\Support Files\Ex11\EX11\_Test\_Data.xml to load the Exercise 11 test data. Because you are testing the “ineligible” application path, you use the test data that is associated with company AbcCo. The test data for AbcCo automatically sets the **eligibleApplication** field to `false`.
- a. When the integrated test client opens in the **Initial request parameters** section, right-click **Input** and click **Import from File** from the menu.



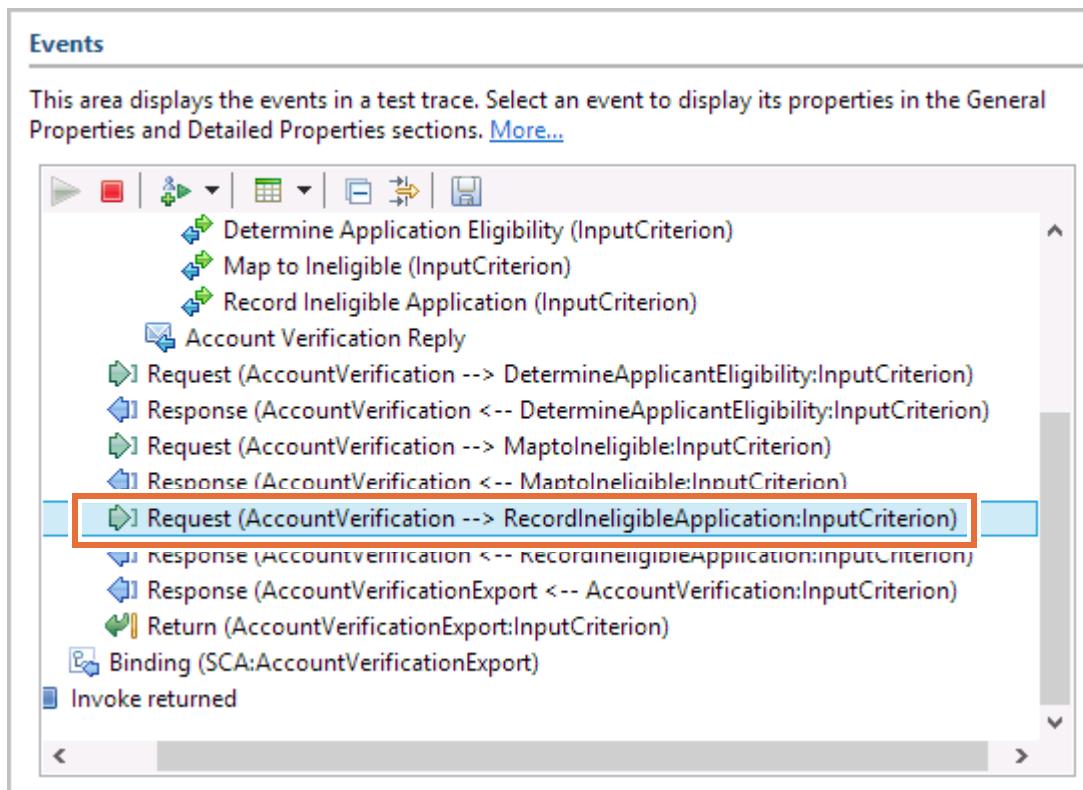
- \_\_ b. Browse to C:\labfiles\Support Files\Ex11\, select **EX11\_Test\_Data.xml**, and click **Open** to populate the input parameters with the required test data.

| Initial request parameters |                     |                   |
|----------------------------|---------------------|-------------------|
|                            | Type                |                   |
| Go to Previous Error       | CustomerApplication | [ab]              |
| Input                      | CustomerApplication | [ab]              |
| accountNumber              | string              | [ab] ABC001       |
| applicationDate            | string              | [ab] 06/10/2014   |
| applicationDecision        | boolean             | [ab] true         |
| comments                   | string              | [ab] Bad credit   |
| companyName                | string              | [ab] AbcCo        |
| contactFirstName           | string              | [ab] Fernando     |
| contactLastName            | string              | [ab] Torres       |
| contactPhoneNumber         | string              | [ab] 315-555-9725 |
| creditRating               | string              | [ab] F            |
| creditReportNeeded         | boolean             | [ab] true         |
| creditRisk                 | string              | [ab] HIGH         |
| creditScore                | int                 | [ab] 1            |
| customerCity               | string              | [ab] Madrid       |
| customerCountry            | string              | [ab] Spain        |
| eligibleApplication        | boolean             | [ab] false        |
| ineligibleReason           | string              | [ab] Bad credit   |
| pricingCode                | string              | [ab] 51           |
| pricingScore               | string              | [ab] 21           |
| productName                | string              | [ab] Tacks        |
| requestAccountAmount       | int                 | [ab] 20000        |

- 
- \_\_\_ c. Alternatively, you can manually enter the following data (maximizing the initial request parameters window assists you with data entry).
- accountNumber: ABC001
  - applicationDate: 06/10/2014
  - applicationDecision: true
  - comments: Bad credit
  - companyName: AbcCo
  - contactFirstName: Fernando
  - contactLastName: Torres
  - contactPhoneNumber: 315-555-9725
  - creditRating: F
  - creditReportNeeded: true
  - creditRisk: HIGH
  - creditScore: 1
  - customerCity: Madrid
  - customerCountry: Spain
  - eligibleApplication: false
  - ineligibleReason: Bad credit
  - pricingCode: 51
  - pricingScore: 21
  - productName: Tacks
  - requestAccountAmount: 20000
- \_\_\_ 5. Click the **Continue** icon to run the test.
- \_\_\_ 6. If the “Select a Deployment Location” dialog box is displayed, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_\_ 7. In the User Login dialog box, accept the default entries for **User ID** (admin) and **Password** (web1sphere) and click **OK**.
- \_\_\_ 8. Wait for the test to complete. When you receive the blue, square stop node in the **Events** window, the test is finished. Depending on resources, it might take a few minutes for the test to complete.

9. Examine the test trace.

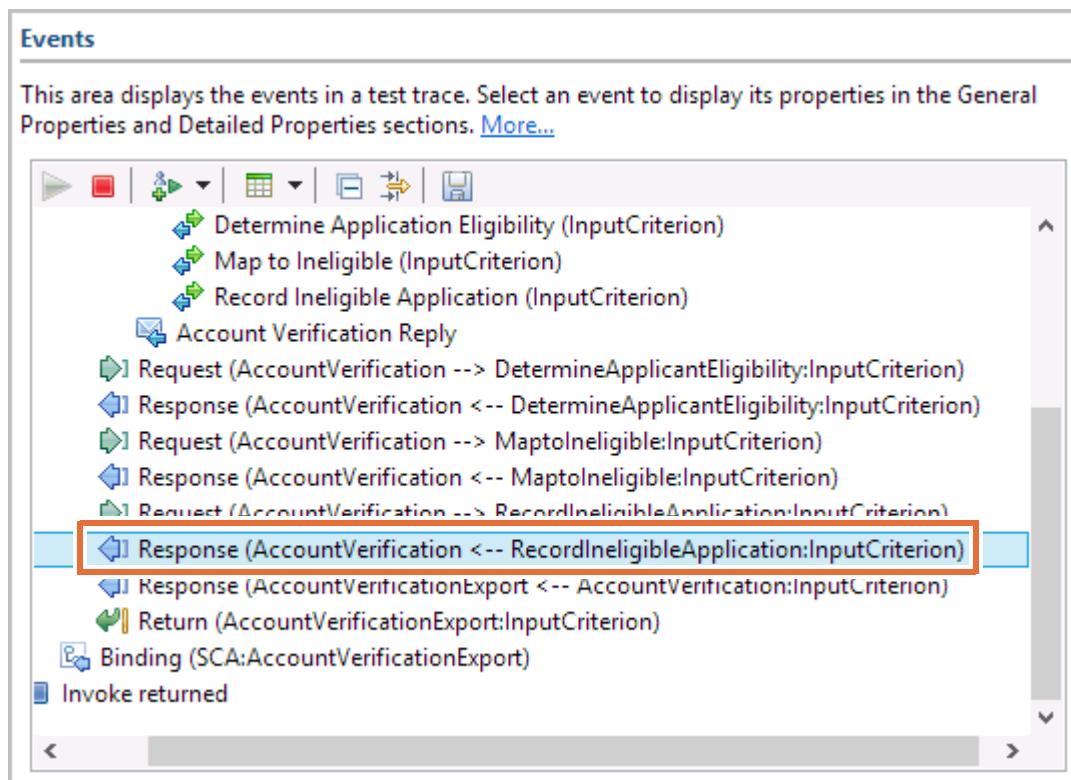
- a. Click the **Request (AccountVerification > RecordIneligibleApplication:InputCriterion)** event.



- b. Examine the **Request parameters** section. The business object was correctly transformed from CustomerApplication to IneligibleApplication.

| Name                 | Type                  | Value        |
|----------------------|-----------------------|--------------|
| Input                | IneligibleApplication |              |
| applicationDate      | string                | May 16, 2016 |
| companyName          | string                | AbcCo        |
| requestAccountAmount | int                   | 20000        |
| comments             | string                | Bad credit   |

- \_\_\_ c. Select the **Response (AccountVerification <-- RecordIneligibleApplication:InputCriterion)** event.

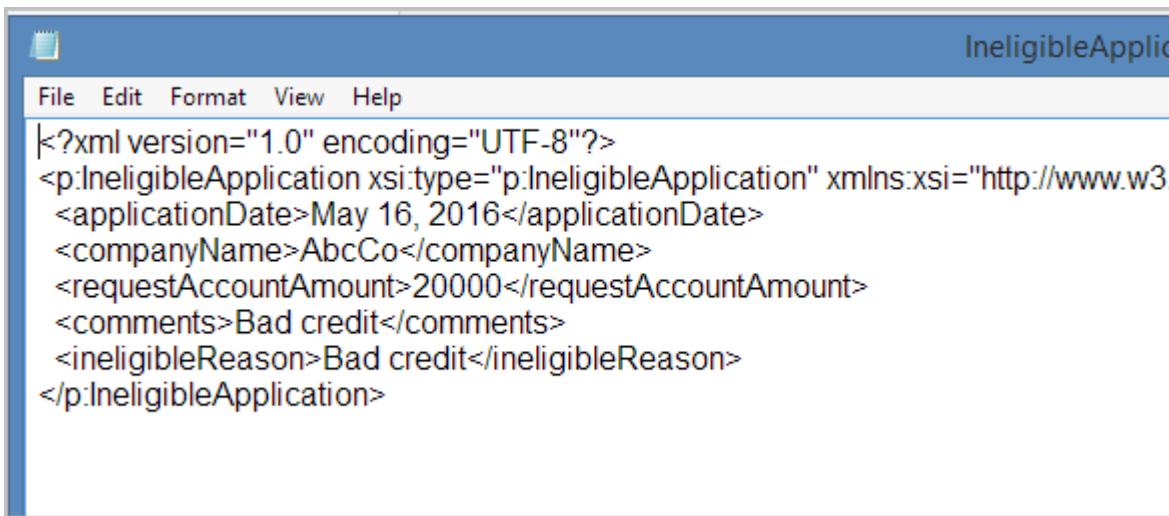


- \_\_\_ d. Examine the **Response parameters** section. Note the response message that indicates that the IBM WebSphere Adapter for Flat Files successfully archived the ineligible application. You can resize the window and expand the table to view the full text.

| Name    | Type    | Value                                                                                  |
|---------|---------|----------------------------------------------------------------------------------------|
| Output  | Message | ...                                                                                    |
| message | string  | Account Verification recorded this application as ineligible for the customer AbcCo... |

- \_\_\_ e. Open Windows Explorer and browse to `C:\IneligibleAppArchive\outdir`.

- \_\_\_ f. Open the file with the highest number at the end, for example, IneligibleApplication.3.txt. The test data corresponding to company AbcCo is displayed.



The screenshot shows a Windows Explorer window with a single file listed: "IneligibleApplication.3.txt". The file is an XML document with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<p:IneligibleApplication xsi:type="p:IneligibleApplication" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <applicationDate>May 16, 2016</applicationDate>
 <companyName>AbcCo</companyName>
 <requestAccountAmount>20000</requestAccountAmount>
 <comments>Bad credit</comments>
 <ineligibleReason>Bad credit</ineligibleReason>
</p:IneligibleApplication>
```

- \_\_\_ g. Close the ineligible application file and close Windows Explorer.
- \_\_\_ 10. Close the **FoundationModule\_test** tab and click **No** when you are prompted to save the test trace.
- \_\_\_ 11. Remove the applications from the server and stop the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
- \_\_\_ b. Click **Remove All** and click **Finish**.
- \_\_\_ c. Optionally, stop the server.
- \_\_\_ 12. Close IBM Integration Designer.

## End of exercise

## **Exercise review and wrap-up**

In this exercise, you defined an XML map and created a mediation module.



# Exercise 12.Creating mediation services, part II

## What this exercise is about

In this exercise, you use a message filter primitive to implement a mediation module that routes messages.

## What you should be able to do

After completing this exercise, you should be able to:

- Create a mediation module that contains a message filter mediation primitive and a Mapping primitive
- Define an XML data map
- Test a mediation module that contains a message filter mediation primitive and a Mapping primitive

## Introduction

Mediation primitives are the building blocks of mediation flows in business integration modules and mediation modules. Mediation flows operate on messages that are in-flight between service requesters and service providers. You can use each mediation primitive to do different things with a message. Mediation primitives process messages as service message objects (SMOs) because SMOs allow different types of messages to be processed in a common way.

For example, different messages can use the message filter mediation primitive to take different paths. A message might need forwarding to different service providers based on the request details. You can also use the message content as the basis for bypassing unnecessary steps. If the criterion is not met, you can use the fail mediation primitive to raise a fault, or you can send an error response.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

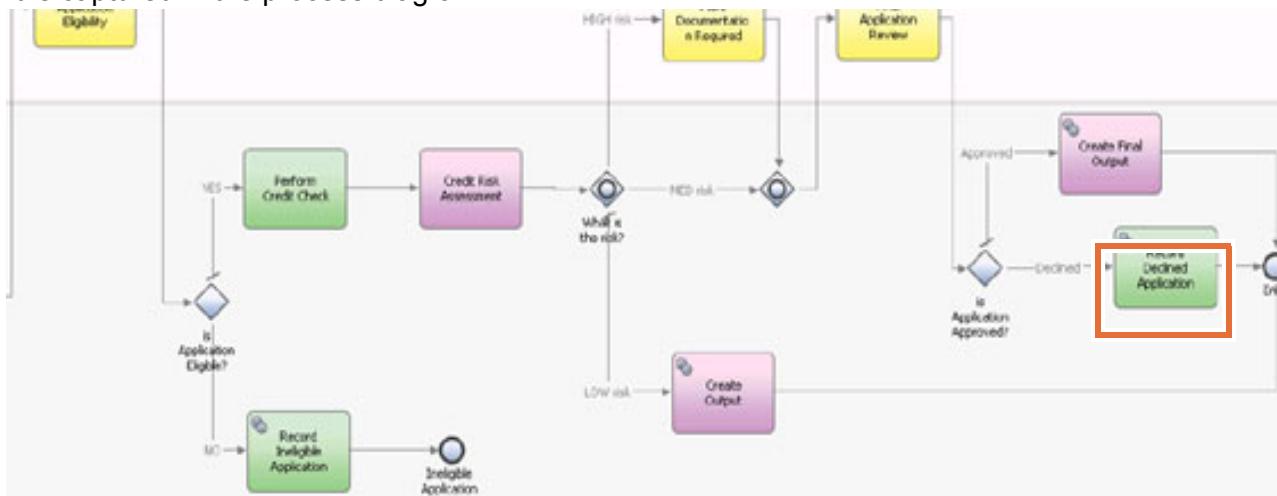
## Exercise instructions

In this exercise, you implement a simple mediation module that is named `RouterMediationService` that is designed to route messages between services. A message filter primitive is used to route the messages. When you implement the mediation service, you wire it to the `AccountVerification` process component on the `FoundationModule` assembly diagram.

Remember in the account verification process narrative, a customer application can be declined. The approval and decline human task user interface is created in another exercise. If the customer's creditRisk is `MED` (short for medium), the application is routed directly to the `FinalApplicationReview` activity. If the customer's creditRisk is `HIGH`, more documentation is requested before the application is routed to the `FinalApplicationReview` activity.

If the employee approves the application, the process completes successfully. However, if the application is declined, it is routed to one of two possible services. If the application is declined and the creditRisk is `HIGH`, the generate decline service is invoked. If the application is declined and the creditRisk is `MED`, the special decline service is invoked. The implementation of both services uses simple Java code to return messages to the console that indicates which "declined" service was called. These services would normally be implemented as adapters or other types of services that would archive the applications for auditing or send them for more review.

It is captured in the process diagram:



### Part 1: Create a mediation module that contains a message filter mediation primitive and a Mapping primitive

In this portion of the exercise, you generate the implementation for the "generate decline" service. If the application is declined during `FinalApplicationReview` and the customer's creditRisk is `HIGH`, the "generate decline" service is called.

To implement the "generate decline" service:

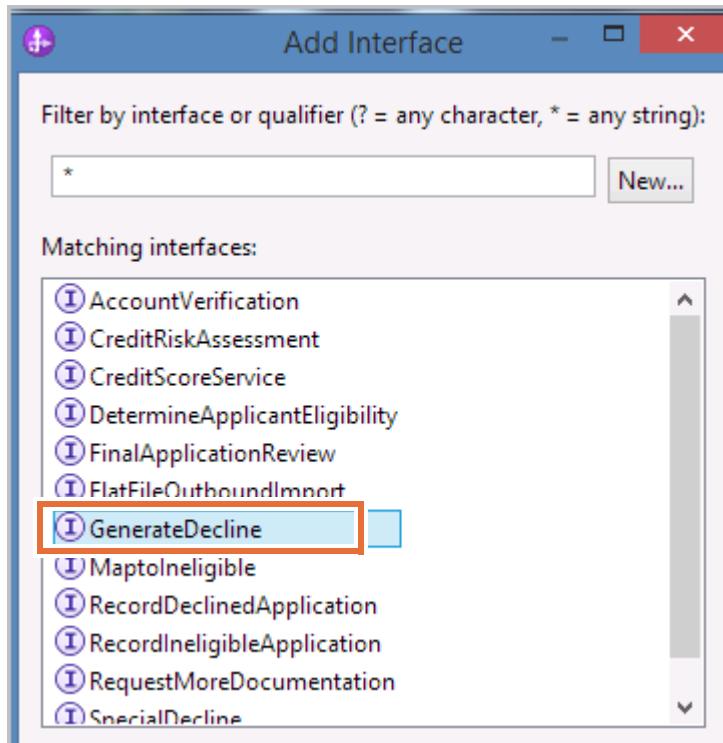
- \_\_ 1. Open the Exercise 12 workspace.
- \_\_ a. On your desktop, open the **Exercise Shortcuts** folder.

- \_\_\_ b. Double-click the **Exercise 12** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - \_\_\_ c. If a message that the server is already set to publish is displayed, click **OK**.
  - \_\_\_ d. Close the **Getting Started** tab.
- \_\_\_ 2. Create a Java component that is named `GenerateDecline` on the **FoundationServices** assembly diagram.
- \_\_\_ a. In the Business Integration view, expand **FoundationServices** and double-click **Assembly Diagram**.
  - \_\_\_ b. In the palette, expand **Components** and select **Java**.
  - \_\_\_ c. Click any blank space on the assembly diagram to add the Java component.
  - \_\_\_ d. Switch to the **Description** tab in the **Properties** view.
  - \_\_\_ e. Change the **Name** of the Java component to: `GenerateDecline`

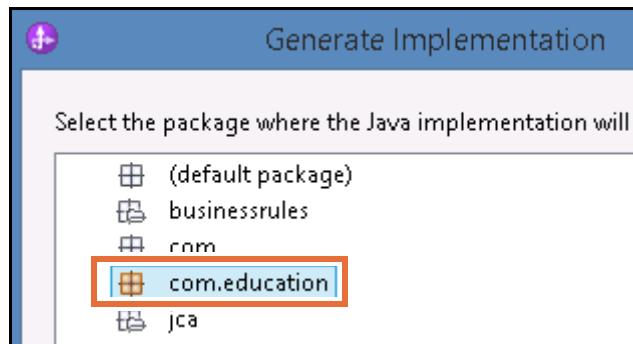


- \_\_\_ f. Save your changes.
- \_\_\_ 3. Add the `GenerateDecline` interface to the component.
- \_\_\_ a. Right-click the **GenerateDecline** Java component and click **Add > Interface** from the menu.

- \_\_ b. In the Add Interface dialog box, select **GenerateDecline**.



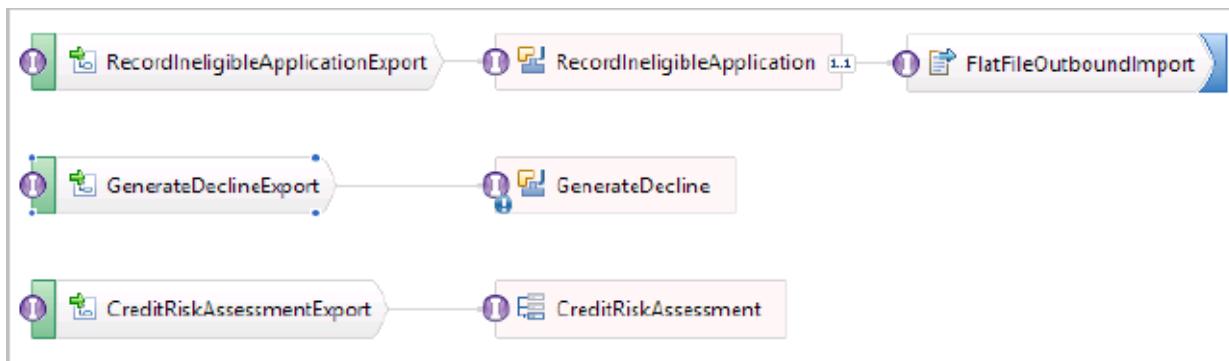
- \_\_ c. Click **OK**.
- 4. Use the code in `C:\labfiles\Support Files\Ex12\GenerateDecline_InputCriterion.txt` to create the implementation for the Java component. The code returns a `Message` business object that contains the message: "Account for customer <company name> was declined and the credit risk was <credit risk>."
- \_\_ a. Right-click **GenerateDecline** and click **Generate Implementation** from the menu.
- \_\_ b. In the Generate Implementation dialog box, select **com.education**.



- \_\_ c. Click **OK** to open `GenerateDeclineImpl.java` in the Java editor.
- \_\_ d. Open Windows Explorer and browse to `C:\labfiles\Support Files\Ex12`.
- \_\_ e. Open `GenerateDecline_InputCriterion.txt` in a text editor such as Notepad.
- \_\_ f. Copy the text that is in the file.

- \_\_\_ g. On the **GenerateDeclineImpl.java** tab, scroll to the `public DataObject InputCriterion` method at the end of the file.
  - \_\_\_ h. Replace the comment lines (the green text that begins with `//`) with the code from `GenerateDecline_InputCriterion.txt`. Be sure to remove the `return null;` line from the method.
  - \_\_\_ i. Alternatively, enter the following code:

```
System.out.println("[Java] Generate Decline - begins");
String ret = "Account for customer " + input.getString("companyName") + " was declined and the credit risk was " + input.getString("creditRisk");
System.out.println("[Java] Generate Decline - " + ret);
DataObject response =
com.ibm.websphere.sca.sdo.DataFactory.INSTANCE.create("http://FoundationLibrary/businessitems", "Message");
response.setString("message", ret);
System.out.println("[Java] Generate Decline - ends");
return response;
```
  - \_\_\_ j. Press **Ctrl+S** to save your changes. The left margin of the Java editor contains no error markers.
  - \_\_\_ k. Close the Java editor.
  - \_\_\_ l. Close the `GenerateDecline_InputCriterion.txt` file, but leave Windows Explorer open.
- \_\_\_ 5. Generate an export component for `GenerateDecline` on the FoundationServices assembly diagram that is named `GenerateDeclineExport`. The export has an SCA binding.
- \_\_\_ a. On the **FoundationServices** assembly diagram, right-click **GenerateDecline** and click **Generate Export > SCA Binding** from the menu.
  - \_\_\_ b. Accept the default export name: `GenerateDeclineExport`
  - \_\_\_ c. Save your changes. Your assembly diagram resembles the following figure:

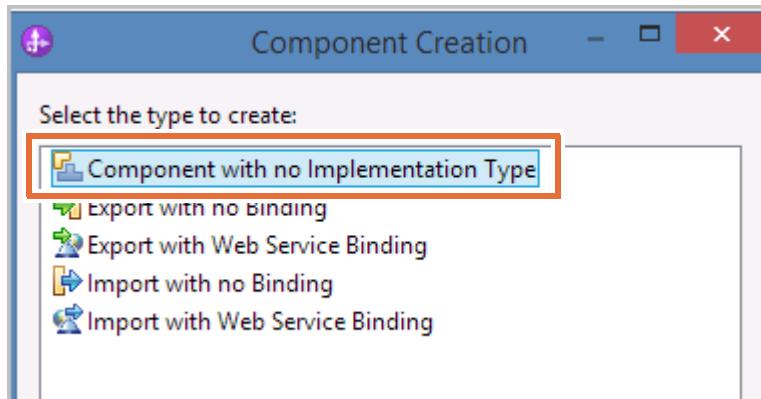


## Implementing the special decline service

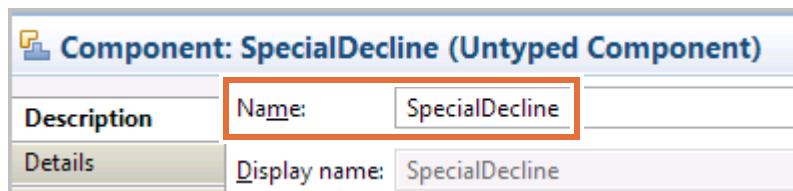
In this portion of the exercise, you generate the implementation for the “special decline” service. The “special decline” service is called when the application is declined during FinalApplicationReview and the creditRisk is MED.

To implement the “special decline” service:

- \_\_ 1. On the **FoundationServices** assembly diagram, create an untyped component that is named `SpecialDecline` that uses the **SpecialDecline** interface. An untyped component can be used as a placeholder for future development. In this case, you generate the Java implementation immediately after creating the component. This exercise is done for educational purposes. You can also use a Java component instead of an untyped component.
  - \_\_ a. In the Business Integration view, expand **FoundationLibrary > Interfaces**.
  - \_\_ b. Drag the **SpecialDecline** interface onto the **FoundationServices** assembly diagram.
  - \_\_ c. In the Component Creation dialog box, select **Component with no Implementation Type**.

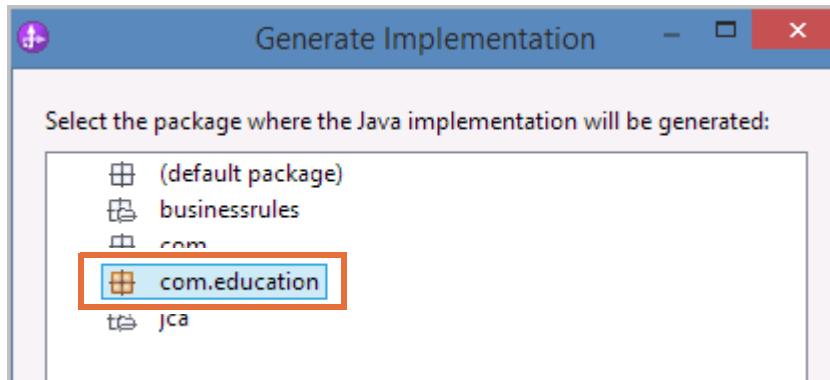


- \_\_ d. Click **OK**.
- \_\_ e. Switch to the **Description** tab in the **Properties** view.
- \_\_ f. Change the **Name** of the component to: `SpecialDecline`



- \_\_ g. Save your changes.

2. Use the code in C:\labfiles\Support Files\Ex12\SpecialDecline\_InputCriterion.txt to generate the Java implementation for the SpecialDecline component. The code in SpecialDecline returns a Message business object that contains the message: "Account for customer <company name> was routed through special decline because the credit risk was <credit risk>."
- a. On the FoundationServices assembly diagram, right-click **SpecialDecline** and click **Generate Implementation > Java** from the menu.
  - b. In the **Generate Implementation** dialog box, select the **com.education** package.

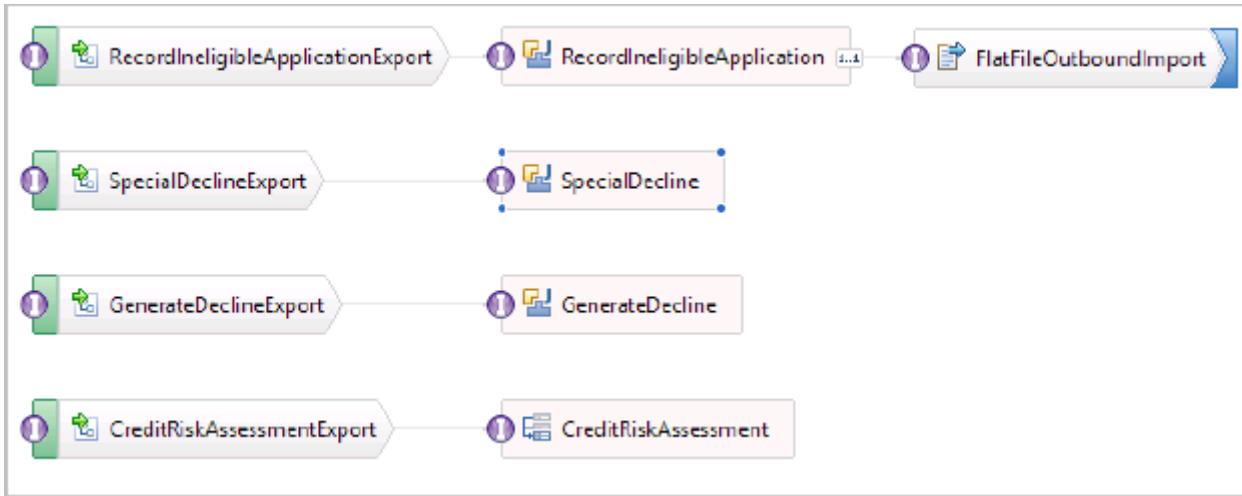


- c. Click **OK**. The `SpecialDeclineImpl.java` file opens in the Java editor.
- d. In Windows Explorer, open C:\labfiles\Support Files\Ex12\SpecialDecline\_InputCriterion.txt in a text editor such as Notepad.
- e. Copy the text from `SpecialDecline_InputCriterion.txt`.
- f. On the **SpecialDeclineImpl.java** tab, scroll to the `public DataObject InputCriterion` method at the end of the file.
- g. Paste the code from `SpecialDecline_InputCriterion.txt` over the green comment lines (that begin with `//`). Be sure to remove `return null;` from the method.
- h. Alternatively, enter the following code:

```
public DataObject InputCriterion(DataObject input) {
 System.out.println("[Java] Generate Decline Special - begins");
 String ret = "Account for customer "
 + input.getString("companyName") + " was routed through special decline
 because the credit risk was " + input.getString("creditRisk");
 System.out.println("[Java] Generate Decline Special - " + ret);
 DataObject response =
 com.ibm.websphere.sca.sdo.DataFactory.INSTANCE.create(
 "http://FoundationLibrary/businessitems", "Message");
 response.setString("message", ret);
 System.out.println("[Java] Generate Decline Special - ends");
 return response;
}
```

- i. Click **File > Save All** from the menu options.
- j. Close the Java editor.

- \_\_\_ k. Close `SpecialDecline_InputCriterion.txt` and close Windows Explorer.
- \_\_\_ 3. Generate an export component for SpecialDecline on the FoundationServices assembly diagram that is named `SpecialDeclineExport`. The export has an SCA binding.
  - \_\_\_ a. On the **FoundationServices** assembly diagram, right-click **SpecialDecline** and click **Generate Export > SCA Binding** from the menu.
  - \_\_\_ b. Accept the default export name: `SpecialDeclineExport`
  - \_\_\_ c. Save your changes. Your assembly diagram resembles the following figure:



## Implementing the RouteRequest mediation flow component

If `applicationDecision` is set to `false` during `FinalApplicationReview` (the application is declined) and the customer's `creditRisk` is `HIGH`, the application is routed through the "generate decline" component. If `applicationDecision` is set to `false` during `FinalApplicationReview` and the customer's `creditRisk` is `MED` (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the **RouteRequest** component. The **RouteRequest** flow component contains the mediation logic that routes the application to the appropriate decline service.

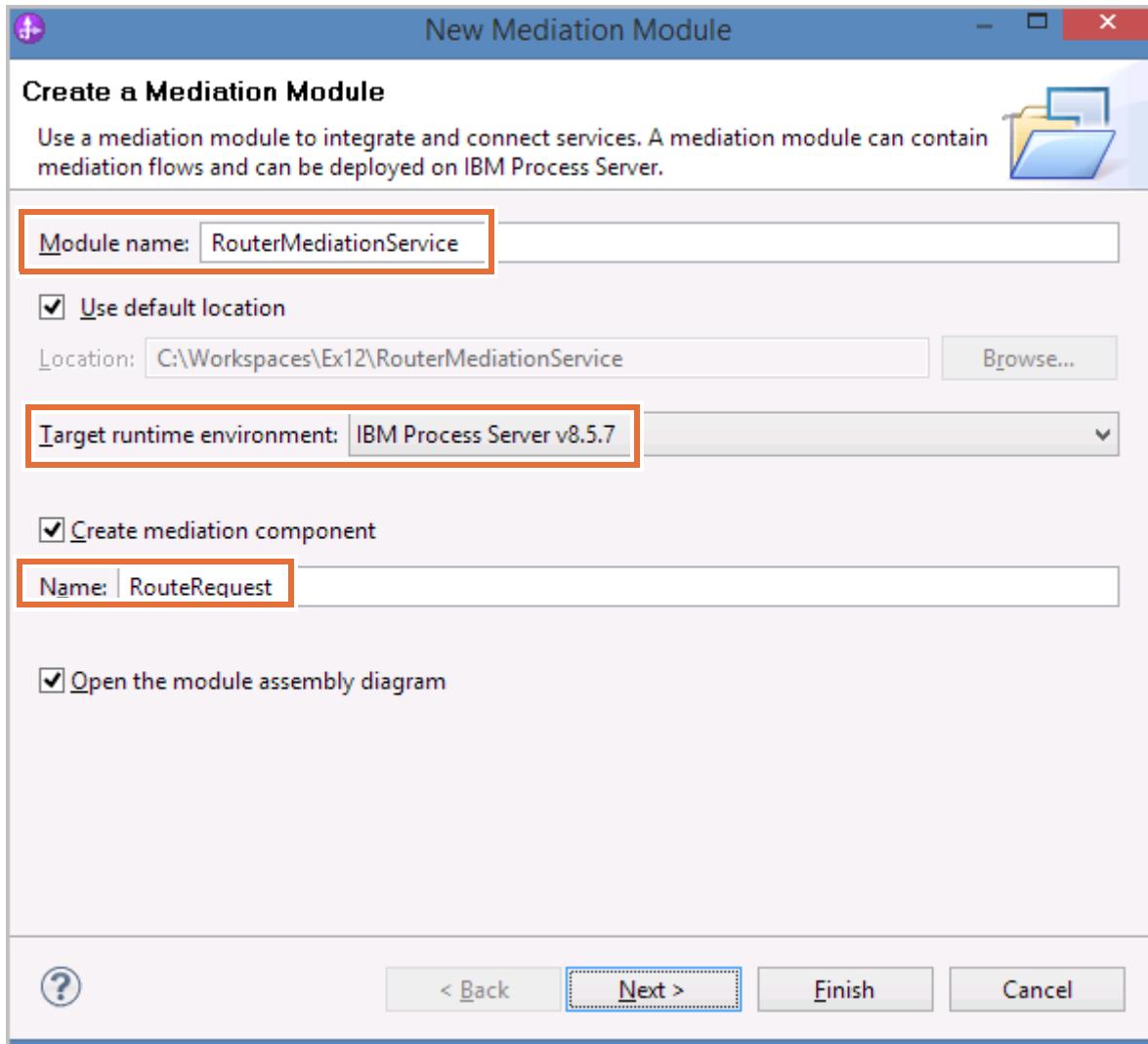
The **RouteRequest** mediation flow consists of both a request flow and a response flow. In the flow, the `CustomerApplication` is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the `AccountVerification` process.

To implement the **RouteRequest** mediation flow component:

- \_\_\_ 1. Create a mediation module that is named `RouterMediationService`. The target runtime environment for the module is **IBM Process Server v8.5.7** and the module has a dependency on **FoundationLibrary**. The mediation component in the module is named `RouteRequest`.
  - \_\_\_ a. Click **File > New > Mediation Module** from the menu options.

\_\_\_ b. In the “Create a Mediation Module” window, enter the following information.

- In the **Module name** field, type: RouterMediationService
- In the **Target runtime environment** field, select **IBM Process Server v8.5.7**.
- Change the **Name** of the mediation component to: RouteRequest

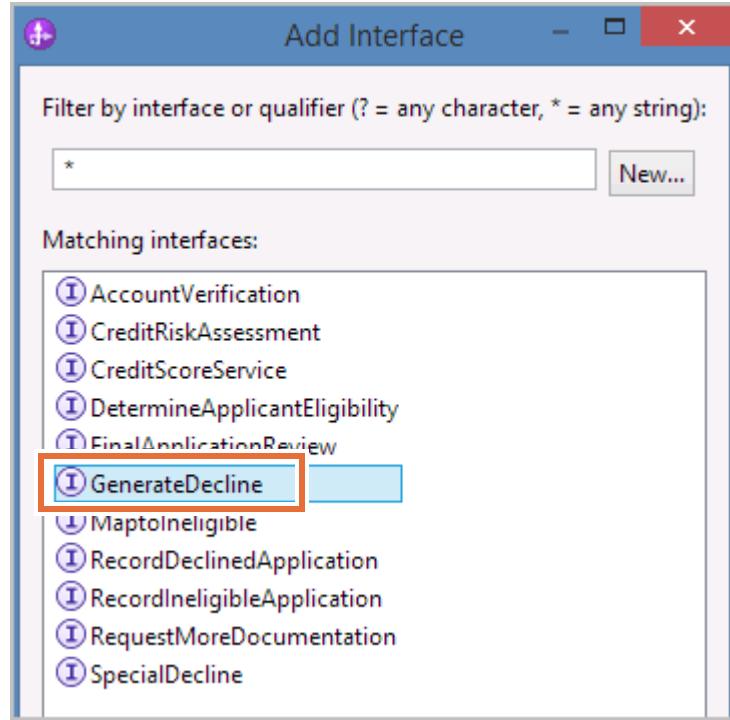


\_\_\_ c. Accept the remaining default options and click **Next**.

\_\_\_ d. In the “Select required libraries” window, select the **FoundationLibrary** check box to include it as a dependency.

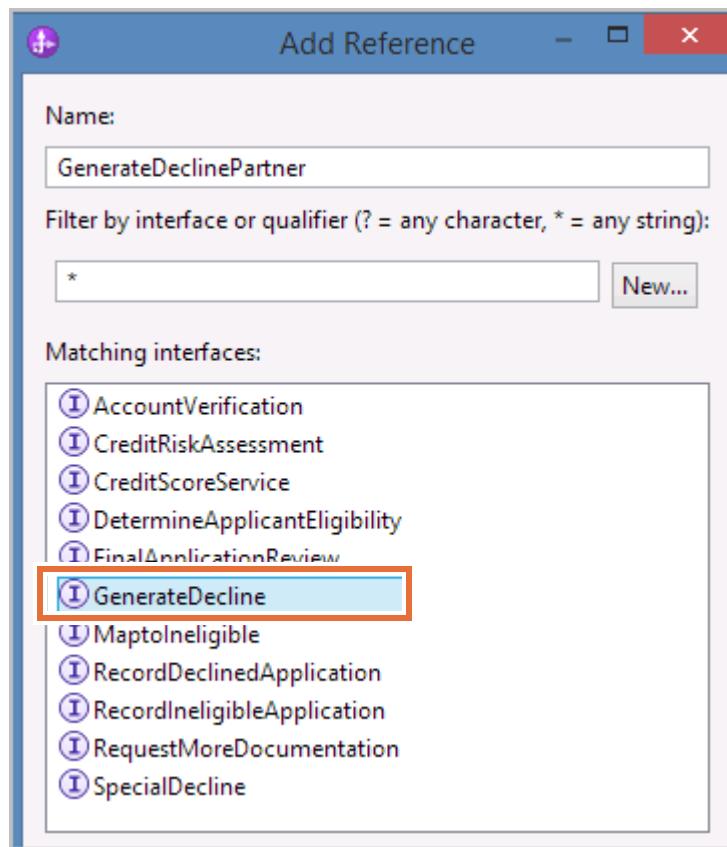


- \_\_ e. Click **Finish**.
- \_\_ 2. Add the GenerateDecline interface to the RouteRequest component.
  - \_\_ a. On the **RouterMediationService** assembly diagram, right-click the **RouteRequest** component and click **Add > Interface** from the menu.
  - \_\_ b. In the Add Interface dialog box, select the **GenerateDecline** interface.



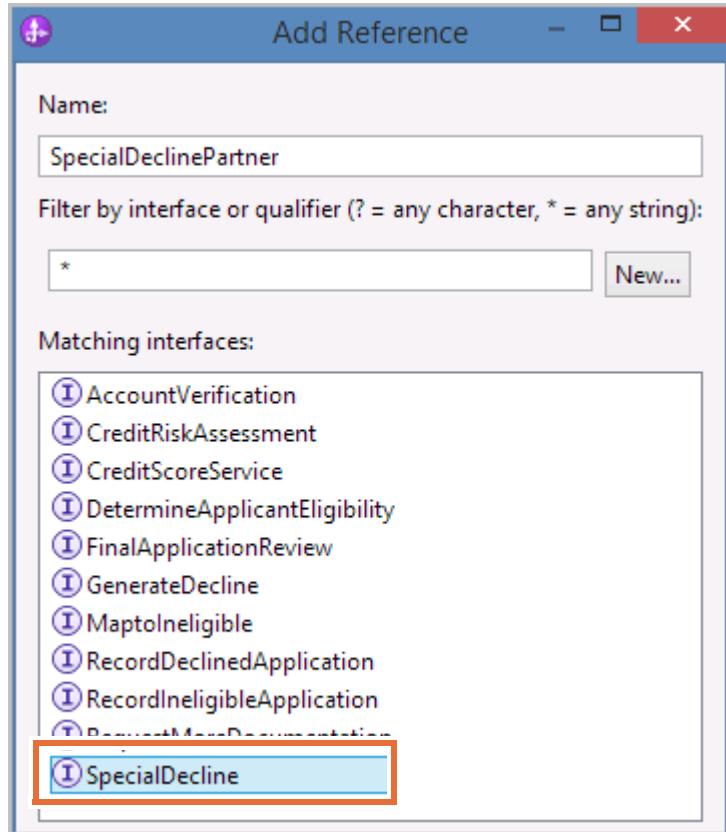
- \_\_ c. Click **OK**.
- \_\_ d. Save your changes.
- \_\_ 3. Add two references to the RouteRequest flow component. Add the GenerateDeclinePartner reference that uses the GenerateDecline interface and add the SpecialDeclinePartner that uses the SpecialDecline interface.
  - \_\_ a. On the **RouterMediationService** assembly diagram, right-click the **RouteRequest** component and click **Add > Reference**.

- \_\_\_ b. In the Add Reference dialog box, select the **GenerateDecline** interface.

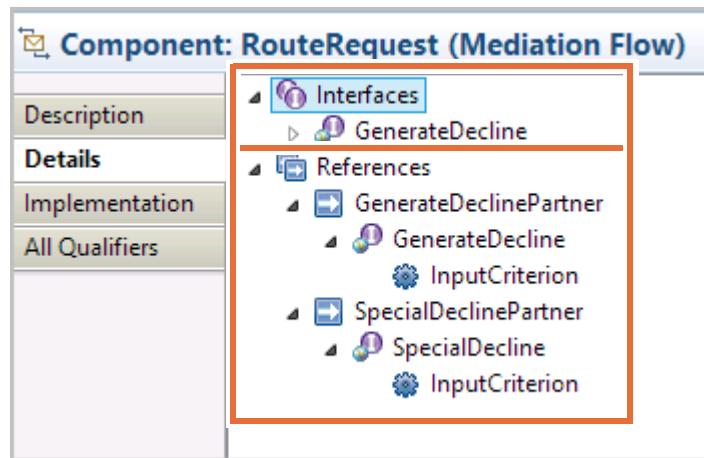


- \_\_\_ c. Click **OK**.
- \_\_\_ d. On the **RouterMediationService** assembly diagram, right-click the **RouteRequest** component and click **Add > Reference**.

- \_\_ e. In the Add Reference dialog box, select the **SpecialDecline** interface.



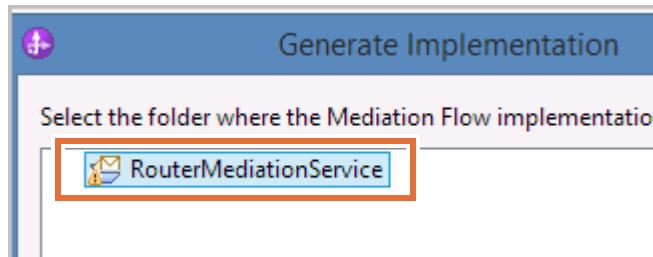
- \_\_ f. Click **OK**.
- \_\_ g. Switch to the **Details** tab in the **Properties** view.
- \_\_ h. Verify the interface and references for the **RouteRequest** flow component.



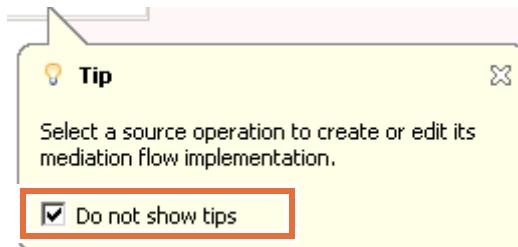
- \_\_ i. Save your changes.

## Part 2: Define an XML data map

- 1. Use the **Blank Mediation Flow** template to generate the implementation for the **RouteRequest** request flow component in the **RouterMediationService** folder.
  - a. Right-click the **RouteRequest** flow component and click **Generate Implementation**.
  - b. In the Generate Implementation dialog box, leave the **RouterMediationService** folder selected.

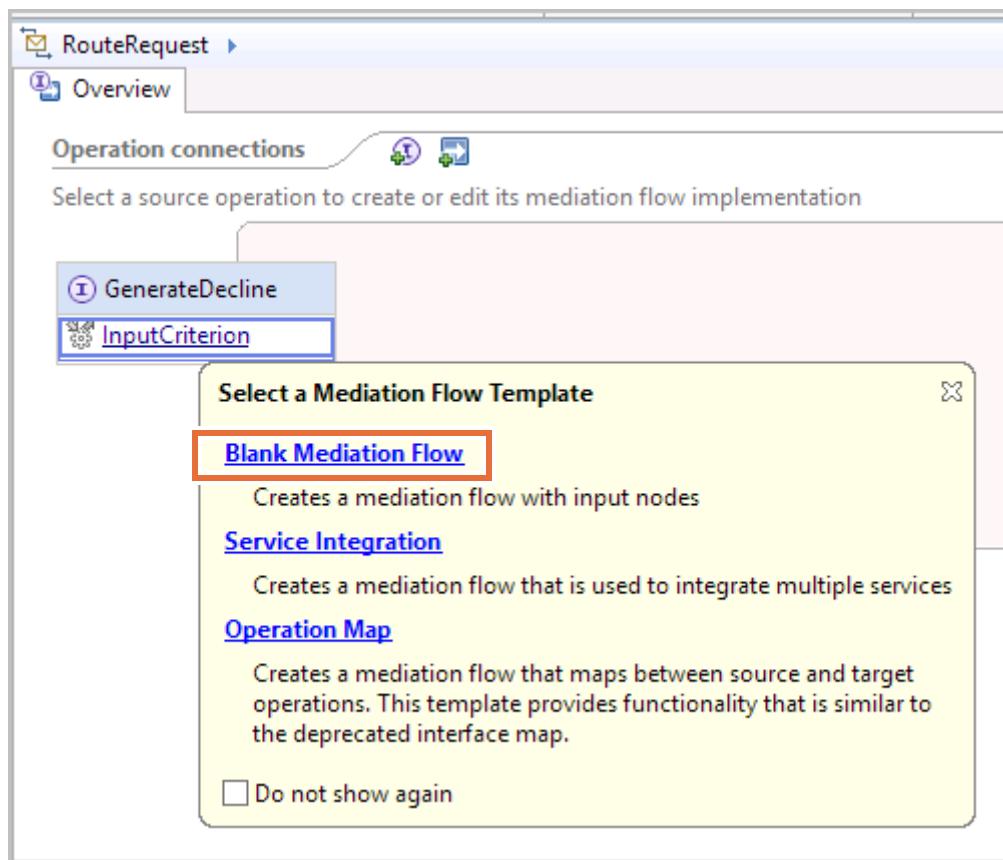


- c. Click **OK**.
- d. In the mediation flow editor, select the **Do not show tips** check box in the Tip dialog box.



- e. Close the Tip dialog box.

- \_\_\_ f. In the Operation connections section of the editor, click the **InputCriterion** operation in the **GenerateDecline** interface.



- \_\_\_ g. In the “Select a Mediation Flow Template” dialog box, click the **Blank Mediation Flow** link.

The generated request flow consists of an **Input** node and an **Input Response** node. The **Input Response** node might be off screen to the right.



- \_\_\_ 2. Add a **Message Filter** primitive in the request flow named `ChooseService`.

- \_\_\_ a. In the palette, expand the **Routing** folder and click **Message Filter**.

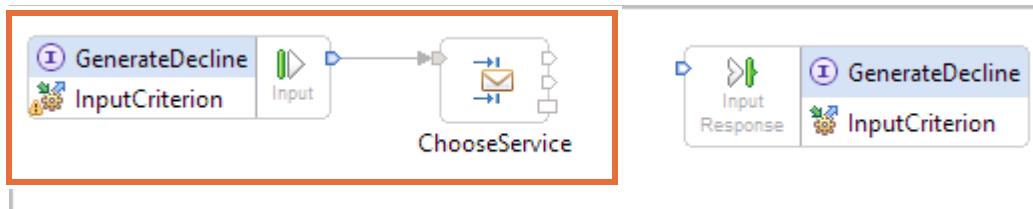
- \_\_\_ b. Click the blank space on the canvas between the **Input** and **Input Response** nodes to add the **Message Filter** primitive to the flow.



- \_\_\_ c. With the **Message Filter** primitive selected, switch to the **Description** tab in the **Properties** view.  
 \_\_\_ d. Change the **Display Name** to: ChooseService



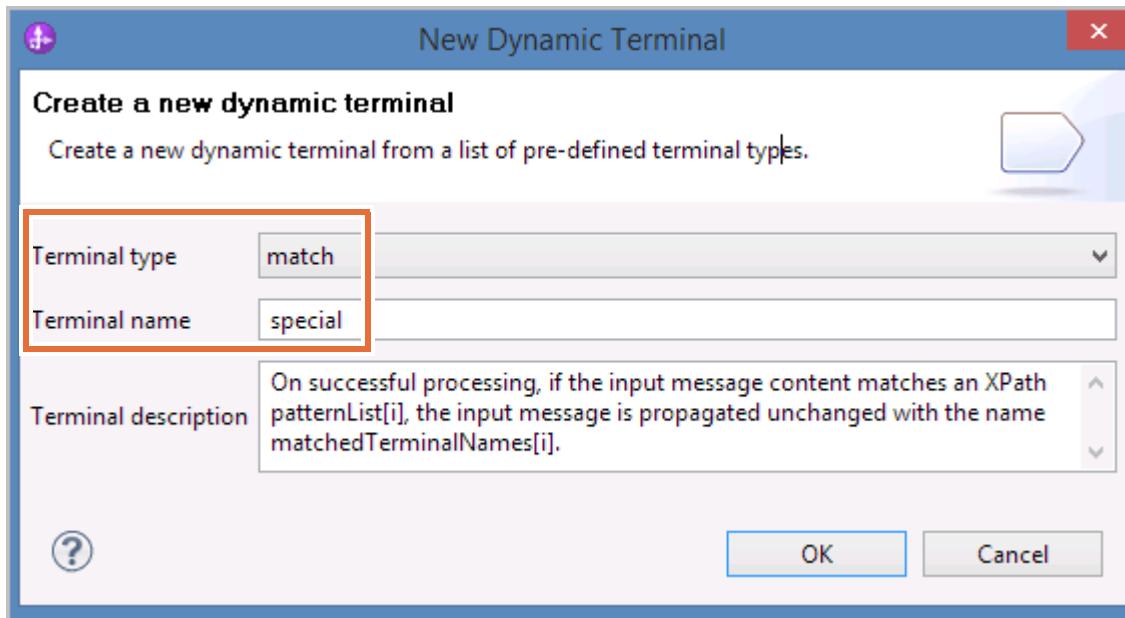
- \_\_\_ e. Save your changes.  
 \_\_\_ 3. Wire the **out** terminal of the input node to the **in** terminal of the **ChooseService** primitive.  
 \_\_\_ a. Right-click the **out** terminal of the **InputCriterion:GenerateDecline** input node and select **Add Connection** from the menu.  
 \_\_\_ b. Click the **in** terminal of the **ChooseService** primitive to add the connection.



- \_\_\_ 4. Delete the **match1** terminal on the **ChooseService** primitive and create an out terminal that is named: **special**  
 \_\_\_ a. Hover over each outbound terminal on **ChooseService** until you locate the **match1** terminal. The terminal name is listed in the descriptor.

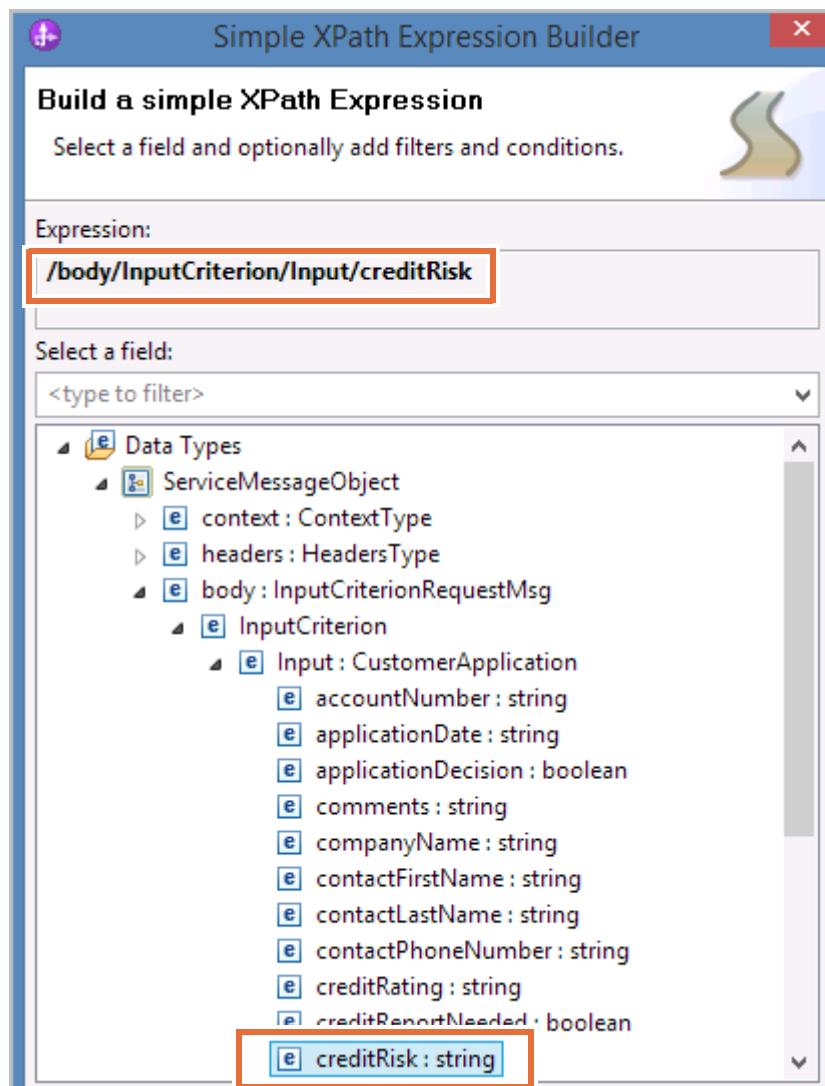


- \_\_\_ b. Right-click the **match1** terminal and click **Delete** from the menu.
- \_\_\_ c. Right-click the **ChooseService** primitive and click **Add Output Terminal** from the menu.
- \_\_\_ d. In the New Dynamic Terminal dialog box, take the following actions.
  - Leave the **Terminal type** set to: **match**
  - Change the **Terminal name** to: **special**

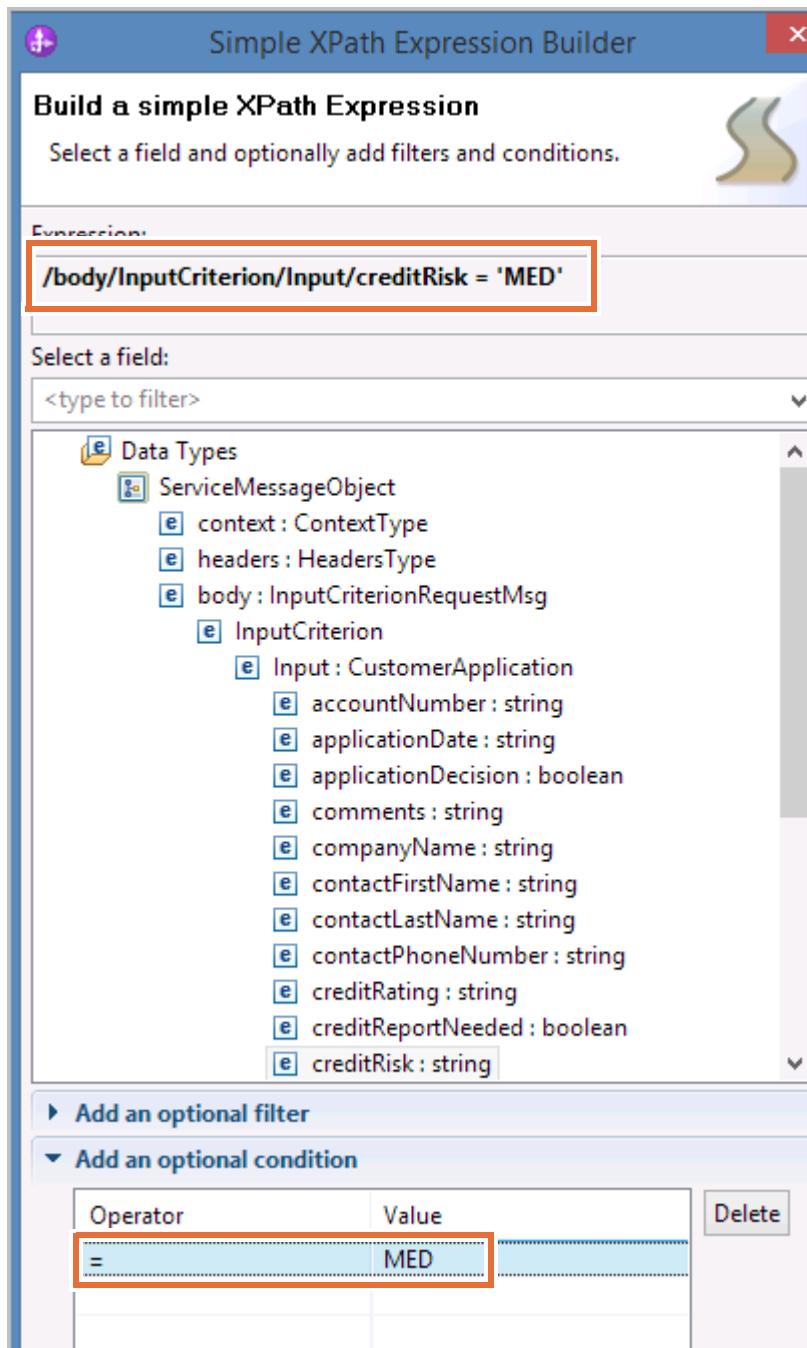


- \_\_\_ e. Click **OK**.
- \_\_\_ 5. Define a pattern on the **ChooseService** primitive that passes control to the **special** out terminal when the incoming **creditRisk** is **MED**.  
The XPath expression that the pattern uses is:  
`/body/InputCriterion/Input/creditRisk='MED'`
- \_\_\_ a. Select the **ChooseService** primitive and switch to the **Details** tab in the **Properties** view.
- \_\_\_ b. Click **Add** to add a pattern.
- \_\_\_ c. In the "Add/Edit properties" dialog box, by the **Pattern** field, click **Edit** to open the **XPath Expression Builder**.
- \_\_\_ d. In the "Build a simple XPath Expression" window, in the "Select a field" window, expand **Data Types > ServiceMessageObject > body: InputCriterionRequestMessage > InputCriterion > Input: CustomerApplication**.

- e. Select **creditRisk: string** to see `/body/InputCriterion/Input/creditRisk` in the Expression field.

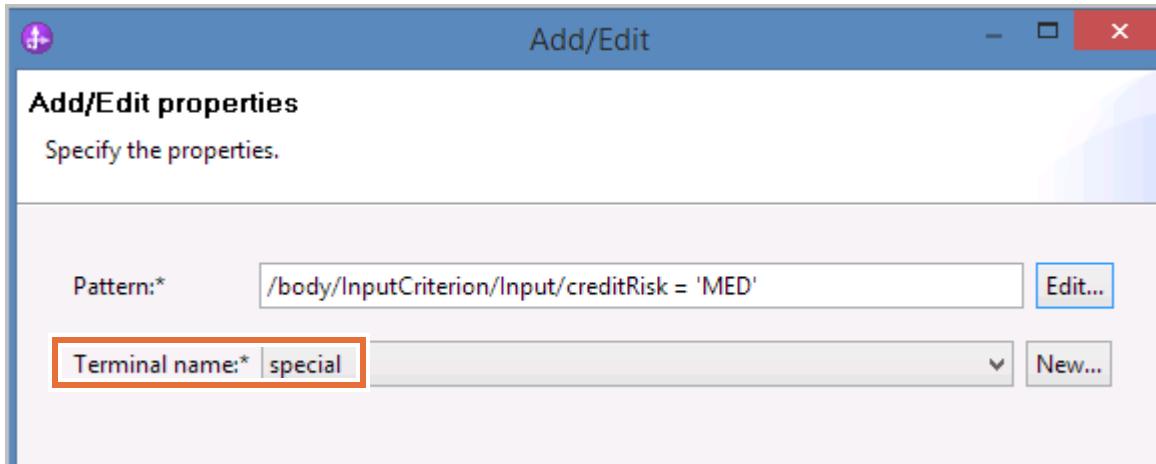


- \_\_\_ f. In the **Add an optional condition** section, leave **Operator** set to **=**, place the cursor in the **Value** column, type **MED**, and press Enter. The **Expression** field contains:  
`/body/InputCriterion/Input/creditRisk='MED'`



- \_\_\_ g. Click **OK**.

- \_\_\_ h. The **Pattern** field is populated with the XPath expression that you built for the **special** terminal. Verify that `special` is listed in the **Terminal name** field.



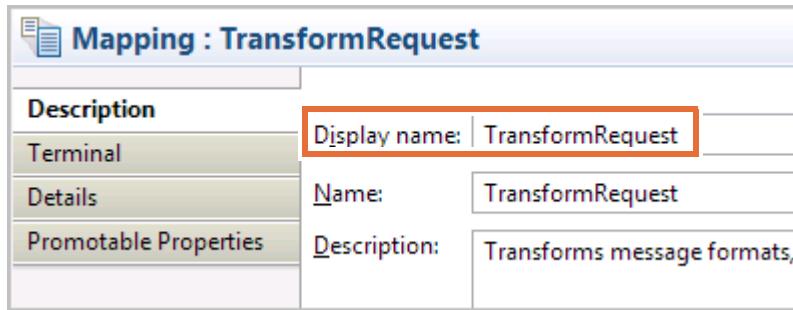
- \_\_\_ i. Click **Finish**. The expression is added to the **Pattern** list in the **Properties** view.

| Pattern                                       | Terminal name |
|-----------------------------------------------|---------------|
| /body/InputCriterion/Input/creditRisk = 'MED' | special       |

- \_\_\_ j. Save your changes.

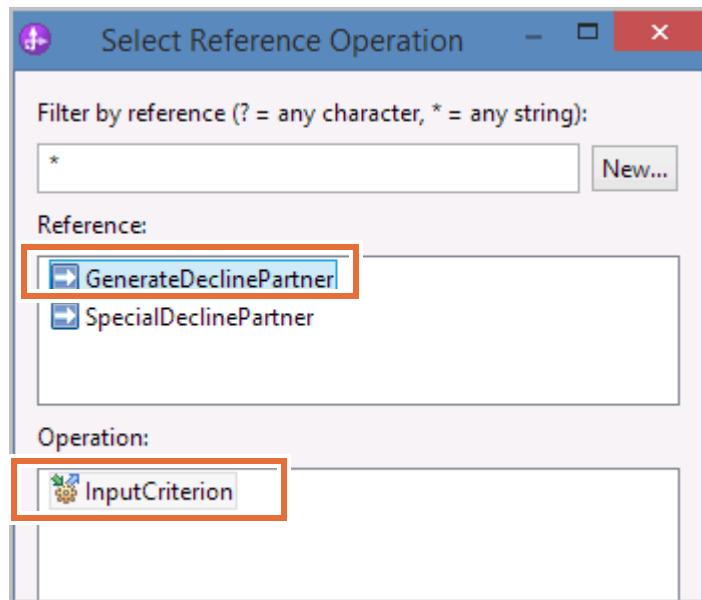
- \_\_\_ 6. Add a Mapping primitive that is named `TransformRequest` to the right of the **ChooseService** primitive. Because of the disparity between the `GenerateDecline` and `SpecialDecline` interfaces, you must create a data map before the `SpecialDeclinePartner` callout node can be invoked.
- In the palette, expand **Transformation** and click **Mapping**.
  - Click an empty area of the canvas to the right of the **ChooseService** primitive.
  - With the Mapping primitive selected, switch to the **Description** tab in the **Properties** view.

- \_\_ d. Change the **Display name** to: TransformRequest



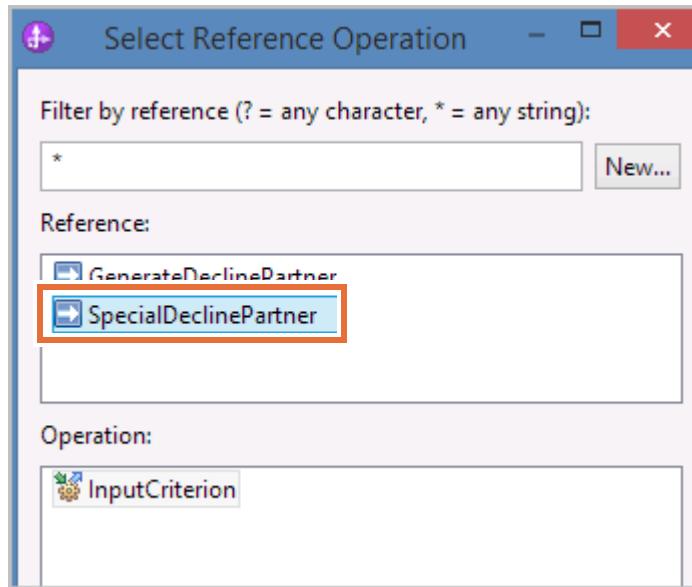
- \_\_ e. Save your changes.

- 7. Add two callout nodes to the request flow. One callout node is for the **InputCriterion** operation on the **SpecialDecline** interface of **SpecialDeclinePartner**. The second is for the **InputCriterion** operation on the **GenerateDecline** interface of **GenerateDeclinePartner**.
- \_\_ a. In the palette, expand **Service Invocation** and click **Callout**.
  - \_\_ b. Click a blank area on the canvas to the right of the **ChooseService** primitive.
  - \_\_ c. In the Select Reference Operation dialog box, select the **GenerateDeclinePartner** reference.
  - \_\_ d. Because the interface has only one operation, **InputCriterion** is automatically selected.

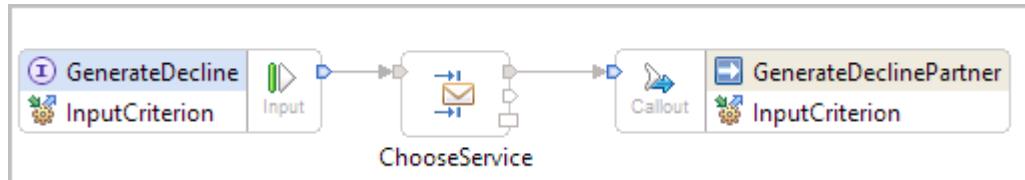


- \_\_ e. Click **OK**.
- \_\_ f. In the palette, expand **Service Invocation** and click **Callout**.
- \_\_ g. Click a blank area on the canvas to the right of the **ChooseService** primitive.
- \_\_ h. In the Select Reference Operation dialog box, select the **SpecialDeclinePartner** reference.

- \_\_\_ i. Because the interface has only one operation, **InputCriterion** is automatically selected.

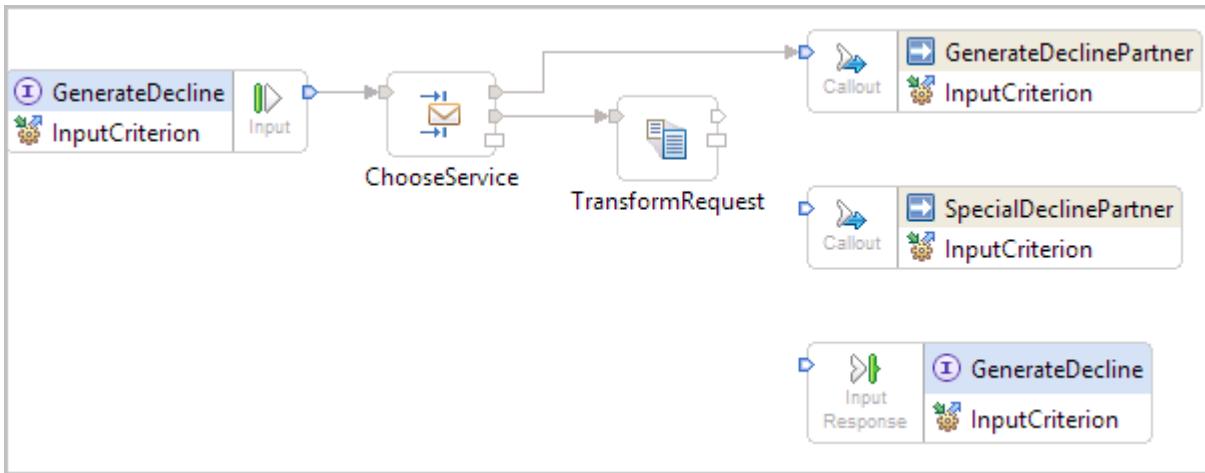


- \_\_\_ j. Click **OK**.
- \_\_\_ k. For readability, right-click the canvas and click **Layout Contents** from the menu.
- \_\_\_ 8. Wire the **default** terminal of the **ChooseService** primitive to the **in** terminal of the **GenerateDeclinePartner** callout node.
- Right-click the **default** terminal on the **ChooseService** primitive and click **Add Connection** from the menu.
  - Click the **in** terminal of the **GenerateDeclinePartner** callout node to add the connection.



- \_\_\_ c. Save your changes.
- \_\_\_ 9. Wire the **special** terminal of the **ChooseService** primitive to the **in** terminal of the **TransformRequest** primitive.
- Right-click the **special** terminal on the **ChooseService** primitive and click **Add Connection** from the menu.

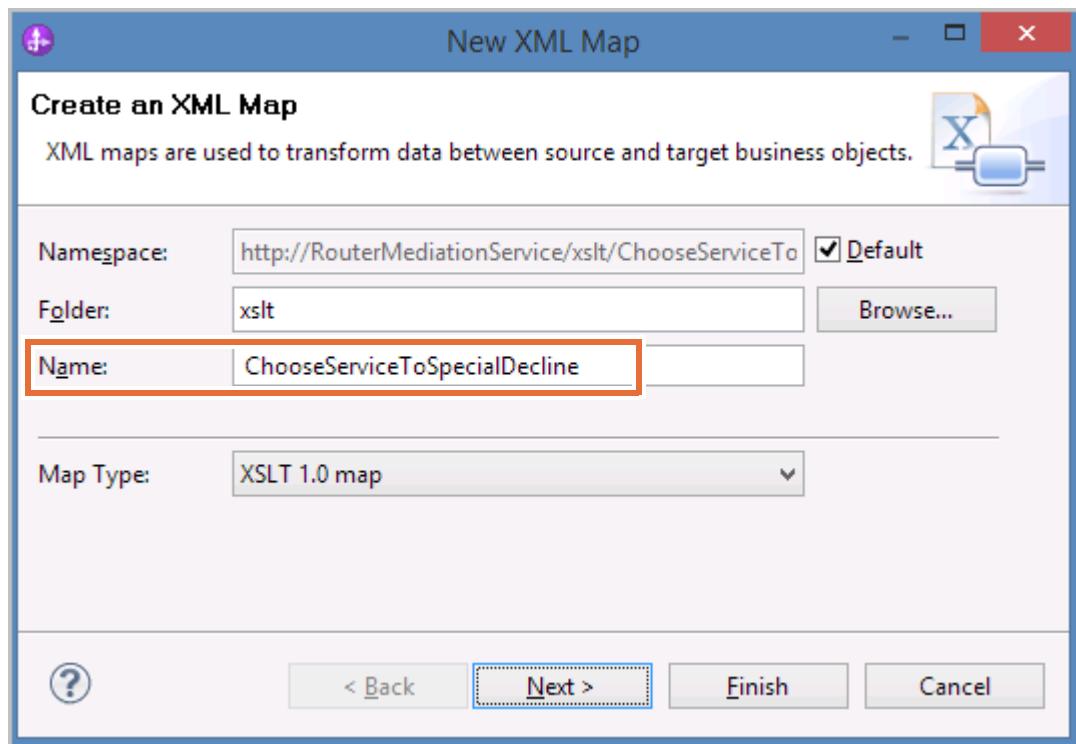
- \_\_ b. Click the **in** terminal of the **TransformRequest** primitive to add the wire.



- \_\_ c. Save your changes.
- 10. Wire the **out** terminal of the **TransformRequest** primitive to the **in** terminal of the **SpecialDeclinePartner** callout node.
- \_\_ a. Right-click the **out** terminal on the **TransformRequest** primitive and click **Add Connection** from the menu.
  - \_\_ b. Click the **in** terminal of the **SpecialDeclinePartner** callout node to add the connection.
  - \_\_ c. Save your changes.
- 11. Implement the **TransformRequest** XML data map to transform the message from the **ChooseService** primitive to the **SpecialDeclinePartner** callout. The map uses a local map to move the contents of the input SMO to the contents of the output SMO.
- \_\_ a. Select the **TransformRequest** primitive and switch to the **Details** tab in the **Properties** view.
  - \_\_ b. By the **Mapping File** field, click **New** to create a map.

\_\_\_ c. In the “Create an XML Map” window, take the following actions.

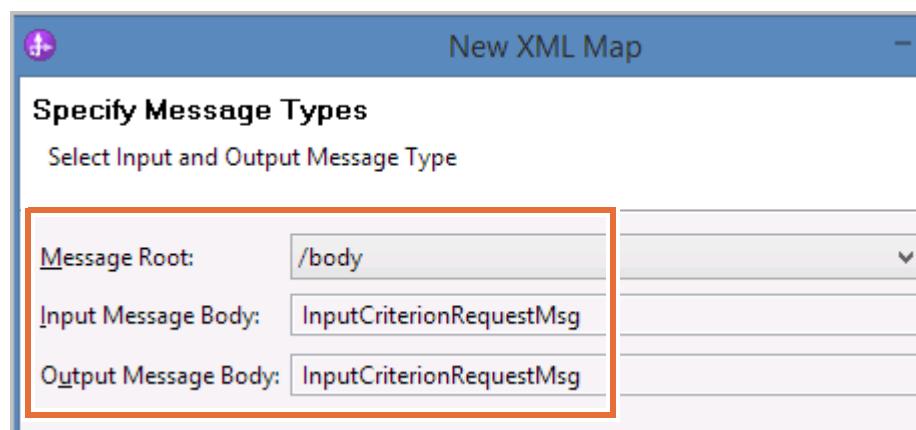
- Verify that `xslt` is listed in the **Folder** field.
- Change the **Name** to: `ChooseServiceToSpecialDecline`



\_\_\_ d. Accept the remaining default options and click **Next**.

\_\_\_ e. In the Specify Message Types window, verify the following information.

- **Message Root** is set to: `/body`
- **Input Message Body** is set to: `InputCriterionRequestMsg`
- **Output Message Body** is set to: `InputCriterionRequestMsg`

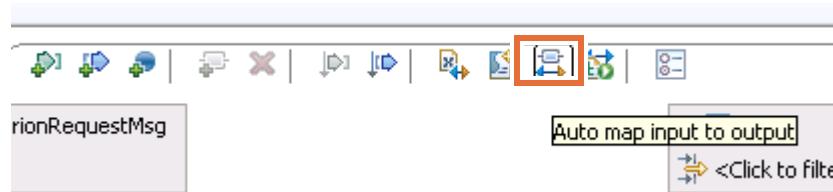


\_\_\_ f. Ignore any warnings in the dialog box and click **Finish** to open the map editor.

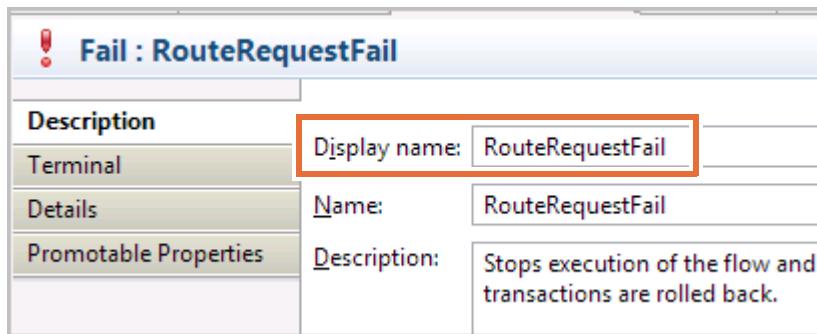
- \_\_\_ g. In the map editor, expand **InputCriterion** in the input SMO and **InputCriterion** in the output SMO.



- \_\_\_ h. Click the **Auto map input to output** icon.

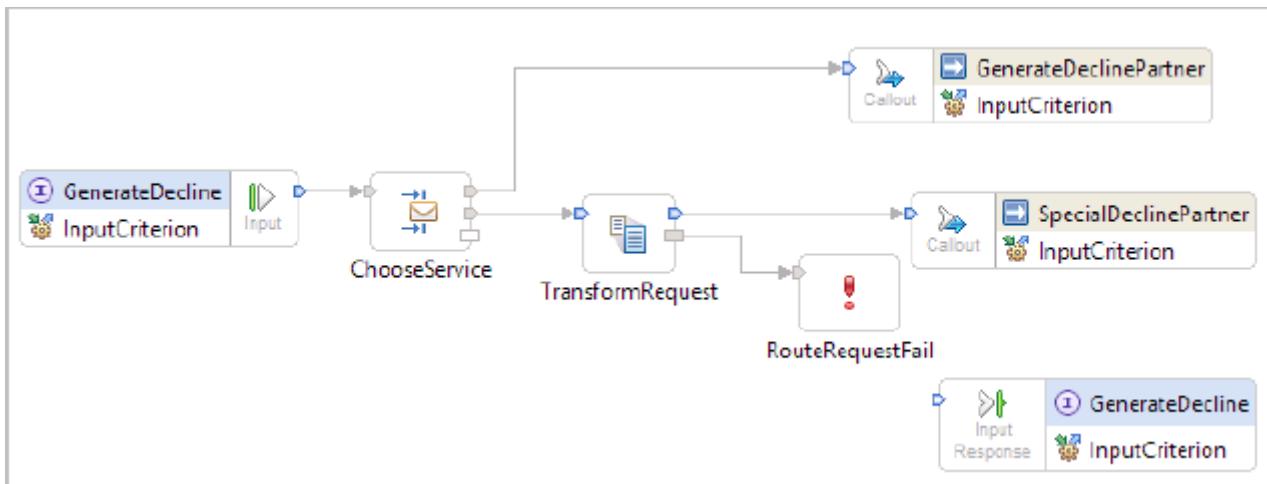


- \_\_\_ i. Accept the defaults in the Auto Map window and click **Finish**.
- \_\_\_ j. Click **File > Save All** from the menu options. The **Problems** view contains no errors. You can ignore any warnings.
- \_\_\_ k. Close the mapping editor.
- \_\_\_ 12. Add a **Fail** primitive that is named `RouteRequestFail` to the **RouteRequest** request mediation flow and wire it to the **fail** terminal on the **TransformRequest** primitive. If a failure is encountered in the map, the fail primitive stops flow execution and throws an exception.
- \_\_\_ a. In the palette, expand **Error Handling** and select **Fail**.
- \_\_\_ b. Click an empty portion of the flow diagram to the right of the **TransformRequest** primitive.
- \_\_\_ c. With the fail primitive selected, switch to the **Description** tab in the **Properties** view.
- \_\_\_ d. Change the **Display Name** to: `RouteRequestFail`

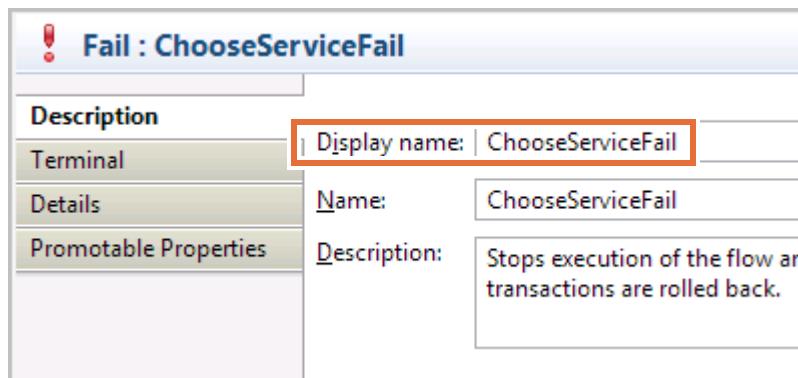


- \_\_\_ e. Right-click the **fail** terminal on the **TransformRequest** primitive and click **Add Connection** from the menu.
- \_\_\_ f. Click the **in** terminal on the **RouteRequestFail** primitive to add the connection.

- \_\_ g. To aid readability, right-click the flow diagram and click **Layout Contents** from the menu. The flow diagram resembles the following figure:

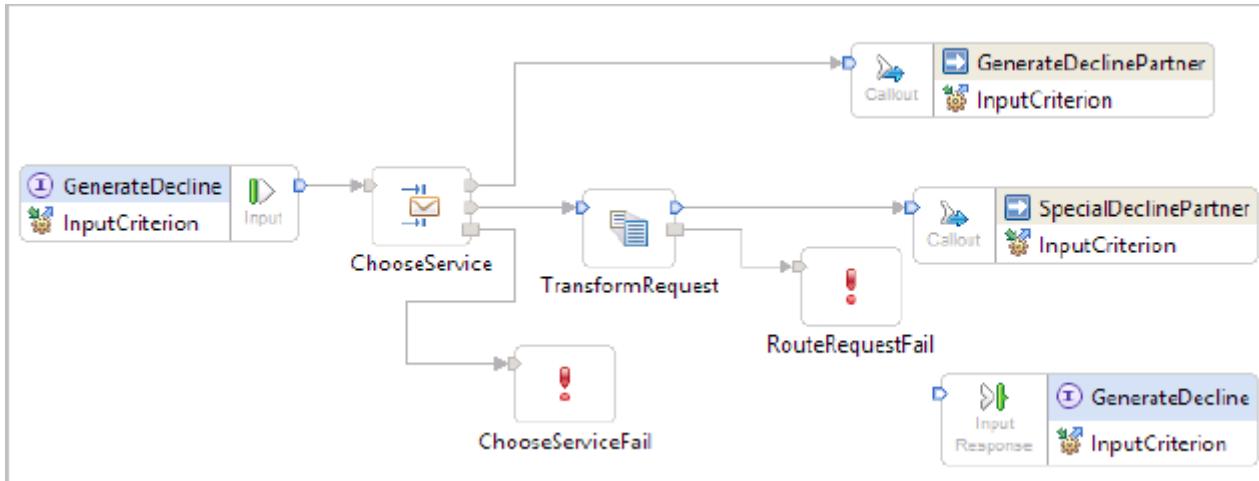


- \_\_ 13. Add a **Fail** primitive named `ChooseServiceFail` to the **RouteRequest** request mediation flow and wire it to the **fail** terminal on the **ChooseService** primitive. If a failure is encountered in the message filter, the fail primitive stops flow execution and throws an exception.
- In the palette, expand **Error Handling** and select **Fail**.
  - Click an empty portion of the flow diagram under the **ChooseService** primitive.
  - With the fail primitive selected, switch to the **Description** tab in the **Properties** view.
  - Change the **Display Name** to: `ChooseServiceFail`

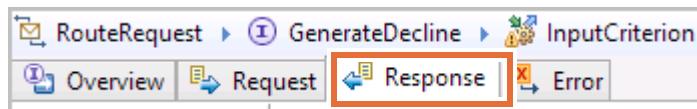


- Right-click the **fail** terminal on the **ChooseService** primitive and click **Add Connection** from the menu.
- Click the **in** terminal on the **ChooseServiceFail** primitive to add the connection.

- \_\_\_ g. To aid readability, right-click the flow diagram and click **Layout Contents** from the menu. The request flow diagram resembles the following figure:

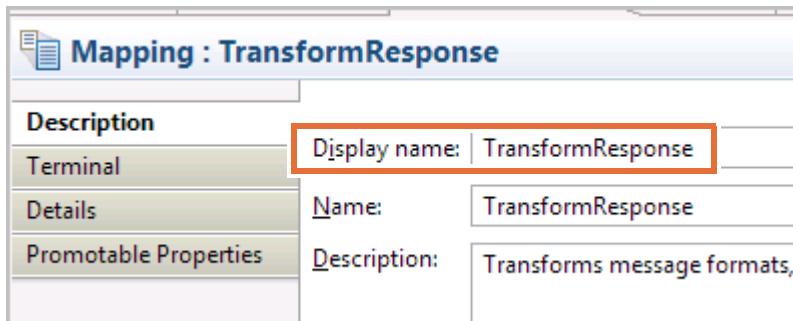


- \_\_\_ 14. In the mediation flow editor, click the **Response** tab. You now implement the **response** message flow for the mediation.



- \_\_\_ 15. Add a Mapping primitive named **TransformResponse** to the left of the **Input Response** node.

- In the palette, expand **Transformation** and click **Mapping**.
- Click the empty space on the canvas to the left of the **Input Response** node.
- Switch to the **Description** tab in the **Properties** view.
- Change the **Display Name** to: **TransformResponse**

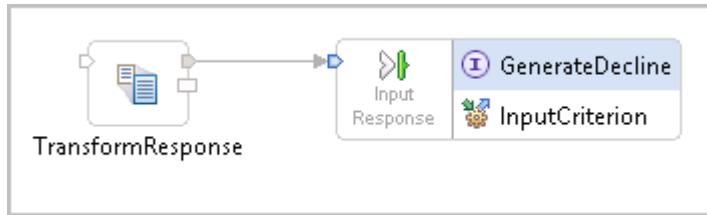


- Save your changes.

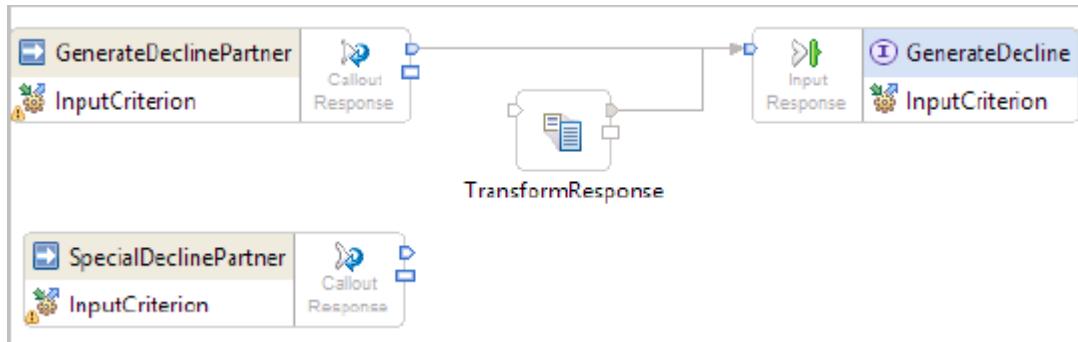
- \_\_\_ 16. Wire the **out** terminal of the **TransformResponse** primitive to the **in** terminal of the **GenerateDecline** Input Response node.

- Right-click the **out** terminal of the **TransformResponse** primitive and click **Add Connection** from the menu.

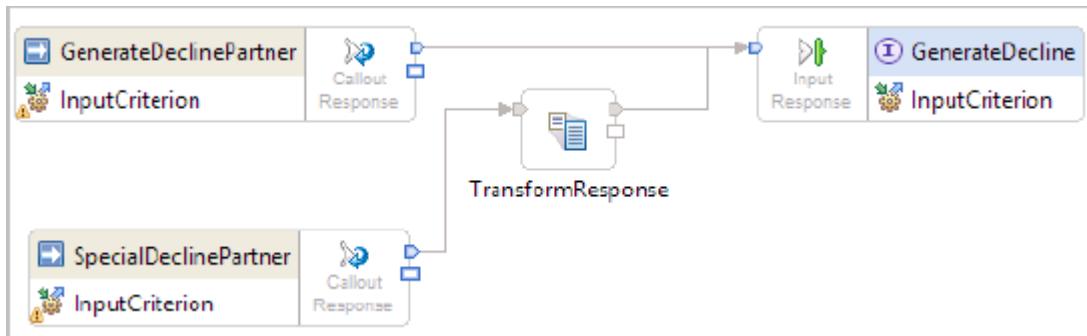
- \_\_\_ b. Click the **in** terminal of the **GenerateDecline** Input Response node to add the connection.



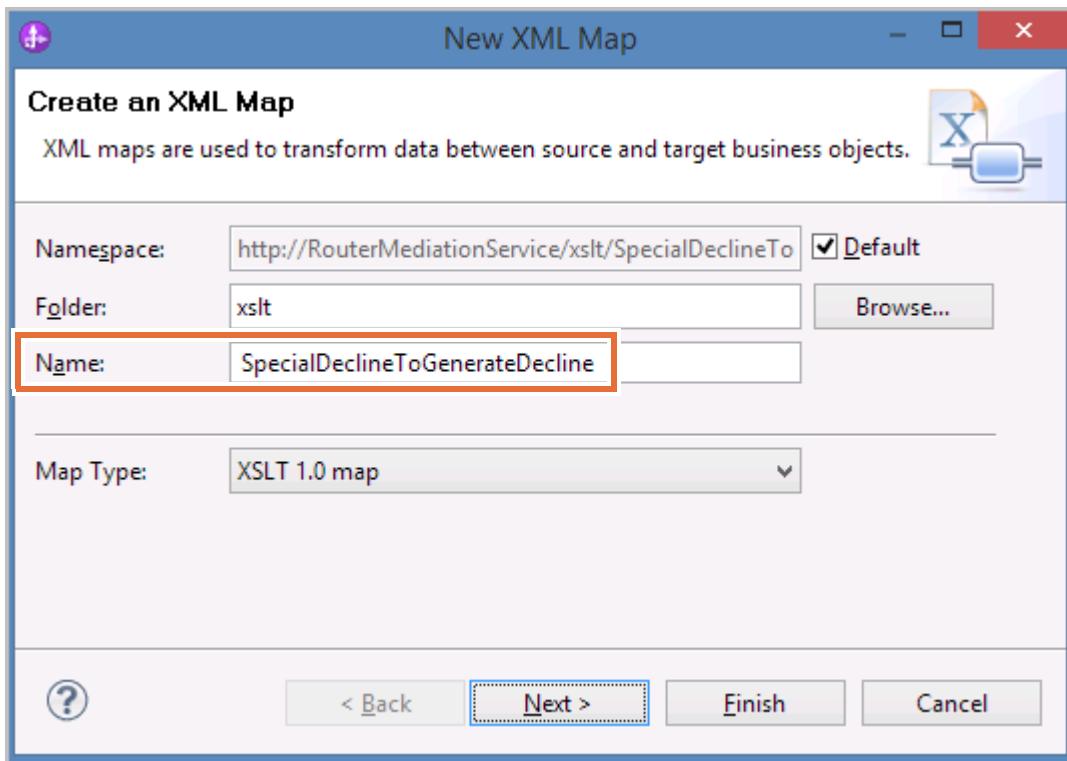
- \_\_\_ 17. Wire the **out** terminal of the **GenerateDeclinePartner** Callout Response node to the **in** terminal of the **GenerateDecline** Input Response node.
- \_\_\_ a. Right-click the **out** terminal of the **GenerateDeclinePartner** Callout Response node and click **Add Connection** from the menu.
  - \_\_\_ b. Click the **in** terminal of the **GenerateDecline** Input Response node to add the connection. Your diagram resembles the following figure:



- \_\_\_ 18. Wire the **out** terminal of the **SpecialDeclinePartner** Callout Response node to the **in** terminal of the **TransformResponse** primitive.
- \_\_\_ a. Right-click the **out** terminal of the **SpecialDeclinePartner** Callout Response node and click **Add Connection** from the menu.
  - \_\_\_ b. Click the **in** terminal of the **TransformResponse** primitive to add the connection.
  - \_\_\_ c. Save your changes. Continue ignoring any errors in the **Problems** view.
  - \_\_\_ d. For readability, right-click the flow diagram and click **Layout Contents** from the menu. Your diagram resembles the following figure:



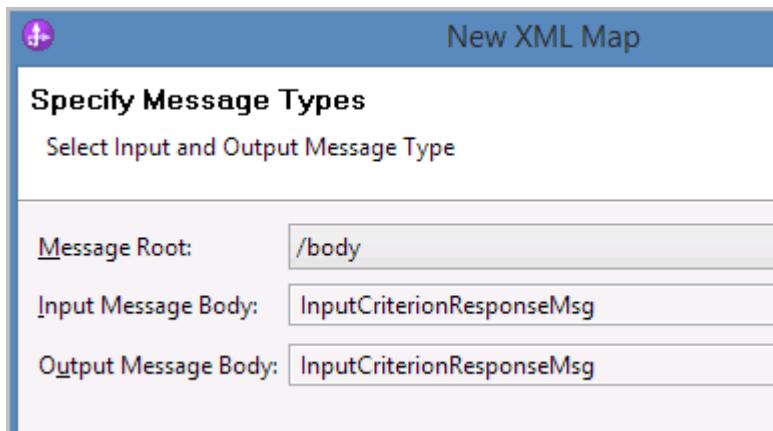
- 19. Implement the **TransformResponse** XML map to transform the input message from the **SpecialDeclinePartner** Callout Response to the **GenerateDecline** Input Response. The map uses a **Move** transformation to move the contents of the input SMO to the contents of the output SMO.
- a. Select the **TransformResponse** primitive and switch to the **Details** tab in the **Properties** view.
- b. By the **Mapping File** field, click **New** to create a map.
- c. In the “Create an XML Map” window, do the following steps.
- Verify that the **Folder** field is set to: xsslt
  - Change the **Name** to: SpecialDeclineToGenerateDecline



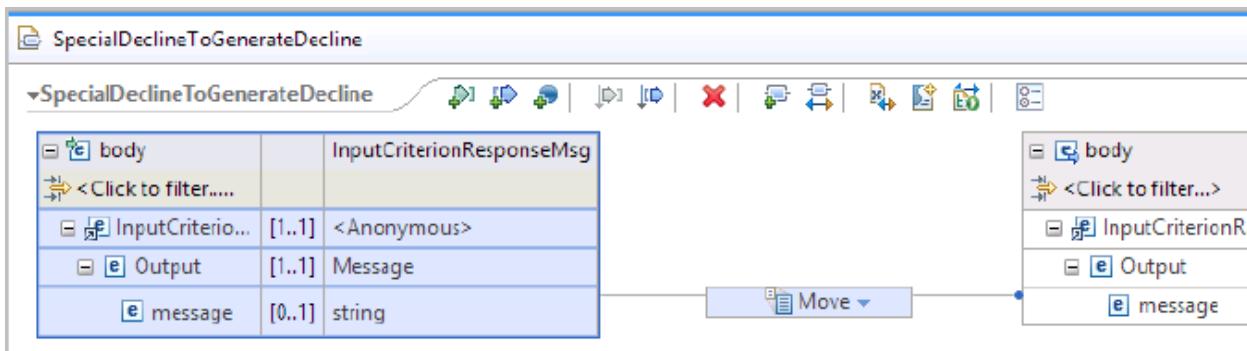
- d. Click **Next**.

- \_\_\_ e. In the Specify Message Types window, verify the following information.

- **Message Root:** /body
- **Input Message Body:** InputCriterionResponseMsg
- **Output Message Body:** InputCriterionResponseMsg

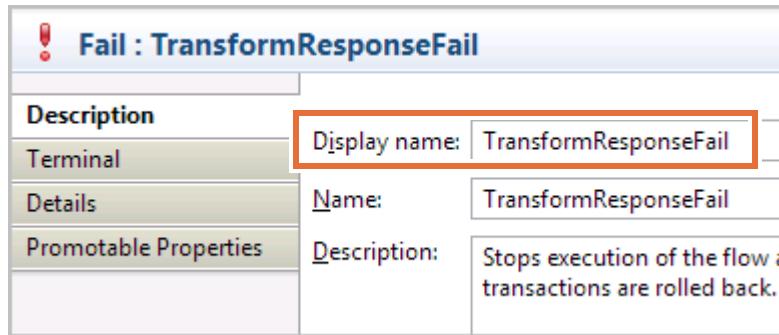


- \_\_\_ f. Ignore any warnings in the dialog box and click **Finish**.
- \_\_\_ g. Expand **InputCriterionResponse > Output** in the input SMO and expand **InputCriterionResponse > Output** in the output SMO.
- \_\_\_ h. In the input SMO, right-click the **message** attribute and click **Create Connection** from the menu.
- \_\_\_ i. Click the **message** attribute in the output SMO to create the **Move** transformation.

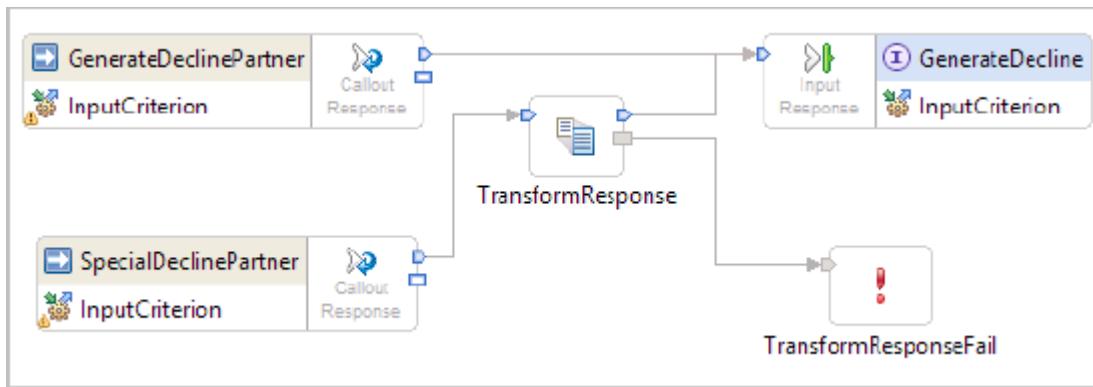


- \_\_\_ j. Click **File > Save All** from the menu options.
- \_\_\_ k. Close the mapping editor.
- \_\_\_ 20. Add a **Fail** primitive that is named `TransformResponseFail` to the **RouteRequest** response mediation flow and wire it to the **fail** terminal on the **TransformResponse** primitive. If a failure is encountered in the map, the fail primitive stops flow execution and throws an exception.
- \_\_\_ a. In the palette, expand **Error Handling** and select **Fail**.
- \_\_\_ b. Click an empty portion of the flow diagram to the right of the **TransformResponse** primitive.
- \_\_\_ c. With the fail primitive selected, switch to the **Description** tab in the **Properties** view.

- \_\_ d. Change the **Display Name** to: TransformResponseFail

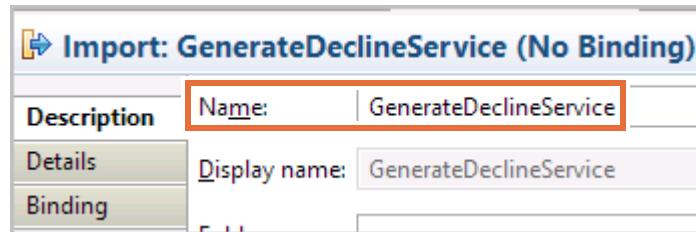


- \_\_ e. Right-click the **fail** terminal on the **TransformResponse** primitive and click **Add Connection** from the menu.
- \_\_ f. Click the **in** terminal on the **TransformResponseFail** primitive to add the connection.
- \_\_ g. To aid readability, right-click the flow diagram and click **Layout Contents** from the menu. The response flow diagram resembles the following figure:

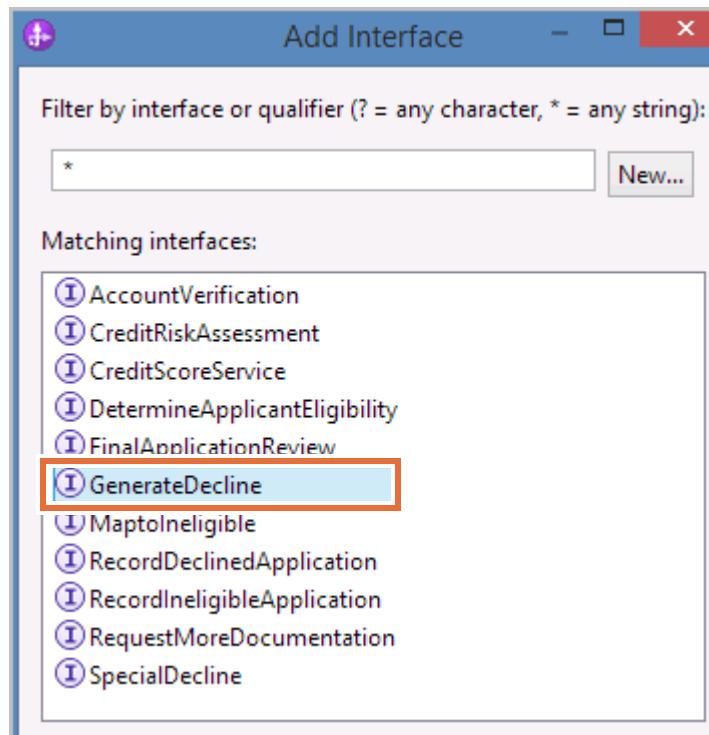


- \_\_ 21. Save your changes. Verify that the **Problems** view contains no errors. You can ignore any warnings.
- \_\_ 22. Close the mediation flow editor.
- \_\_ 23. Add an import component with an SCA binding that is named **GenerateDeclineService** to the **RouterMediationService** assembly diagram. The **GenerateDeclineService** import uses the **GenerateDecline** interface. The target of the import is the **GenerateDeclineExport** in the **FoundationServices** module.
- \_\_ a. In the **RouterMediationService** assembly diagram, expand the **Components** section of the palette and click **Import**.
- \_\_ b. Click any blank space on the canvas to add the import component.
- \_\_ c. Switch to the **Description** tab in the **Properties** view.

- \_\_\_ d. Change the **Name** to: GenerateDeclineService

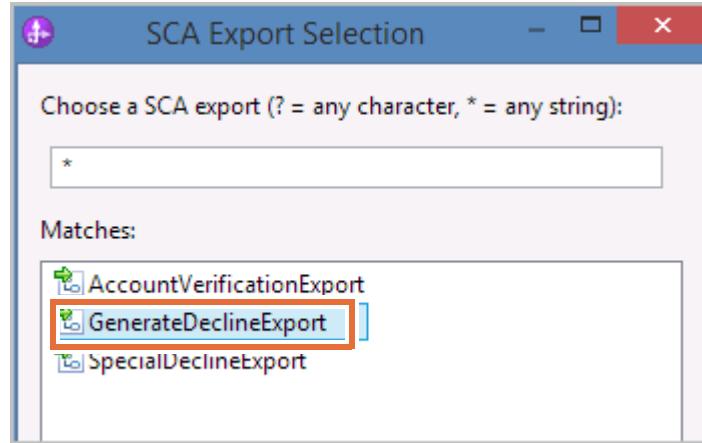


- \_\_\_ e. Right-click **GenerateDeclineService** and click **Add Interface** from the menu.  
\_\_\_ f. In the Add Interface dialog box, select **GenerateDecline**.

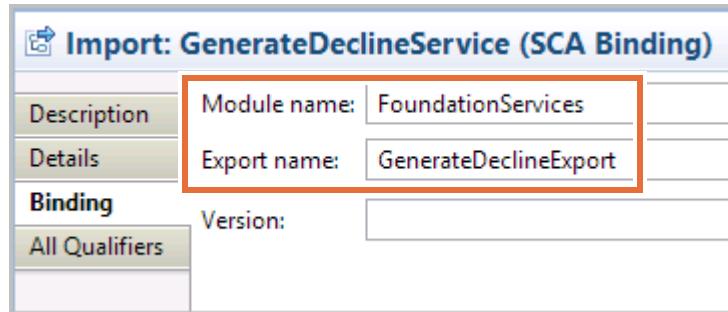


- \_\_\_ g. Click **OK**.  
\_\_\_ h. Right-click the **GenerateDeclineService** import and click **Generate Binding > SCA Binding**.  
\_\_\_ i. Switch to the **Binding** tab in the **Properties** view.

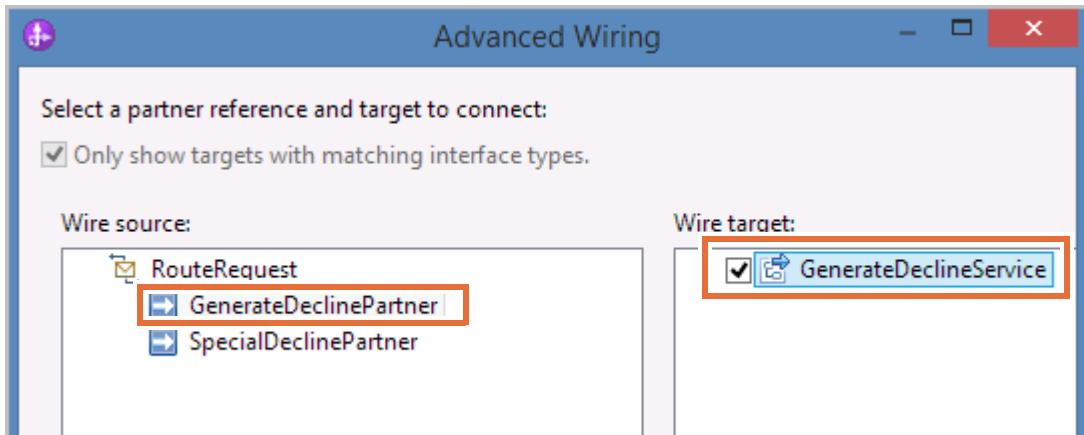
- \_\_ j. For **Export Name**, click **Browse** and select **GenerateDeclineExport** in the SCA Export Selection dialog box.



- \_\_ k. Click **OK**. Your **Binding** tab resembles the following figure:

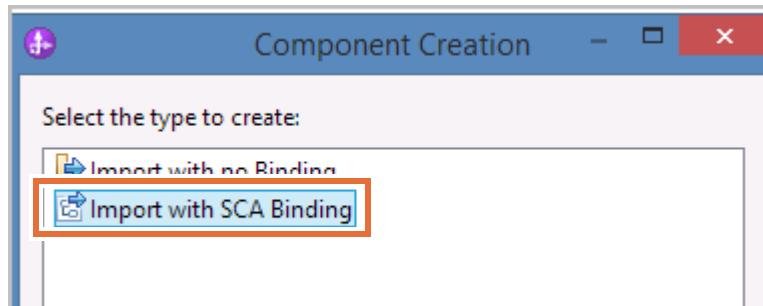


- \_\_ l. Right-click the **RouteRequest** component and click **Wire (Advanced)** from the menu.  
 \_\_ m. In the Advanced Wiring dialog box, in the “Wire source” window, select **GenerateDeclinePartner**.  
 \_\_ n. In the “Wire target” window, select **GenerateDeclineService**.

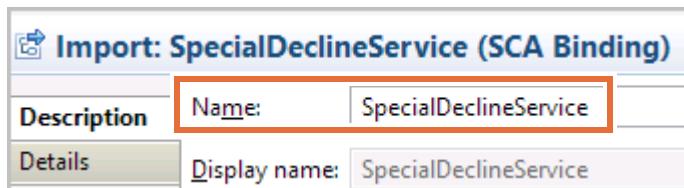


- \_\_ o. Click **OK**.  
 \_\_ p. Save your changes. You can verify the wiring by hovering over the reference. The dialog box lists: **GenerateDeclinePartner**

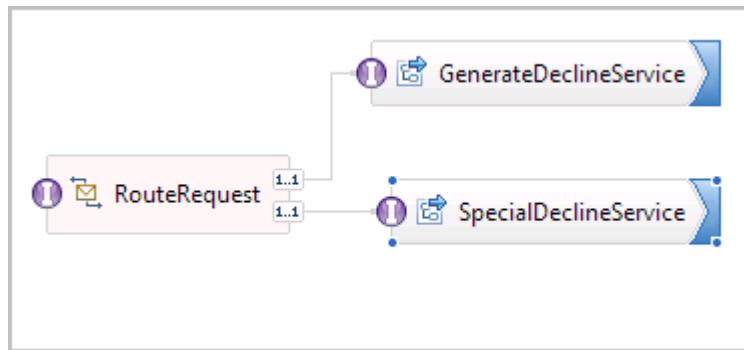
- \_\_\_ 24. Add an import component with an SCA binding that is named `SpecialDeclineService` to the **RouterMediationService** assembly diagram. The `SpecialDeclineService` import uses the `SpecialDecline` interface. The target of the import is the `SpecialDeclineExport` in the `FoundationServices` module.
- \_\_\_ a. In the Business Integration view, expand **FoundationServices > Assembly Diagram**.
  - \_\_\_ b. Drag **SpecialDeclineExport** onto the **RouterMediationService** assembly diagram.
  - \_\_\_ c. In the Component Creation dialog box, select **Import with SCA Binding**.



- \_\_\_ d. Click **OK**.
- \_\_\_ e. Switch to the **Description** tab in the **Properties** view.
- \_\_\_ f. Change the **Name** to: `SpecialDeclineService`

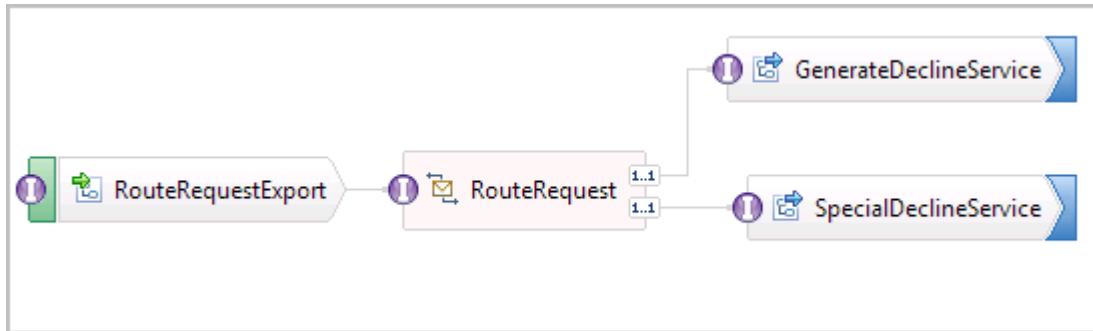


- \_\_\_ g. Right-click the **SpecialDeclineService** component and click **Wire to Existing** from the menu.
- \_\_\_ h. Save your changes. You can verify the wiring by hovering over the reference. The dialog box lists: `SpecialDeclinePartner`. Your assembly diagram resembles the following figure:



- \_\_\_ 25. Create an export that is named **RouteRequestExport**, with SCA binding, for the **RouteRequest** mediation component.
- \_\_\_ a. Right-click **RouteRequest** and click **Generate export > SCA binding**.

- \_\_\_ b. Accept the default export name: `RouteRequestExport`. Your assembly diagram resembles the following figure:



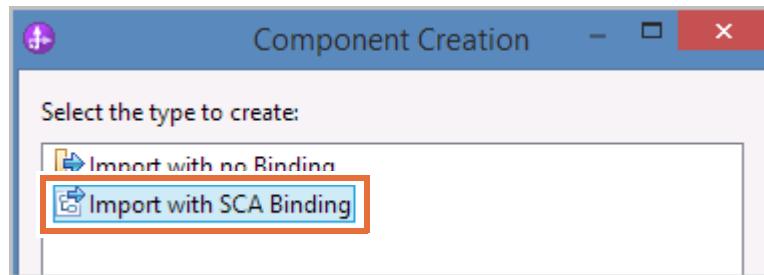
- \_\_\_ c. Save and close the assembly diagram. The **Problems** view contains no errors. You can ignore any warnings.

### **Part 3: Test a mediation module that contains a message filter mediation primitive and a Mapping primitive**

In this exercise, you implemented a mediation module that is named `RouterMediationService`. The `AccountVerification` business process invokes this service when an application is declined during `FinalApplicationReview`. In this portion of the exercise, you wire the `RouterMediationService` to the `AccountVerification` process component on the `FoundationModule` assembly diagram.

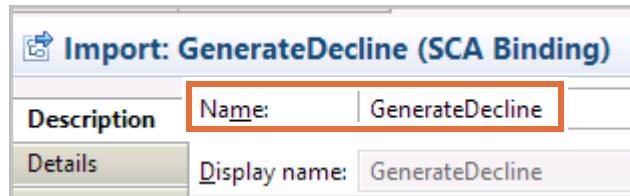
To assemble the applications:

- \_\_\_ 1. Create an import with SCA binding on the `FoundationModule` assembly diagram. The import component is named `GenerateDecline`. The target of the import is the `RouteRequestExport` component in the `RouterMediationService` module.
- \_\_\_ a. In the Business Integration view, expand **FoundationModule** and double-click **Assembly Diagram**.
- \_\_\_ b. In the Business Integration view, expand **RouterMediationService > Assembly Diagram**.
- \_\_\_ c. Drag `RouteRequestExport` from `RouterMediationService` onto the `FoundationModule` assembly diagram.
- \_\_\_ d. In the Component Creation dialog box, select **Import with SCA Binding**.



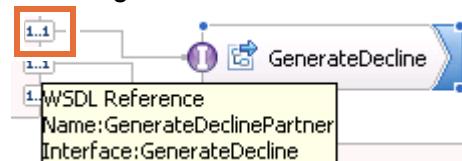
- \_\_\_ e. Click **OK**.
- \_\_\_ f. Switch to the **Description** tab in the **Properties** view.

- \_\_\_ g. Change the **Name** to: GenerateDecline

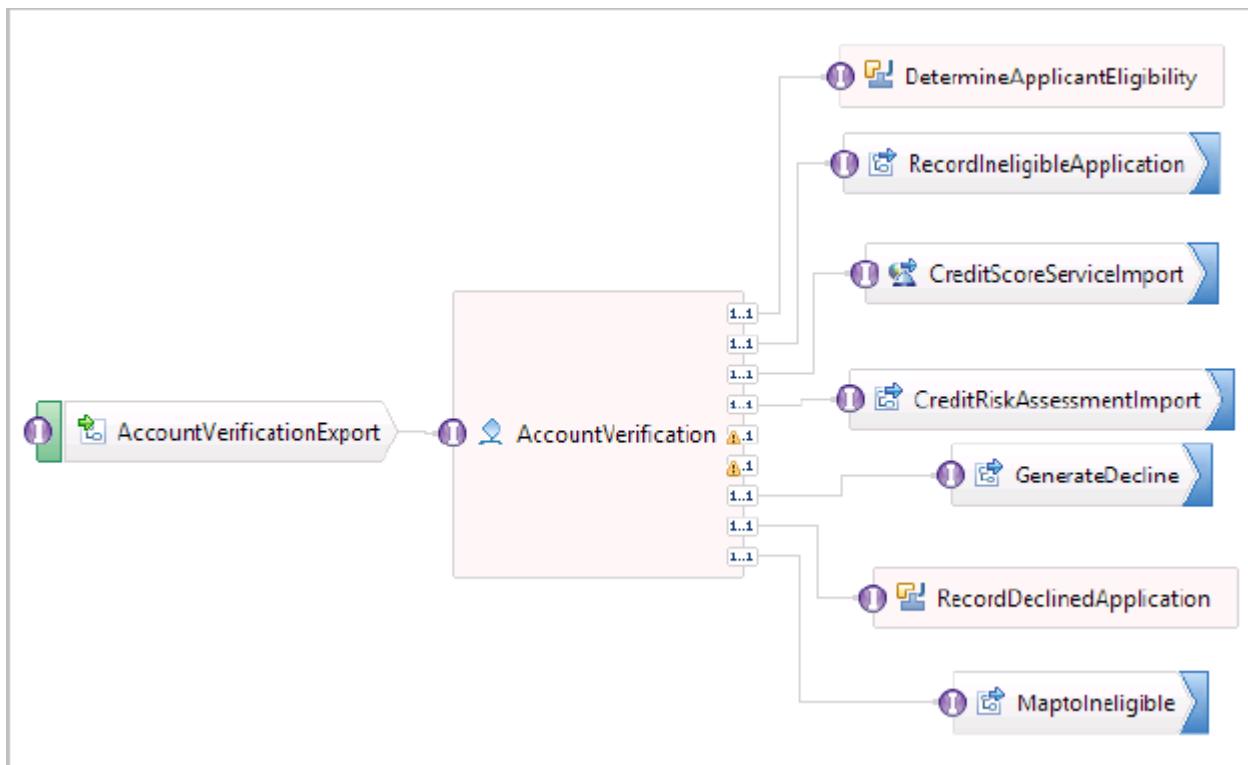


- \_\_\_ 2. Wire the **GenerateDeclinePartner** reference on the **AccountVerification** process component to the newly created **GenerateDecline** import.

- \_\_\_ a. Right-click **GenerateDecline** and click **Wire to Existing** from the menu.
- \_\_\_ b. Verify the wiring by hovering over the reference component that is wired to **GenerateDecline**. The dialog box lists: **GenerateDeclinePartner**



- \_\_\_ c. Press Ctrl+S to save the **FoundationModule** assembly diagram. Your assembly diagram resembles the following figure:



- \_\_\_ 3. Close the assembly editor.

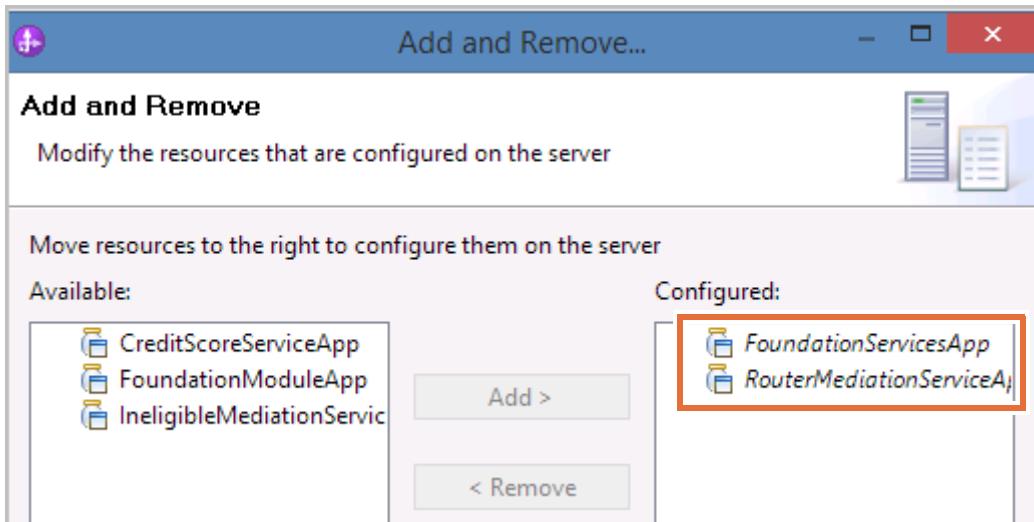
## Testing the mediation module

In this portion of the exercise, you test the RouterMediationService mediation module. When an application passes through FinalApplicationReview, the business process follows either the “approved” path or the “declined” path, which depends on the value in applicationDecision.

If applicationDecision is `true` (the application was approved), the process proceeds to the AccountVerificationReply activity. If applicationDecision is `false` (the application was declined), the process proceeds to the GenerateDecline activity. GenerateDecline processes the declined application and bases its action on the value in the creditRisk attribute. If creditRisk is `MED`, then SpecialDeclineService is called. If creditRisk is `HIGH`, then GenerateDeclineService is called.

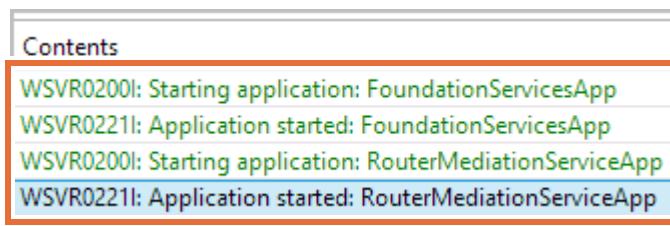
To test the **RouterMediationService** mediation module:

- 1. Start the server (if it is not already running) and deploy **RouterMediationServiceApp** and **FoundationServicesApp**.
  - a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Wait for the startup process to complete before continuing.  
The server is started when the message Server server1 open for e-business is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
  - b. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
  - c. In the **Available** list, double-click **RouterMediationServiceApp** and **FoundationServicesApp** to add them to the **Configured** list.



- d. Click **Finish**.

- \_\_\_ e. When the applications are deployed and started, the messages Application started: RouterMediationServiceApp and Application started: FoundationServicesApp are listed in the **Server Logs** view.



- \_\_\_ f. If any module has a **Stopped** status, then right-click the module and click **Restart** from the menu. If prompted to do so, republish the module. Wait for the server status to change to **Started, Synchronized**. If the server has a status of **Started, Publishing**, then clicking the server refreshes the status to **Started, Synchronized**.
- \_\_\_ 2. Test the RouteRequest flow component by invoking the RouteRequestExport. Test the SpecialDeclineService path by setting applicationDecision to `false` and creditRisk to `MED`. Use the sample values in `C:\labfiles\Support Files\Ex12\ EX12_Test_Data.xml`.
- \_\_\_ a. In the Business Integration view, expand **RouterMediationService** and double-click **Assembly Diagram**.
- \_\_\_ b. Right-click **RouteRequestExport** and click **Test Component** from the menu.
- \_\_\_ c. When the integrated test client opens, in the **Initial request parameters** section, right-click **Input**, click **Import from File** from the menu, select `C:\labfiles\Support Files\Ex12\EX12_Test_Data.xml`, and click **Open**.

The input parameters are populated with the required test data. The value for applicationDecision is false and creditRisk is MED. They test the SpecialDeclineService.

| Name                 | Type                |                    |
|----------------------|---------------------|--------------------|
| Input                | CustomerApplication | [ab]               |
| accountNumber        | string              | [ab] ACM002        |
| applicationDate      | string              | [ab] 06/10/2014    |
| applicationDecision  | boolean             | [ab] false         |
| comments             | string              | [ab] Some comments |
| companyName          | string              | [ab] ACME          |
| contactFirstName     | string              | [ab] Torsten       |
| contactLastName      | string              | [ab] Frings        |
| contactPhoneNumber   | string              | [ab] 905-555-7234  |
| creditRating         | string              | [ab] A++           |
| creditReportNeeded   | boolean             | [ab] true          |
| creditRisk           | string              | [ab] MED           |
| creditScore          | int                 | [ab] 0             |
| customerCity         | string              | [ab] Berlin        |
| customerCountry      | string              | [ab] Germany       |
| eligibleApplication  | boolean             | [ab] true          |
| ineligibleReason     | string              | [ab] None          |
| pricingCode          | string              | [ab] 23            |
| pricingScore         | string              | [ab] 32            |
| productName          | string              | [ab] Labels        |
| requestAccountAmount | int                 | [ab] 10000         |

- \_\_\_ d. Alternatively, enter the following test data in the **Initial request parameters** section (verify that the data matches the XML file):

- accountNumber: ACM002
- applicationDate: 06/10/2014
- applicationDecision: false
- comments: Some comments
- companyName: ACME
- contactFirstName: Torsten
- contactLastName: Frings
- contactPhoneNumber: 905-555-7234
- creditRating: A++
- creditReportNeeded: true
- creditRisk: MED
- creditScore: 0
- customerCity: Berlin
- customerCountry: Germany
- eligibleApplication: true
- ineligibleReason: None
- pricingCode: 23
- pricingScore: 32
- productName: Labels
- requestAccountAmount: 10000

- \_\_\_ e. Click the **Continue** icon to start the test thread.
  - \_\_\_ f. In the “Select a Deployment Location” dialog box, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
  - \_\_\_ g. In the User Login dialog box, accept the default entries for **User ID** and **Password** and click **OK**. These fields are set to `admin` and `web1sphere` by default.
  - \_\_\_ h. The test run is complete when the blue, square stop node is displayed in the Events window.
3. Examine the test trace and the log messages.
- \_\_\_ a. Switch to the **Server Logs** view.
  - \_\_\_ b. The following messages are displayed in the **Server Logs** view. They indicate that the application was sent to the special decline service.

Generate Decline Special - begins

Generate Decline Special - Account for customer ACME was routed through  
special decline because the credit risk was MED

Generate Decline Special - ends

#### Contents

WSVR0200I: Starting application: FoundationServicesApp

WSVR0221I: Application started: FoundationServicesApp

WSVR0200I: Starting application: RouterMediationServiceApp

WSVR0221I: Application started: RouterMediationServiceApp

WSVR0217I: Stopping application: RouterMediationServiceApp

WSVR0220I: Application stopped: RouterMediationServiceApp

WSVR0200I: Starting application: RouterMediationServiceApp

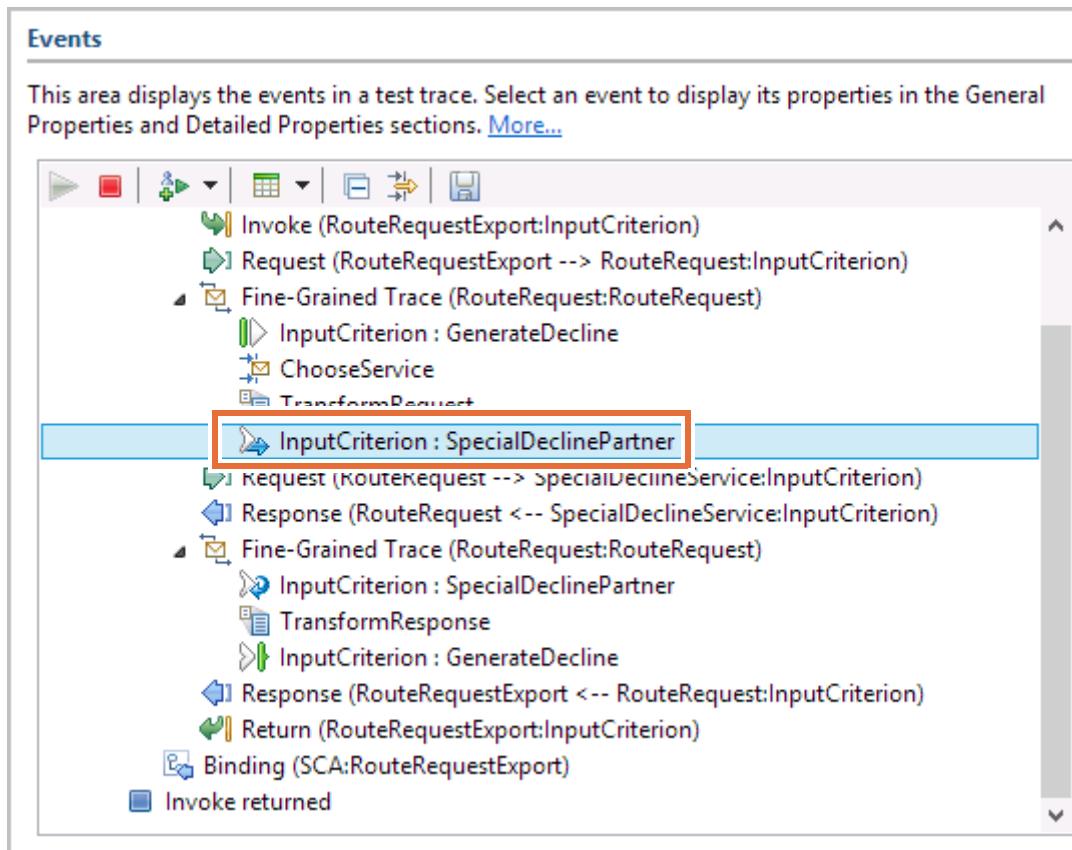
WSVR0221I: Application started: RouterMediationServiceApp

[Java] Generate Decline Special - begins

[Java] Generate Decline Special - Account for customer ACME was routed through special decline because the credit risk was MED

[Java] Generate Decline Special - ends

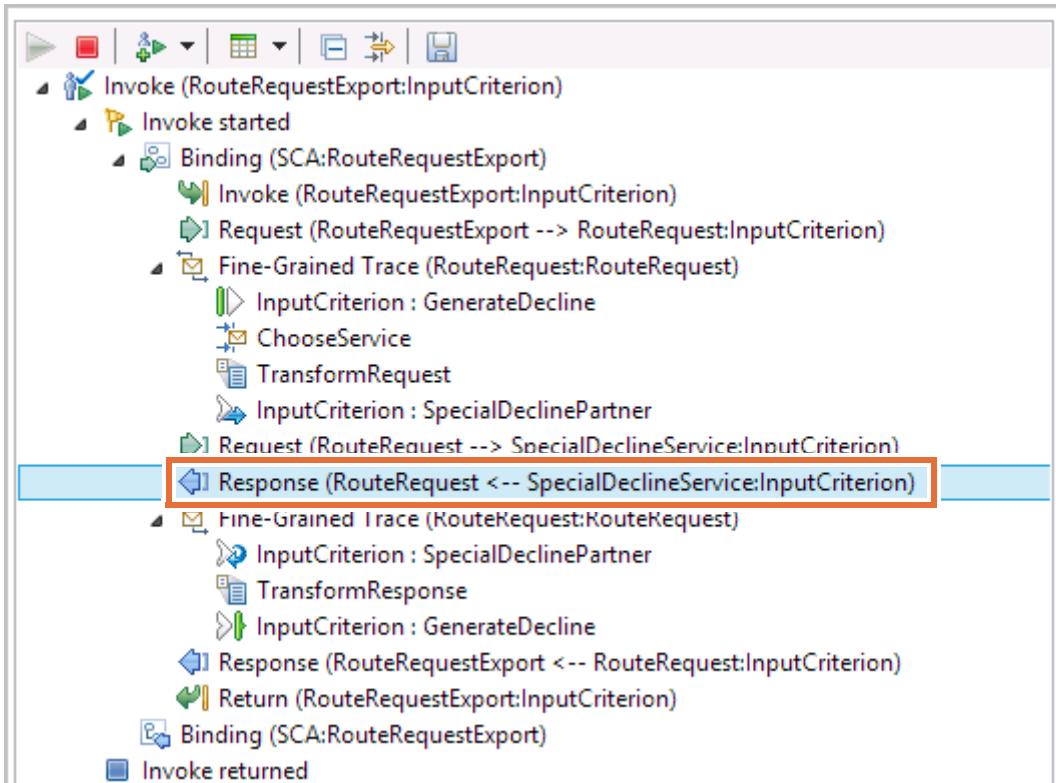
- c. In the Events window of the test client, select the first instance of the **InputCriterion : SpecialDeclinePartner** callout event. The **SpecialDeclinePartner** service is invoked when **applicationDecision** is **false** and **creditRisk** is **MED**.



- \_\_\_ d. In the **Mediation Message** section, examine the data in the **body** of the message. You might find it necessary to collapse the headers section to see the data. The XML map successfully transformed the message by moving all of the attribute values.

| Name                 | Type                     | Value              |
|----------------------|--------------------------|--------------------|
| context              | ContextType              | [ab]               |
| headers              | HeadersType              | [ab]               |
| body                 | InputCriterionRequestMsg | [ab]               |
| InputCriterion       | InputCriterion_.type     | [ab]               |
| Input                | CustomerApplication      | [ab]               |
| pricingCode          | String                   | [ab] 23            |
| companyName          | String                   | [ab] ACME          |
| applicationDecision  | boolean                  | [ab] false         |
| pricingScore         | String                   | [ab] 32            |
| contactFirstName     | String                   | [ab] Torsten       |
| requestAccountAmount | int                      | [ab] 10000         |
| ineligibleReason     | String                   | [ab] None          |
| accountNumber        | String                   | [ab] ACM002        |
| creditReportNeeded   | boolean                  | [ab] true          |
| eligibleApplication  | boolean                  | [ab] true          |
| creditRating         | String                   | [ab] A++           |
| contactLastName      | String                   | [ab] Frings        |
| creditScore          | int                      | [ab] 0             |
| customerCity         | String                   | [ab] Berlin        |
| applicationDate      | String                   | [ab] 06/10/2014    |
| productName          | String                   | [ab] Labels        |
| creditRisk           | String                   | [ab] MED           |
| comments             | String                   | [ab] Some comments |

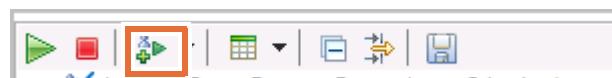
- \_\_ e. In the Events window, select **Response (RouteRequest <-- SpecialDeclineService:InputCriterion)**.



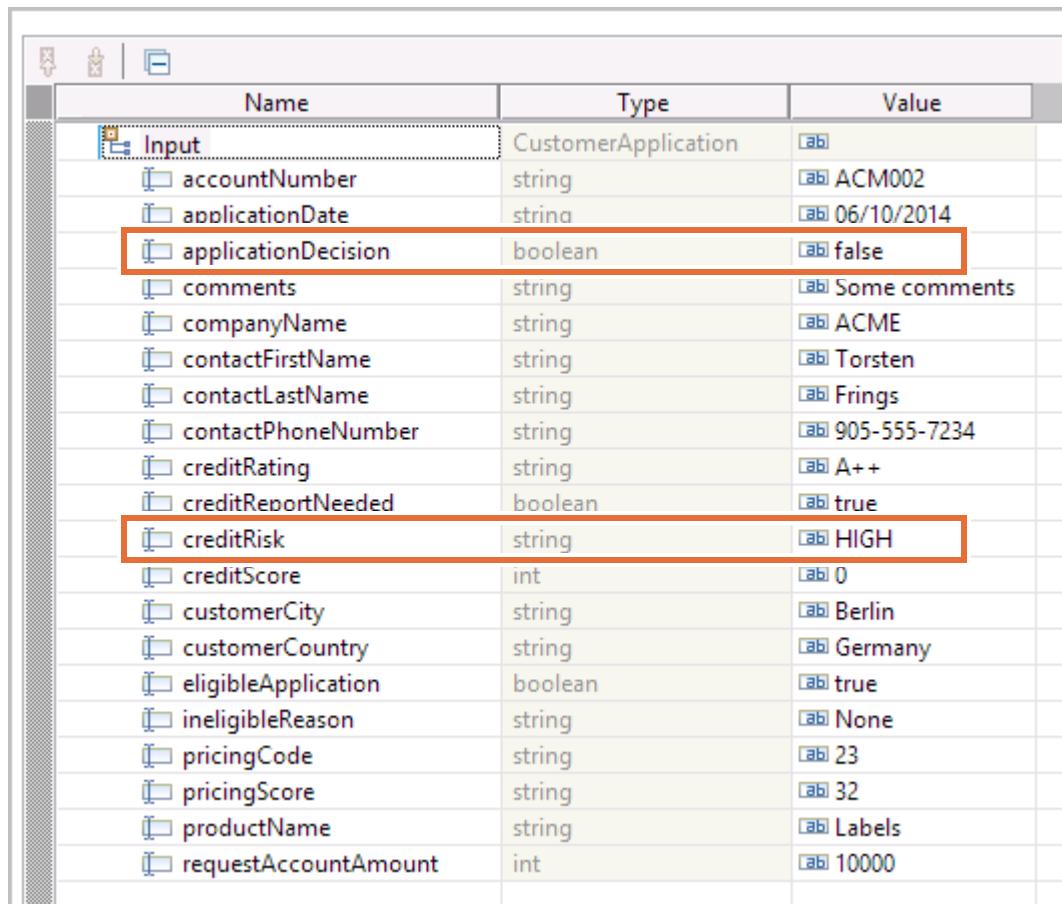
- \_\_ f. In the **Response parameters** section, examine the output message. You can resize the columns to view the message. The message reads: Account for customer ACME was routed through special decline because the credit risk was MED

|     | Value                                                                                        |
|-----|----------------------------------------------------------------------------------------------|
| abi |                                                                                              |
| lab | Account for customer ACME was routed through special decline because the credit risk was MED |

- \_\_ 4. Invoke the test a second time. Test the **GenerateDeclineService** path by setting **applicationDecision** to `false` and **creditRisk** to `HIGH`. Use the sample values in `C:\labfiles\Support Files\Ex12\EX12_Test_Data2.xml`.
- \_\_ a. In the Events window, click the **Invoke** icon on the action bar.



- \_\_\_ b. In the **Initial request parameters** section, right-click **Input**, click **Import from File** from the menu, select C:\labfiles\Support Files\Ex12\EX12\_Test\_Data2.xml, and click **Open**. The input parameters are populated with the required test data. The value for **applicationDecision** is **false** and **creditRisk** is **HIGH**. They test the GenerateDeclineService.



| Name                 | Type                | Value         |
|----------------------|---------------------|---------------|
| CustomerApplication  | CustomerApplication |               |
| accountNumber        | string              | ACM002        |
| applicationDate      | string              | 06/10/2014    |
| applicationDecision  | boolean             | false         |
| comments             | string              | Some comments |
| companyName          | string              | ACME          |
| contactFirstName     | string              | Torsten       |
| contactLastName      | string              | Frings        |
| contactPhoneNumber   | string              | 905-555-7234  |
| creditRating         | string              | A++           |
| creditReportNeeded   | boolean             | true          |
| creditRisk           | string              | HIGH          |
| creditScore          | int                 | 0             |
| customerCity         | string              | Berlin        |
| customerCountry      | string              | Germany       |
| eligibleApplication  | boolean             | true          |
| ineligibleReason     | string              | None          |
| pricingCode          | string              | 23            |
| pricingScore         | string              | 32            |
| productName          | string              | Labels        |
| requestAccountAmount | int                 | 10000         |

- \_\_\_ c. Alternatively, enter the following test data in the **Initial request parameters** section:
- accountNumber: ACM002
  - applicationDate: 06/10/2014
  - applicationDecision: false
  - comments: Some comments
  - companyName: ACME
  - contactFirstName: Torsten
  - contactLastName: Frings
  - contactPhoneNumber: 905-555-7234
  - creditRating: A++
  - creditReportNeeded: true
  - creditRisk: HIGH
  - creditScore: 0
  - customerCity: Berlin
  - customerCountry: Germany
  - eligibleApplication: true
  - ineligibleReason: None
  - pricingCode: 23
  - pricingScore: 32
  - productName: Labels
  - requestAccountAmount: 10000
- \_\_\_ d. In the Events window, click the **Continue** icon in the action bar.
- \_\_\_ e. The test run is complete when the blue, square stop node is displayed in the Events window.
- \_\_\_ 5. Examine the test trace and the log messages.
- \_\_\_ a. Switch to the **Server Logs** view.

- \_\_\_ b. The following messages are displayed in the **Server Logs** view. They indicate that the application was sent to the special decline service:

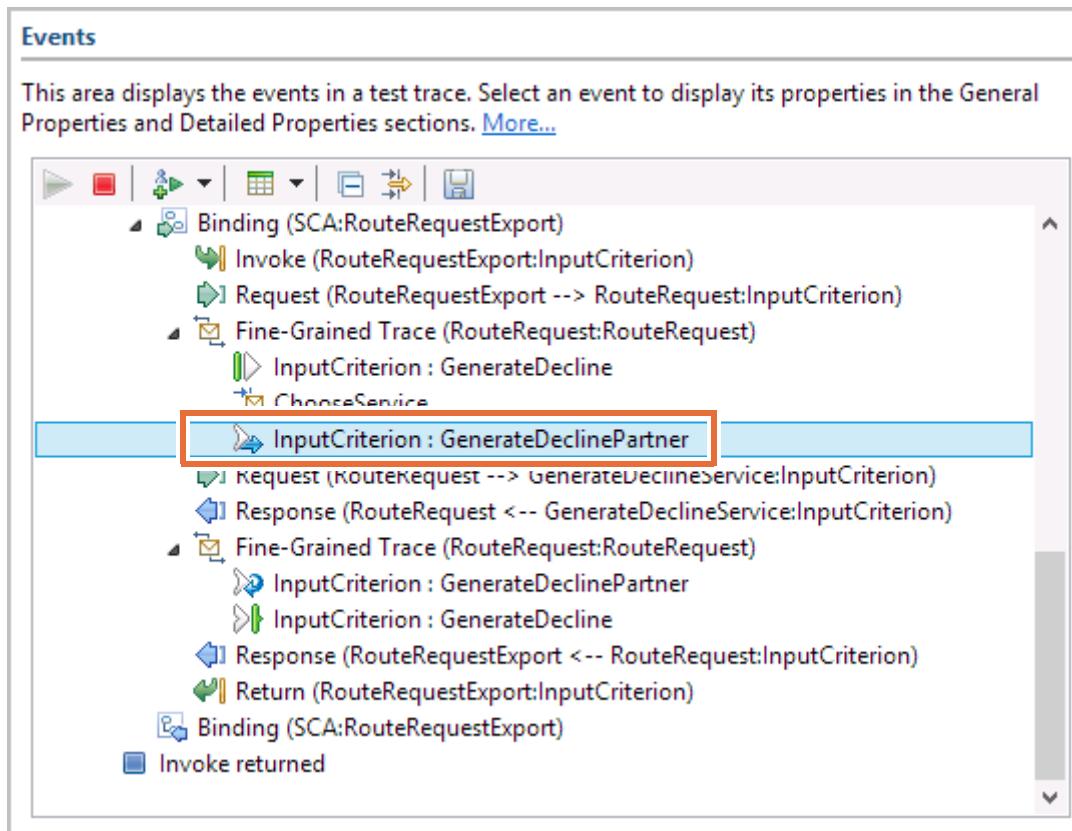
Generate Decline - begins

Generate Decline Special - Account for customer ACME was declined and the credit risk was HIGH

Generate Decline - ends

| Contents                                                                                             |
|------------------------------------------------------------------------------------------------------|
| WSVR0200I: Starting application: FoundationServicesApp                                               |
| WSVR0221I: Application started: FoundationServicesApp                                                |
| WSVR0200I: Starting application: RouterMediationServiceApp                                           |
| WSVR0221I: Application started: RouterMediationServiceApp                                            |
| WSVR0217I: Stopping application: RouterMediationServiceApp                                           |
| WSVR0220I: Application stopped: RouterMediationServiceApp                                            |
| WSVR0200I: Starting application: RouterMediationServiceApp                                           |
| WSVR0221I: Application started: RouterMediationServiceApp                                            |
| [Java] Generate Decline Special - begins                                                             |
| [Java] Generate Decline Special - Account for customer ACME was routed through special decline       |
| [Java] Generate Decline Special - ends                                                               |
| [Java] Generate Decline - begins                                                                     |
| <b>[Java] Generate Decline - Account for customer ACME was declined and the credit risk was HIGH</b> |
| [Java] Generate Decline - ends                                                                       |

- c. In the Events window of the test client, select the **InputCriterion: GenerateDeclinePartner** callout event. The GenerateDeclinePartner service is invoked when applicationDecision is `false` and creditRisk is `HIGH`.

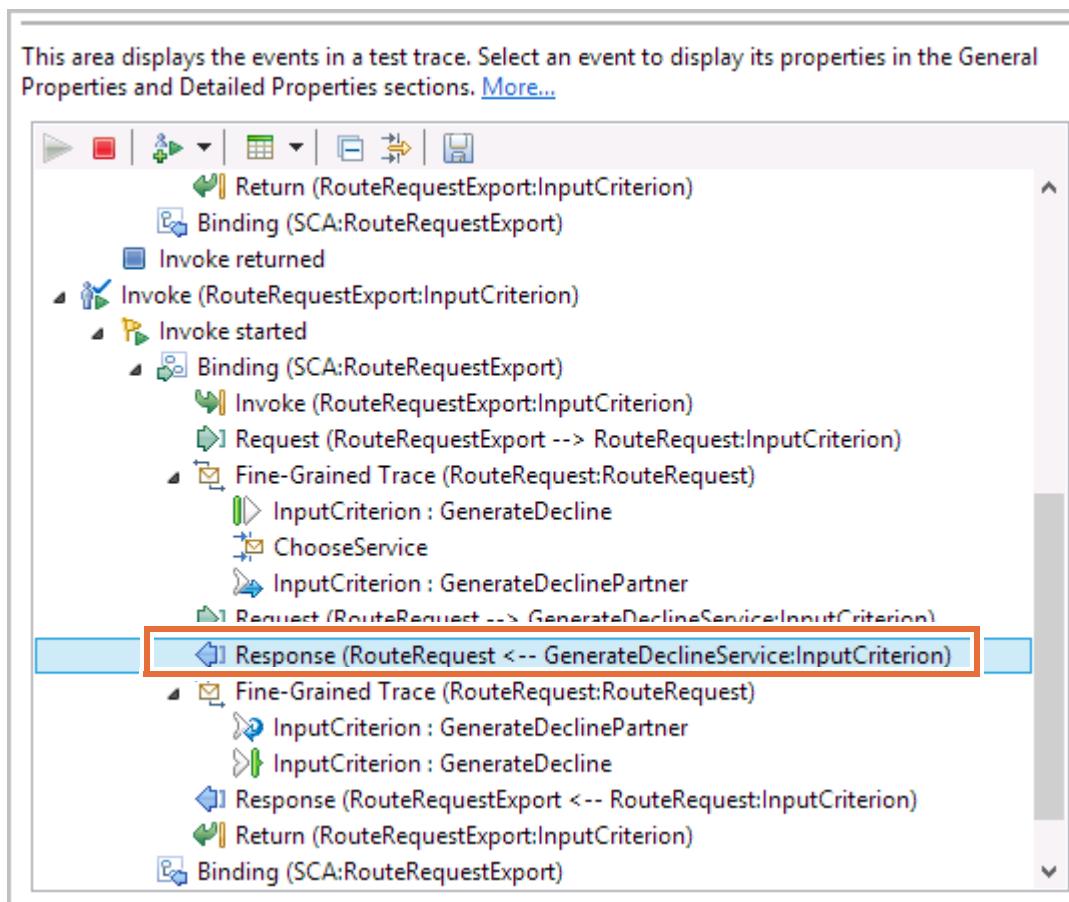


- \_\_\_ d. In the **Mediation Message** section, examine the data in the body of the message. You can collapse the headers section to see the data. The XML map successfully transformed the message by moving all of the attribute values.

The screenshot shows a mediation message editor interface. On the left is a tree view of message components: 'headers' (HeadersType) and 'body' (InputCriterionRequestMsg). The 'body' node has a child 'InputCriterion' (InputCriterion\_type), which in turn has a child 'Input'. The 'Input' node contains several data items: 'pricingCode' (String, value 23), 'companyName' (String, value ACME), 'applicationDecision' (boolean, value false), 'pricingScore' (String, value 32), 'contactFirstName' (String, value Torsten), 'requestAccountAmount' (int, value 10000), 'ineligibleReason' (String, value None), 'accountNumber' (String, value ACM002), 'creditReportNeeded' (boolean, value true), 'eligibleApplication' (boolean, value true), 'creditRating' (String, value A++), 'contactLastName' (String, value Frings), 'creditScore' (int, value 0), 'customerCity' (String, value Berlin), 'applicationDate' (String, value 06/10/2014), 'productName' (String, value Labels), and 'creditRisk' (String, value HIGH).

| Name                 | Type                     | Value           |
|----------------------|--------------------------|-----------------|
| headers              | HeadersType              | [ab]            |
| body                 | InputCriterionRequestMsg | [ab]            |
| InputCriterion       | InputCriterion_type      | [ab]            |
| Input                | CustomerApplication      | [ab]            |
| pricingCode          | String                   | [ab] 23         |
| companyName          | String                   | [ab] ACME       |
| applicationDecision  | boolean                  | [ab] false      |
| pricingScore         | String                   | [ab] 32         |
| contactFirstName     | String                   | [ab] Torsten    |
| requestAccountAmount | int                      | [ab] 10000      |
| ineligibleReason     | String                   | [ab] None       |
| accountNumber        | String                   | [ab] ACM002     |
| creditReportNeeded   | boolean                  | [ab] true       |
| eligibleApplication  | boolean                  | [ab] true       |
| creditRating         | String                   | [ab] A++        |
| contactLastName      | String                   | [ab] Frings     |
| creditScore          | int                      | [ab] 0          |
| customerCity         | String                   | [ab] Berlin     |
| applicationDate      | String                   | [ab] 06/10/2014 |
| productName          | String                   | [ab] Labels     |
| creditRisk           | String                   | [ab] HIGH       |

- \_\_\_ e. In the Events window, select **Response (RouteRequest <-- GenerateDeclineService:InputCriterion)**.



- \_\_\_ f. In the **Response parameters** section, examine the output message. The message reads: Account for customer ACME was declined and the credit risk was HIGH.

| Name    | Type    | Value                                                               |
|---------|---------|---------------------------------------------------------------------|
| Output  | Message |                                                                     |
| message | string  | Account for customer ACME was declined and the credit risk was HIGH |

- \_\_\_ 6. Close the **RouterMediationService\_test** tab and click **No** when you are prompted to save the test trace.
- \_\_\_ 7. Remove all of the projects and stop the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
- \_\_\_ b. Click **Remove All**.
- \_\_\_ c. Click **Finish**.
- \_\_\_ d. Optionally, stop the server.
- \_\_\_ 8. Close IBM Integration Designer.

## **End of exercise**

## Exercise review and wrap-up

In this exercise, you created a mediation module that contained a message filter primitive and a Mapping primitive and then tested the module in the IBM Integration Designer test environment.

# Exercise 13.Exploring Business Space

## What this exercise is about

In this exercise, you explore the capabilities of Business Space by using templates, pages, and widgets.

## What you should be able to do

After completing this exercise, you should be able to:

- Create a space by using the Business Space client
- Use the Business Space client to work with human tasks
- Create a page and display content by using widgets and Business Space templates

## Introduction

Business Space is an integrated user experience for business users across the IBM Business Process Manager portfolio. Business Space provides a customizable and collaborative environment for you to monitor, review, and administer common business processes, such as human task flows, modeling, and performance indicators.

Business Space is a browser-based graphical user interface that you use to view and interact with content from various products in the business process management portfolio. Business Space not only provides a single web-based point of access for the content, you use Business Space to combine the content in useful and interesting ways. These combinations give you insight into your business and the capability to react to changes in it.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

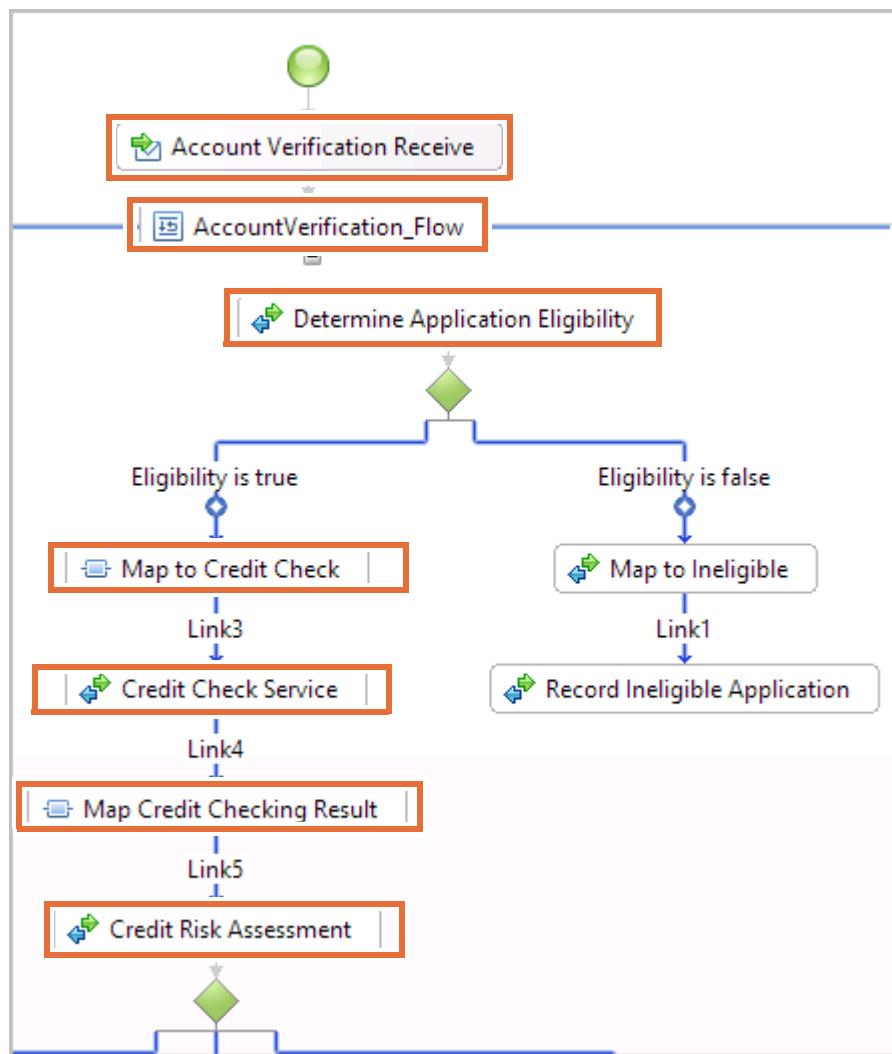
## Exercise instructions

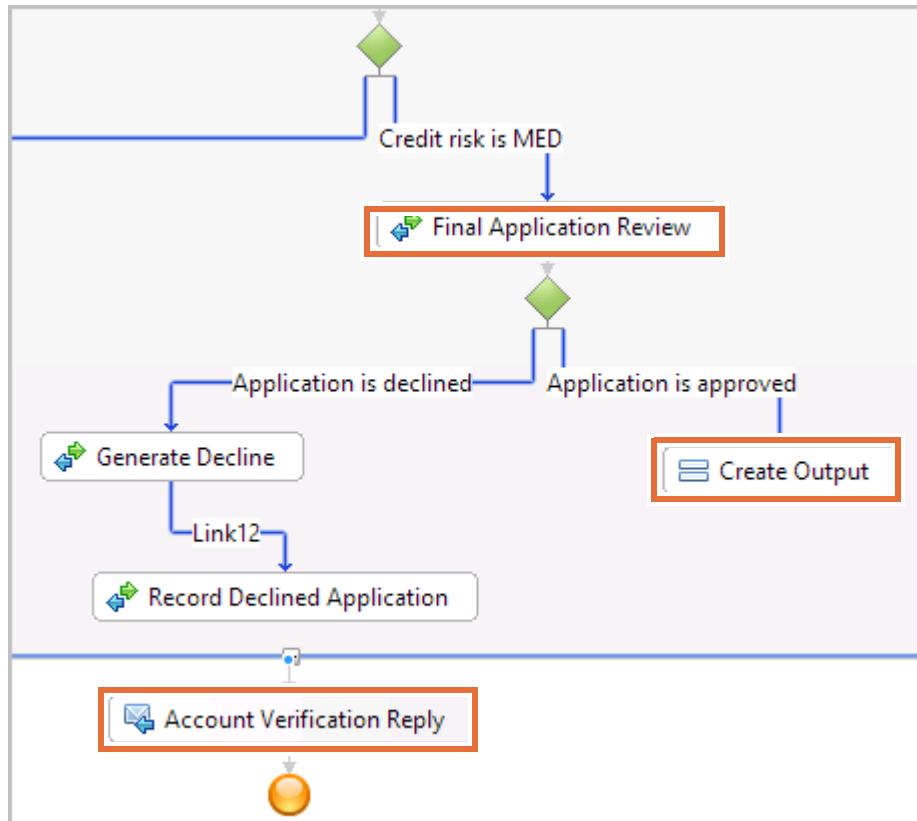
### Part 1: Use the Business Space clients to work with human tasks

In the workspace for this exercise, the entire “create account” solution is assembled, and you can run end-to-end tests of the applications. You can use different possible paths through the AccountVerification business process. You run a test to verify a test case to work with the Business Space client. You must familiarize yourself with the overall project to understand how data flows through the applications. For clarification, see the scenario overview in the appendixes.

### Use Business Space to test for eligible applications with MED credit risk

When you use company name `ACME` to test the applications, the `eligibleApplication` attribute is set to `true` and the `creditRisk` evaluates to `MED`. The `creditScore` returned is `6`. When you use `companyName ACME` to submit an application, the application flows through these activities: **Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment > Final Application Review > Create Output > Account Verification Reply**.





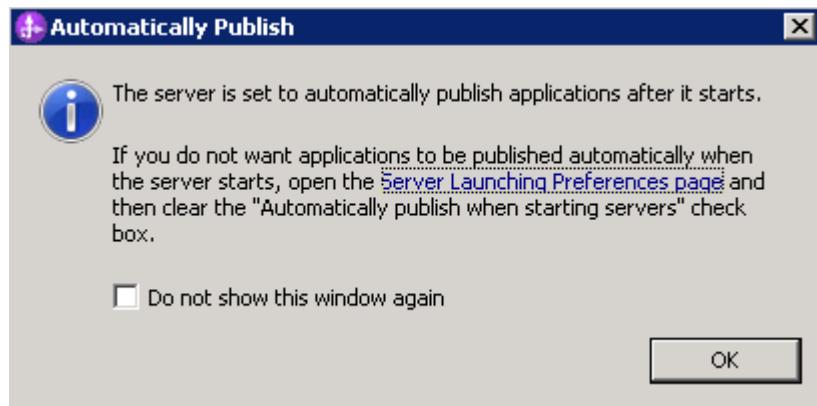
A user interface for Final Application Review is used to update the applicationDecision field. If applicationDecision is `true`, the application flows to **Create Output > Account Verification Reply**.

To test an eligible application with a `MED` credit risk:

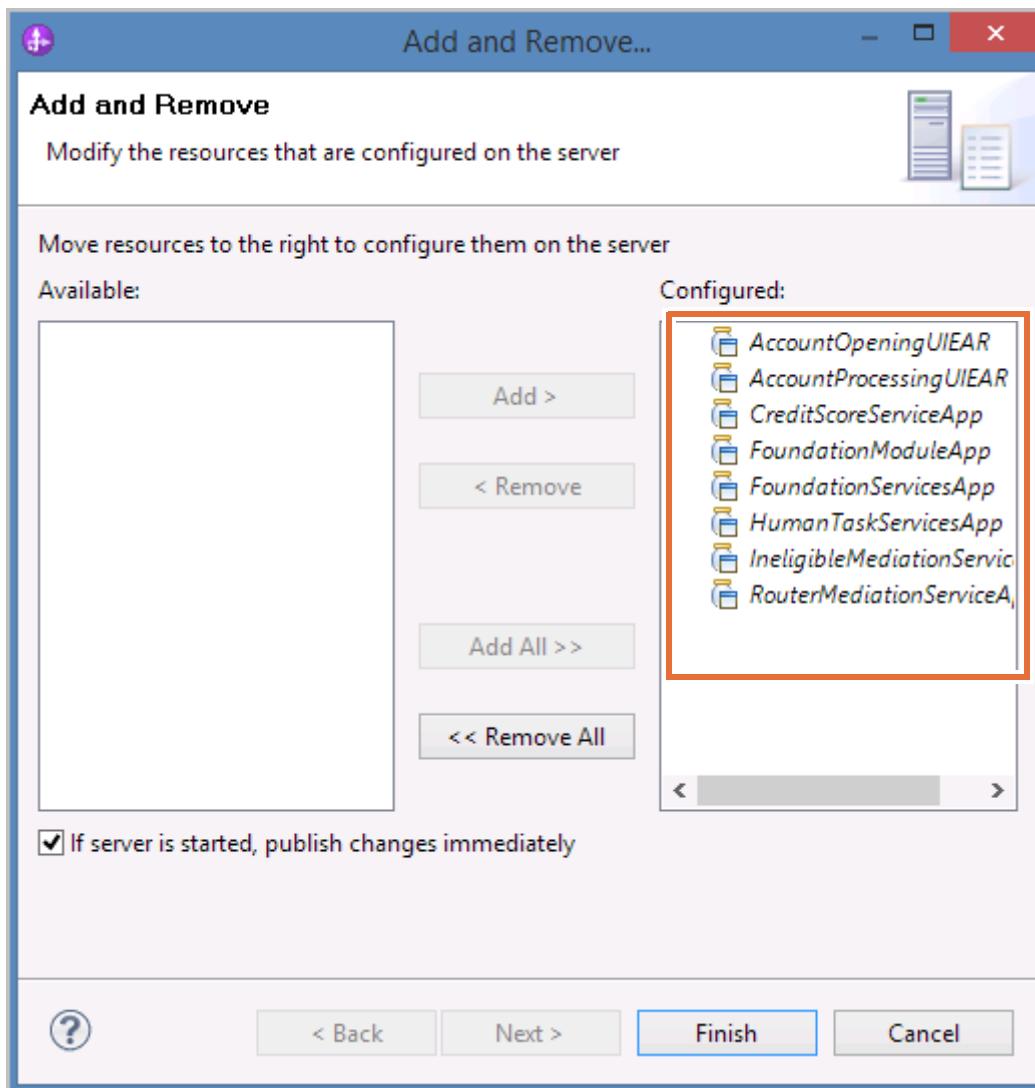
- 1. Open the Exercise 13 workspace.
- a. On your desktop, open the **Exercise Shortcuts** folder.
- b. Double-click the **Exercise 13** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
- c. Close the **Getting Started** tab.
- 2. Start the server (if it is not already running) and deploy all of the modules.
- a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Wait for the startup process to complete before continuing.

The server is started when the message `Server server1 open for e-business` is displayed in the **Server Logs** view. The server status also changes to **Started** in the **Servers** view.

- \_\_\_ b. If you receive a message that the server is automatically set to publish after it starts, then click **OK** to close the dialog box.



- \_\_\_ c. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.
- \_\_\_ d. Click **Add All** to move each of the projects to the **Configured** list.



- e. Click **Finish**.
- f. Wait until the modules are published and started. It can take several minutes, depending on your system resources. To see the status of the modules, expand **IBM Process Server v8.5.7 at localhost** in the **Servers** view. Verify that all modules started.

## Part 2: *Creating a business space (your own workspace) in the Business Space user interface*

In Business Space, you can have many business spaces with each one having a different purpose. For example, a business space with widgets from IBM Business Monitor can be used to monitor key performance indicators in your business. Similarly, widgets from IBM Business Process Manager can be used to manage the tasks that people do. Business Space can display the contents of one business space at a time. This space is the open space.

Before you can use the Business Space client to work with your human tasks, you must configure a user workspace. You use this workspace, and the available human task widgets, to claim and complete work items. You can also use Business Space to handle escalations and to work with the business calendars used by your human tasks.



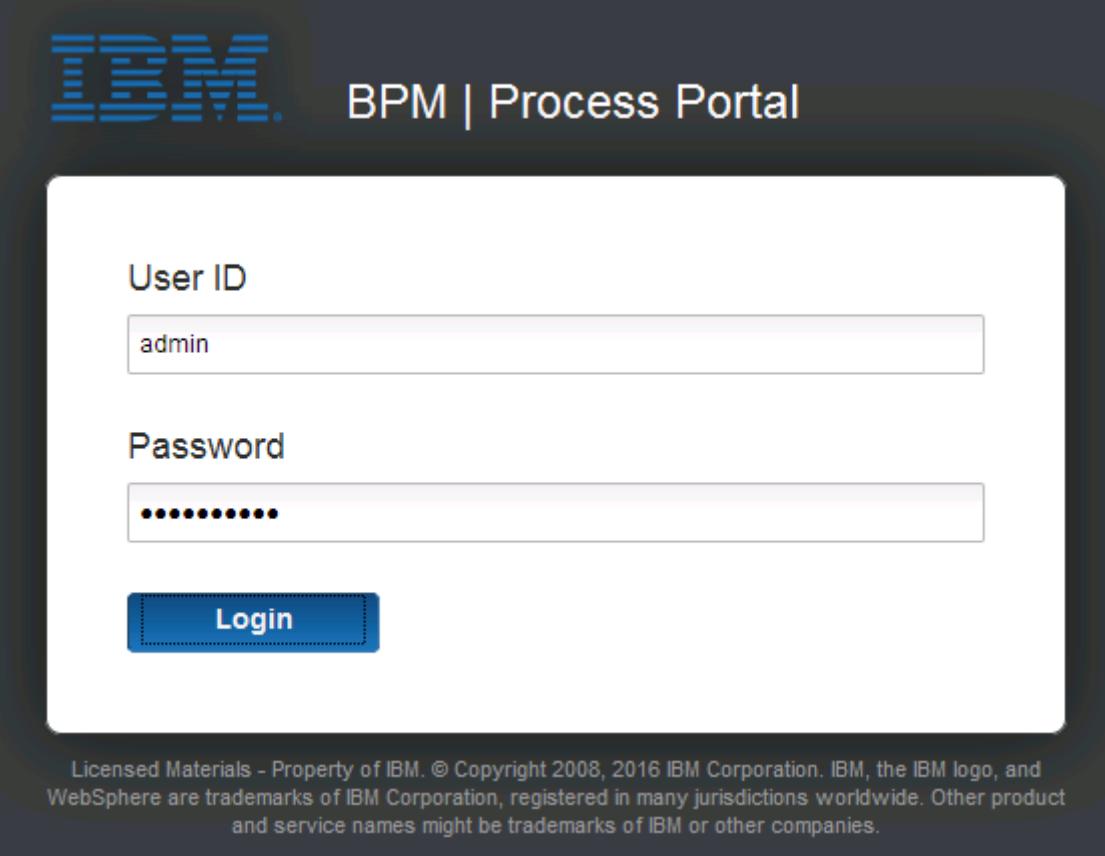
**Stop**

It is important to use Firefox as the browser to view the Business Space content for this exercise. During the creation of this lab, the business space client was missing space styles if Internet Explorer was used.

To create a user space in Business Space:

1. Use the Business Space client to create a business space that is named: `MyBusinessSpace`
  - a. Start Firefox and enter `https://localhost:9443/BusinessSpace` in the location bar.
  - b. If the message is displayed that the connection is untrusted, then click **I understand the Risks**.
  - c. Click **Add Exception**.
  - d. In the Add Security Exception window, click **Confirm Security Exception**.

- \_\_ e. At the **Login** prompt, enter `admin` in the **User ID** field and `websphere` in the **Password** field.



The image shows the IBM BPM Process Portal login screen. At the top left is the IBM logo. To its right, the text "BPM | Process Portal" is displayed. Below this is a white rectangular input area. Inside, the label "User ID" is above a text input field containing "admin". Below it, the label "Password" is above another text input field containing "\*\*\*\*\*". At the bottom of this area is a blue "Login" button. At the very bottom of the page, there is a small legal notice: "Licensed Materials - Property of IBM. © Copyright 2008, 2016 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."

- \_\_ f. Click **Login**. Wait a few moments for the page to load.

 **Troubleshooting**

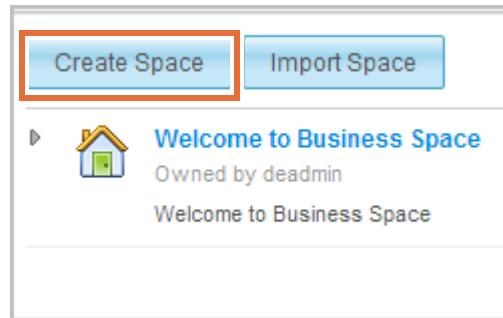
During writing of this lab, sometimes, the Welcome page image did not display or render correctly. If that occurs, then it is OK to ignore the missing Welcome page, and you can continue with the exercise.

---

- \_\_ g. Click the **Manage Spaces** link in the action bar.



- \_\_ h. Create a workspace by clicking **Create Space**.



- \_\_ i. In the **Create Space** dialog box, take the following actions:

- In the **Space Name** field, type: MyBusinessSpace
- Click the **Create a new space using a template** option.
- Select **Advanced Task and Process Management**.

 A screenshot of the "Create Space" dialog box. It has fields for "Space name" (containing "MyBusinessSpace") and "Space description". Below these, there are three options:
 

- Create a blank space
- Create a new space using a template (highlighted with a red box)
- Duplicate an existing space

 The "Create a new space using a template" option has a dropdown menu open, showing "Advanced Task and Process Management" (also highlighted with a red box).

With the preconfigured templates that come with Business Space, you can create a business space that contains pages and widgets. The template option lists the available templates within IBM Process Manager product. Two types of templates are available:

- Product templates have widgets from a single product. The Create Space window lists only the product templates for the products that are installed.
- Cross-product templates have widgets from more than one product.

In this exercise, you use the **Advanced task and Process Management** template to work on the assigned tasks.

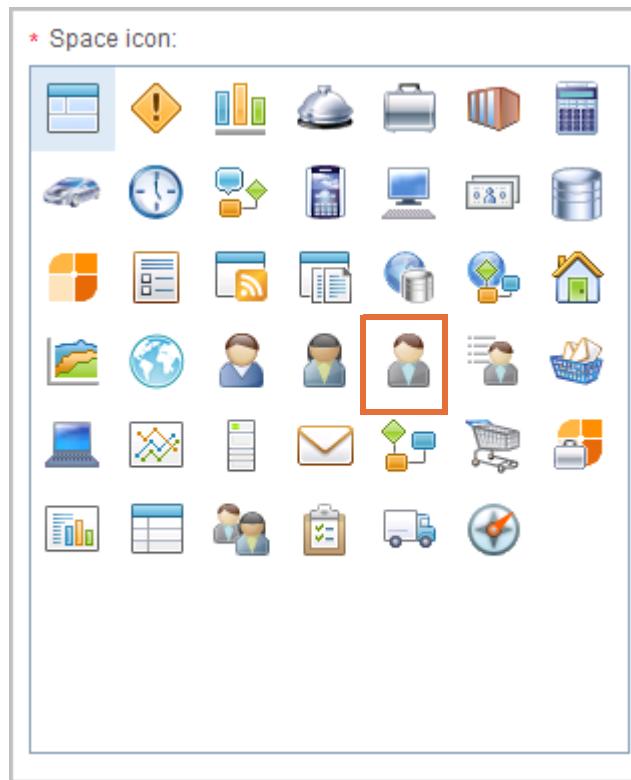
Use the Advanced Management of Tasks and Processes template to create a space for business users. Team leads can also use this template. These team leads work on tasks, collaborate with other people, organize their own task lists, and check the status of processes and related tasks that they are involved with or are responsible for.

This template contains widgets from IBM Business Process Manager. It contains widgets and capabilities that are available only with the Business Process Choreographer engine. The widgets do not support the business process definition engine or federated process engines.

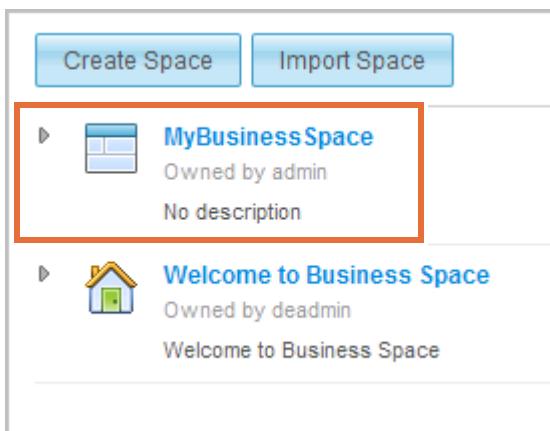
- j. In the **Space style** window, select **Human** as the **Space style**. A style determines the color and appearance of a business space. You can use a particular style, for example, to indicate the purpose of the space. Make sure that you are using Firefox as the browser or else the Space styles are not listed.



- k. You can also choose an icon to represent the business space. In the **Space icon** window, select an icon of your choice.

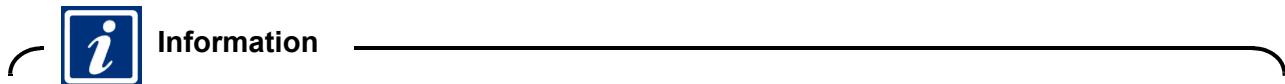


- l. Click **Save** at the bottom of the window. The `MyBusinessSpace` workspace is added to the list of spaces.
- 2. Explore the newly created `MyBusinessSpace` space.
- a. Click **MyBusinessSpace**.

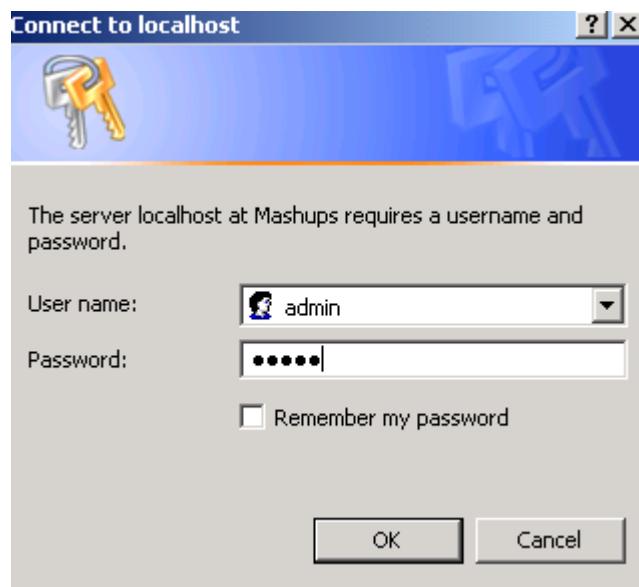


- \_\_ b. Verify that **MyBusinessSpace** opens in the Business Space. Five tabs or pages are created for you by default. These tabs are created because you selected the **Advanced task and Process Management** template during the space creation. This template creates a space with the following pages:

- Initiate Tasks and Processes
- Work on Tasks
- Manage Processes
- Manage Tasks
- Organize Work



If you receive a login window, then log back in the Business Space by entering `admin` in the **User name** field and `web1sphere` in the **Password** field and clicking **OK**.



- \_\_ c. If it is not already selected, select the **Initiate tasks and Processes** tab. On the **Initiate Tasks and Processes** page, you can initiate tasks, processes, and services. Additionally, you can also store incomplete forms as drafts, and check the status of initiated tasks and processes.

A screenshot of the IBM BPM Advanced V8.5.7 - I interface. The top navigation bar includes 'Home', 'Process Portal', 'Go to Spaces', 'Manage Spaces', and 'Actions'. The 'MyBusinessSpace' page is displayed. A horizontal menu bar below the title contains five tabs: 'Initiate Tasks and Processes' (which is highlighted with a red border), 'Work on tasks', 'Manage Processes', 'Manage Tasks', and 'Organize Work'. The main content area is divided into two sections: 'Task Definitions' on the left and 'Task Information' on the right. The 'Task Definitions' section shows a table with columns for 'Name', 'Type', 'Status', and 'Last Modified'. The first row is selected. The 'Task Information' section contains a message: 'Select the task and then s...'. At the bottom of the page, there are links for 'Help', 'Logout', and 'IBM BPM Advanced V8.5.7 - I'.

- \_\_\_ d. Note the different widgets that are already available on the page. The four widgets that are displayed are:
- **Task Definitions:** Displays a list of task definitions that you use to create a task or to initiate services and processes.
  - **Task Information:** Displays information about a task that you selected in the Status widget or the Task Definitions widget. You can check and change the priority and due date of a task.
  - **Status:** Displays the tasks for the definitions that you selected in the Task Definitions widget. You can check and change the priority and due date of a task, or postpone work on a task and then resume it later.
  - **Process Information:** Displays information that is associated with the task or process.
- \_\_\_ e. The **Task Definitions** widget lists several available tasks for the user. The number of tasks in the widgets in your lab image might be different from the tasks that are listed in your lab image.

The screenshot shows a software interface titled "Task Definitions". At the top, there is a toolbar with icons for search, refresh, and close. Below the toolbar, a button labeled "Create" is visible. The main area is titled "All" and contains a list of tasks. Each task entry includes a checkbox, a task name, a description, and a small icon. The tasks listed are:

| Task Name         | Description                          | Action Icon |
|-------------------|--------------------------------------|-------------|
| Approval          | ... requests your approval           | Folder icon |
| CreateApplication |                                      | Folder icon |
| Inquiry           | ... sends the following inquiry: ... | Folder icon |
| Review            | ... requests your review             | Folder icon |
| To-do             | ... gives you a to-do: ...           | Folder icon |

- \_\_\_ f. Switch to the **Work on tasks** page. On this page you can access inbox and task queues, postpone tasks and resume them from the backlog, work on tasks, and complete them. You can also attach notes, create subtasks, set and change priorities and due dates, see and trigger escalations for tasks, and see the process status that is related to the current task.

The screenshot shows the MyBusinessSpace interface. At the top, there is a navigation bar with links: Home, Process Portal, Go to Spaces, Manage Spaces, and Actions. Below the navigation bar is a header titled "MyBusinessSpace". Underneath the header are several tabs: "Initiate Tasks and Processes", "Work on tasks" (which is highlighted with a red box), "Manage Processes", "Manage Tasks", and "Organize Work". The main content area is titled "Tasks" and contains a table with columns: Name, Priority, Status, Due date, Start date, Starter, and Pending. A dropdown menu above the table says "All - My work". An "Actions" button is located at the top right of the table. To the right of the table is a "Task Information" panel with the message "Selected task". Below the table, it says "No tasks were found."

- \_\_\_ g. Note the different widgets that are available on the page. The four widgets that are displayed are:
- **Tasks:** Displays the tasks that you own.
  - **Task Information:** Displays information about a task that you selected in the Tasks widget or the Process Information widget. This widget is the place where you do the work on the tasks that you own.
  - **Escalations:** Displays the escalations for a specific task. Depending on how the Escalations widget is configured, you can also assess the escalation status of the tasks that you own.
  - **Process Information:** Displays information about the process that is associated with the task.

- h. All the widgets on this page are empty. Depending on the other lab exercises you completed earlier, the screen capture might not match with your lab image. Feel free to drag the border of any widget to resize the page display.

The screenshot shows the 'MyBusinessSpace' application interface. At the top, there is a navigation bar with links: Home, Process Portal, Go to Spaces, Manage Spaces, Actions, and a dropdown menu. Below the navigation bar is a header titled 'MyBusinessSpace' with a logo. The main content area has several tabs: 'Initiate Tasks and Processes', 'Work on tasks' (which is highlighted with a red box), 'Manage Processes', 'Manage Tasks', and 'Organize Work'. The 'Work on tasks' tab is active, displaying a 'Tasks' section with a table header (Name, Priority, Status, Due date, Start date) and a message stating 'No tasks were found.' There is also an 'Escalations' section with a similar table header and a message stating 'No escalations were found.' To the right of these sections are two empty panels: 'Task Information' and 'Process Information', each with a placeholder message: 'Select the task and then select an...' and 'Select the process and then select an...' respectively.

- \_\_ i. Click the **Manage Processes** page. On the **Manage Processes** page, you can drill down from process definitions to processes, filter, and sort process instances. You can also see process details and associated tasks with all task details, and act on processes and tasks.

The screenshot shows the 'MyBusinessSpace' interface with the 'Manage Processes' tab selected. The 'Process Definitions' section lists two entries: 'AccountVerification' and 'OpenNewPosition', each with a 'View' button and a folder icon. The 'Processes' section shows a table with one row, 'All', indicating 'No processes were found.'

- \_\_ j. Note the different widgets that are available on the page. The five widgets that are displayed are:

- **Process Definitions:** Displays all process definitions that are available in the runtime system
- **Process Information:** Displays information that is associated with the process definition
- **Processes:** Displays a list of processes
- **Tasks:** All the available user actions for this widget are enabled
- **Task Information:** Displays information that is associated with the task

- k. Verify that the **AccountVerification** process is listed in the **Process Definition** widget. This process is available on the server.

The screenshot shows a 'Process Definitions' widget with a red border. It displays two entries:

- AccountVerification**:  
Account verification for %\InputCriterionParameters  
\Input/accountNumber% %\InputCriterionParameters  
\Input/companyName%  
Jan 8, 2010 2:14:25 AM
- OpenNewPosition**:  
Jun 8, 2011 12:57:07 PM

- l. Click the **Manage Tasks** page. On the **Manage Tasks** page, you can drill down from process and task definitions to filter, sort, and other tasks. You can also see task details, trigger related escalations, related process status, and team information.

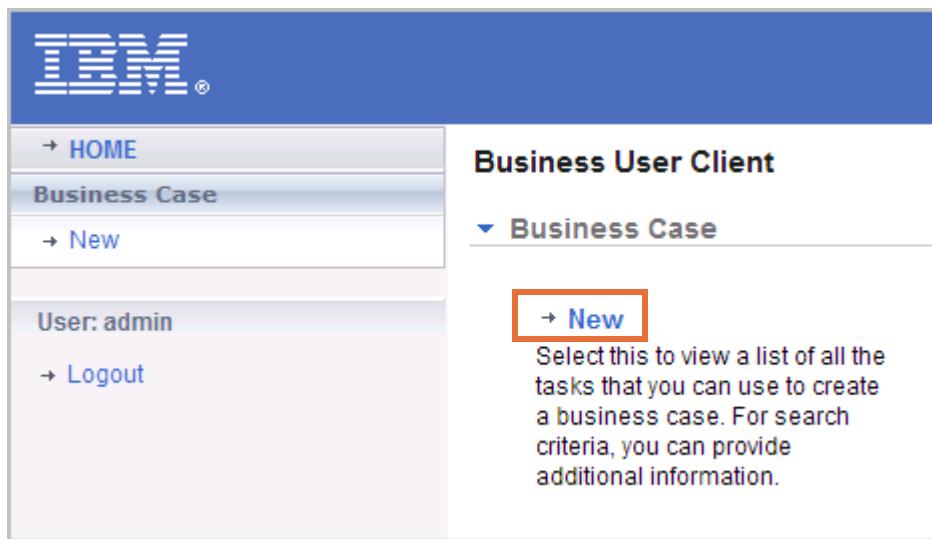
The screenshot shows the 'MyBusinessSpace' interface with a red box around the 'Manage Tasks' button in the top navigation bar. The page includes the following sections:

- Process Definitions**: Shows the same two processes as the previous screenshot.
- Tasks**: A section titled 'All - Unassigned and my work' with a 'Group by: Owner' dropdown. It displays a message: 'No tasks were found.'
- Task Information**: A section with the message: 'Select the task and then select an action.'
- Task Definitions**: A section with a warning message: '⚠ None of the configured filters are supported by this list. Select Edit Settings to reconfigure the list.'

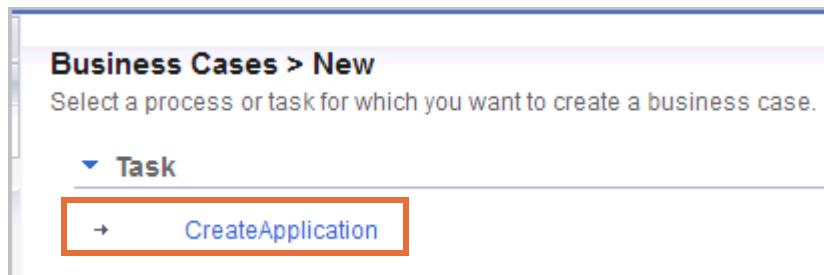
- \_\_\_ m. Note the different widgets that are available on the page. The seven widgets that are displayed are:
    - **Process Definitions:** Displays all process definitions that you can access.
    - **Task Definitions:** Displays a list of all task definitions that you can access.
    - **Tasks:** All the available user actions for this widget are enabled.
    - **Task Information:** This widget can display information about a task that you selected in the Tasks or the Process Information widgets. It can also use the Task Definitions widget to display the form for the task that you are creating.
    - **Process Information:** Displays information that is associated with the task or process.
    - **Team List:** Displays the potential owners for a task that you selected in the Tasks widget.
    - **Escalations:** Displays the escalations for a specific task.
  - \_\_\_ n. The Tasks widget on this page is empty. Later, you create a task instance and you submit it with this widget.
  - \_\_\_ o. Minimize the browser.
- \_\_\_ 3. Open the **CreateApplication** user interface to start an instance of the **AccountVerification** business process.
- \_\_\_ a. Open a new instance of the browser window.
  - \_\_\_ b. Type the following address in the location bar (note the case):  
`https://localhost:9443/AccountOpeningUI/Index.jsp`
  - \_\_\_ c. If prompted for login, type `admin` in the **Name field** and `web1sphere` in the **Password** field, and click **Login**.

The Business User Client page (`Workplace.jsp`) that is displayed is a JSP in the web project you generated. All pages in the project are customizable.

- \_\_\_ d. Under **Business Case**, click **New** to access the **CreateApplication** invocation task.



- \_\_\_ e. The only available task is **CreateApplication**. Click the link.



- \_\_\_ f. On the **CreateApplication** page, in the **Input Data** section, type **ACME** in the **companyName** field. You can leave the remaining fields blank. The Determine Application Eligibility Java snippet populates the remaining fields that are based on the `companyName`.

The screenshot shows the 'Business Cases > New > CreateApplication' page. The title is 'Business Cases > New > CreateApplication' and the sub-instruction is 'Enter the values for the input data and optionally provide additional information to create your task.' Below this, there's an 'Input Data' section with several fields: 'accountNumber', 'applicationDate', 'applicationDecision' (with a checkbox), 'comments', 'companyName' (containing 'ACME'), and 'contactFirstName'. The 'companyName' field is highlighted with a red border.

- \_\_\_ g. Click **Create** at the bottom of the page. The web browser returns to the **Business Cases** page.
- \_\_\_ h. Close the browser.
- \_\_\_ 4. Work with the available task in the **MyBusinessSpace** space.
- \_\_\_ a. Switch back to Business Space. You earlier minimized that browser instance.
- \_\_\_ b. Click the **Manage Processes** page.
- \_\_\_ c. Verify that the **CreateApplication** and **Final Application Review** tasks are listed in the Tasks widget. The **CreateApplication** task has a status of **In progress** and the **Final Application Review** task has a status of **Available**. It is OK if more tasks are listed in the widget. It might be necessary for you to scroll down to see it.

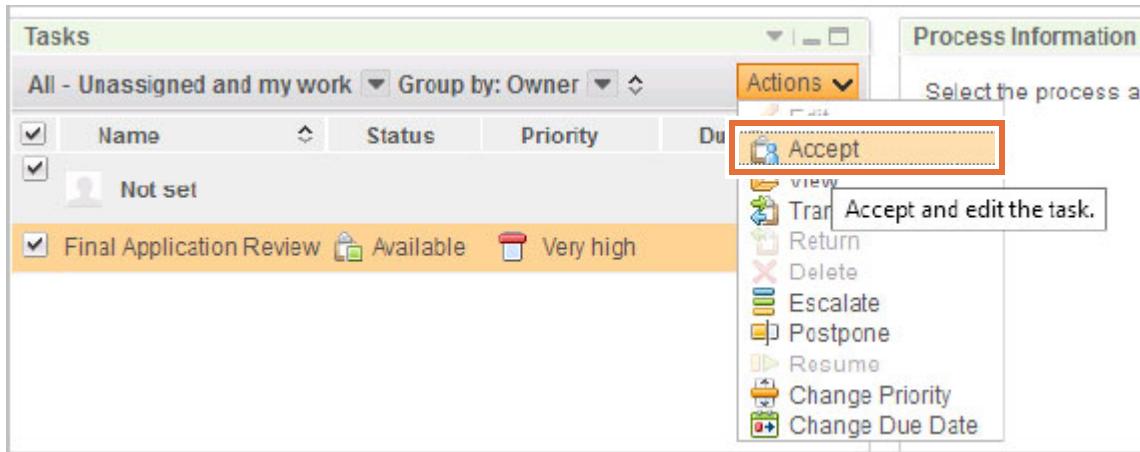
| Tasks           |                          |       |             |
|-----------------|--------------------------|-------|-------------|
| All - All items |                          |       |             |
|                 | Name                     | Owner | Status      |
|                 | CreateApplication        |       | In progress |
|                 | Final Application Review |       | Available   |

- \_\_\_ d. Switch to the **Manage Tasks** page.
- \_\_\_ e. Verify that the **Final Application Review** task is listed in the Tasks widget. Notice that the status is **Available**.

| Tasks                        |                          |                 |           |          |
|------------------------------|--------------------------|-----------------|-----------|----------|
| All - Unassigned and my work |                          | Group by: Owner | Actions   |          |
|                              | Name                     | Status          | Priority  | Due date |
|                              | Not set                  |                 |           |          |
|                              | Final Application Review | Available       | Very high |          |

- \_\_\_ f. Hover over the **Final Application Review** task to reveal a check box to the left of it.

- \_\_ g. Select the check box and click **Actions > Accept** from the widget menu options.



- \_\_ h. Scroll down to the **Task Information** widget. In the **Task Information** widget, click the **Output** form to bring it into focus, and clear the **applicationDecision** check box (to set it to `false`).

|                     |                                     |
|---------------------|-------------------------------------|
| accountNumber       | ACM002                              |
| applicationDate     | May 17, 2016                        |
| applicationDecision | <input checked="" type="checkbox"/> |
| comments            | None                                |
| companyName         | ACME                                |

- \_\_ i. Click **Submit** at the top of the Task Information widget.  
 \_\_ j. Verify that the Tasks widget is now empty because the task is complete.  
 \_\_ k. Switch to the Manage Processes page and scroll down to the **Tasks** widget.

- \_\_ I. Verify that the Final Application Review task is no longer listed. Also, confirm that the **CreateApplication** task has a **Completed** status now.

| Tasks |                   |       |           |          |     |
|-------|-------------------|-------|-----------|----------|-----|
|       | Name              | Owner | Status    | Due date | Sta |
|       | CreateApplication |       | Completed | May 17   |     |

- \_\_ 5. Verify the path that the application took through the remainder of the business process by examining the messages in the Server Logs view.
- \_\_ a. In IBM Integration Designer, switch to the **Server Logs** view.
- \_\_ b. The following messages in the server log confirm the path through the business process.

Generate Decline Special - begins

Generate Decline Special - Account for customer ACME was routed through special decline because the credit risk was MED

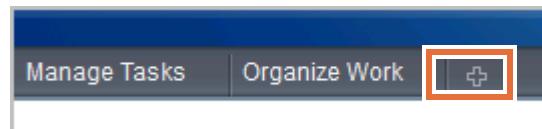
Generate Decline Special - ends

Record Declined Application - begins

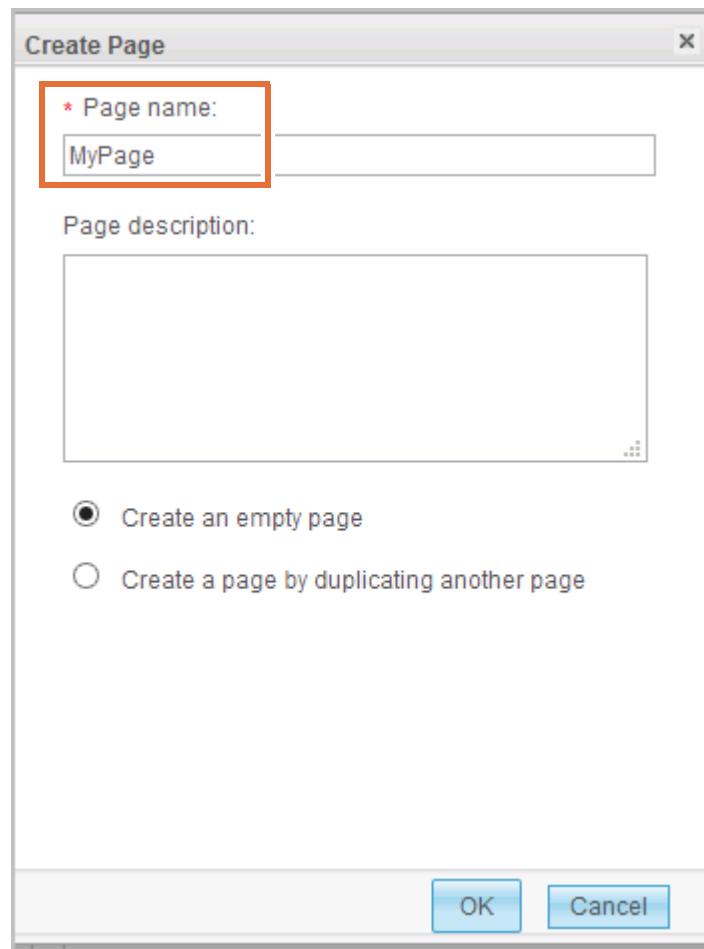
Record Declined Application - ends

|                                                                                                            |
|------------------------------------------------------------------------------------------------------------|
| WSVR0001I: Server server1 open for e-business                                                              |
| [Java] Determine Applicant Eligibility - begins                                                            |
| [Java] Determine Applicant Eligibility - ends                                                              |
| [Java] Generate Decline Special - begins                                                                   |
| [Java] Generate Decline Special - Account for customer ACME was routed through special decline because ... |
| [Java] Generate Decline Special - ends                                                                     |
| [Java] Record Declined Application - begins                                                                |
| [Java] Record Declined Application - ends                                                                  |

- \_\_ 6. If time permits, you can run the test again with **applicationDecision** set to `true`. For this test case, Generate Decline is not invoked so the Generate Decline Special messages are not listed in the Server Logs view.
- \_\_ 7. Create a page in **MyBusinessSpace**.
- \_\_ a. Switch back to the Business Space.
- \_\_ b. Create a page by clicking the + icon next to the **Organize Work** page.

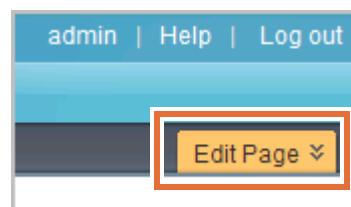


- \_\_\_ c. Enter **MyPage** in the **Page name** field, accept the remaining defaults, and click **OK**.

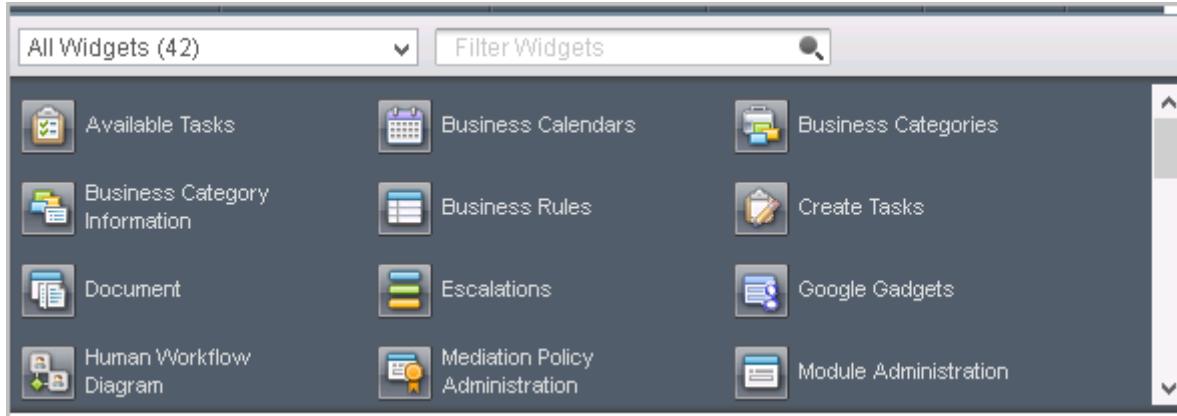


An empty page is displayed.

- \_\_\_ d. Click **Edit Page** at the upper-right corner of the page. After clicking, the text changes from **Edit Page** to **Finish Editing**.

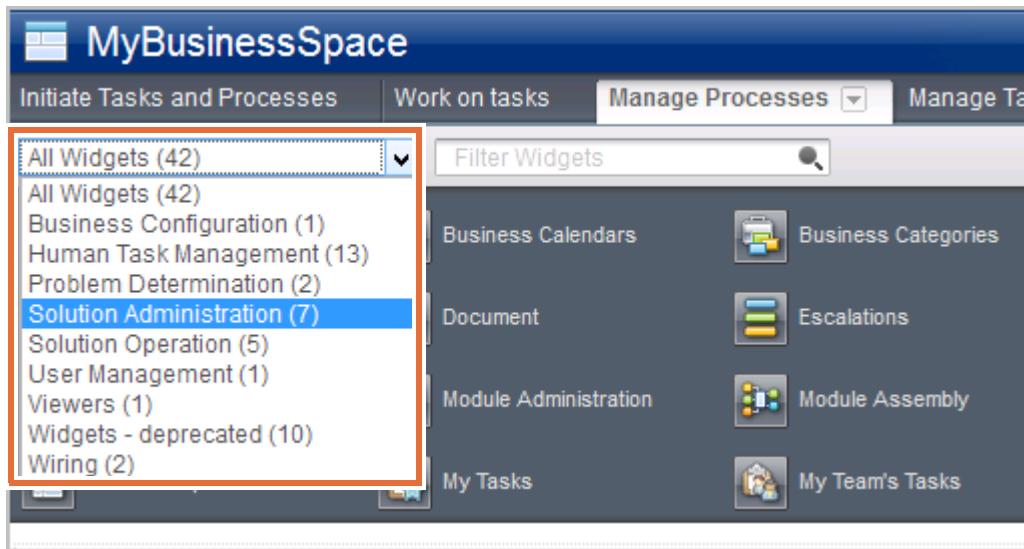


- \_\_ e. Confirm that a list of available widgets is displayed. Several widgets are listed. You can scroll down to view the entire list.

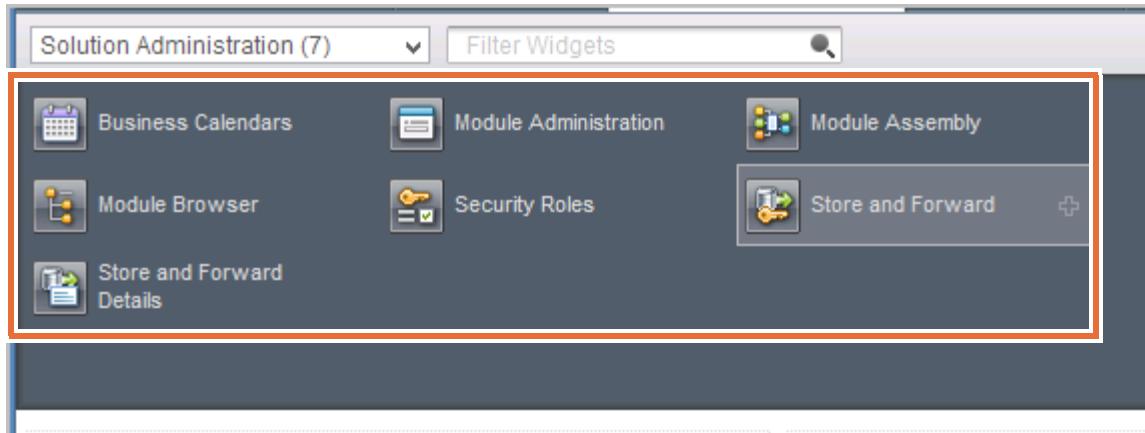


- \_\_ 8. Add widgets from the Solution Administration template.

- \_\_ a. Select **Solution Administration** from the widgets list.

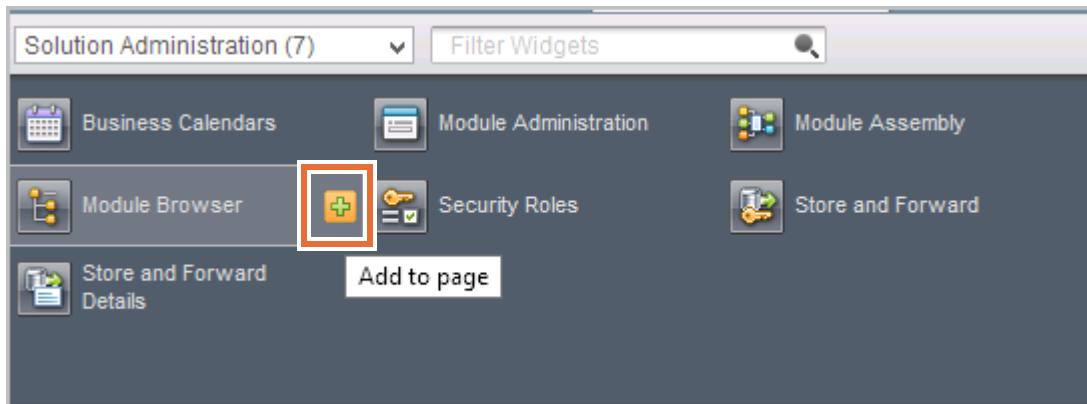


- \_\_ b. The list is filtered to display seven widgets.

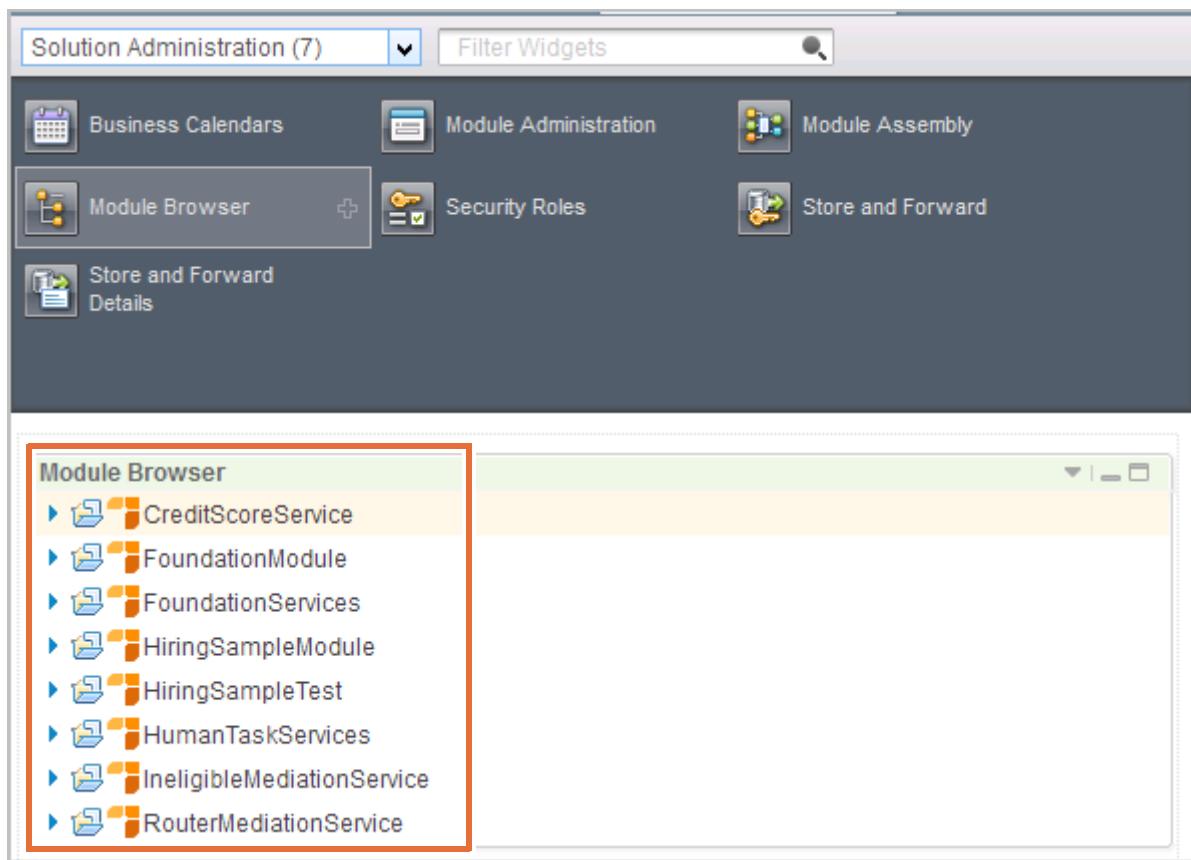


The Solution Administration widget provides access to widgets from which you can view and administer the modules and artifacts in your solution.

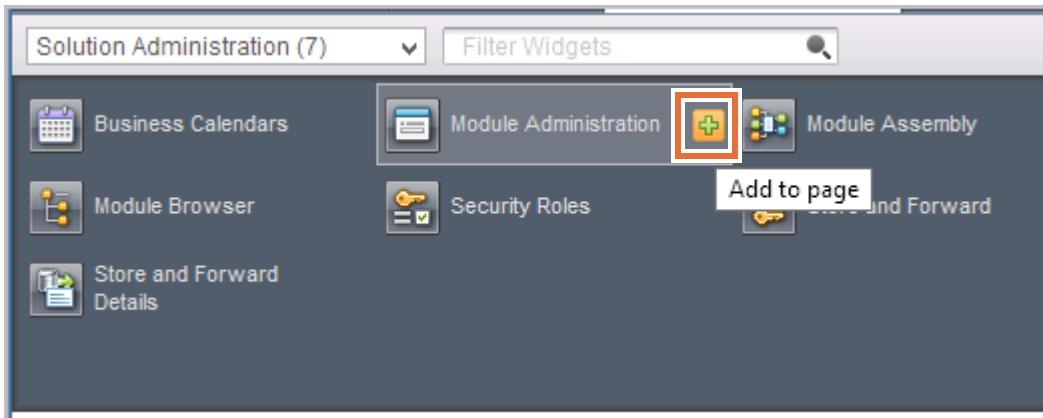
- c. Click the **Add to page** icon for the **Module Browser** widget.



- d. The newly added widget is displayed in **MyPage**. It lists all the Service Component Architecture (SCA) modules that are deployed on the server.



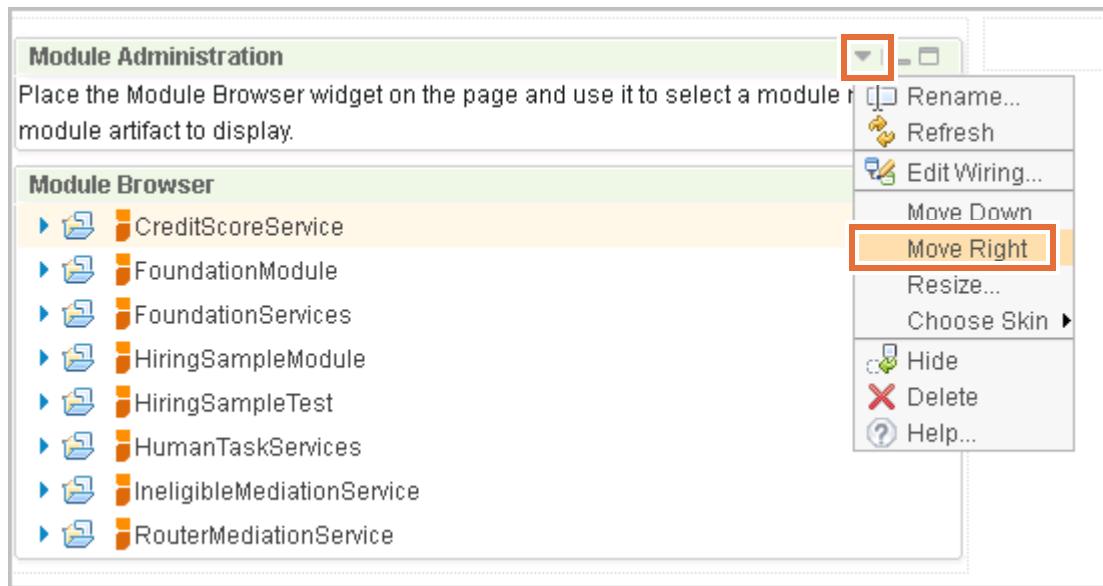
- \_\_ e. Click the Add to page icon for the **Module Administration** widget.



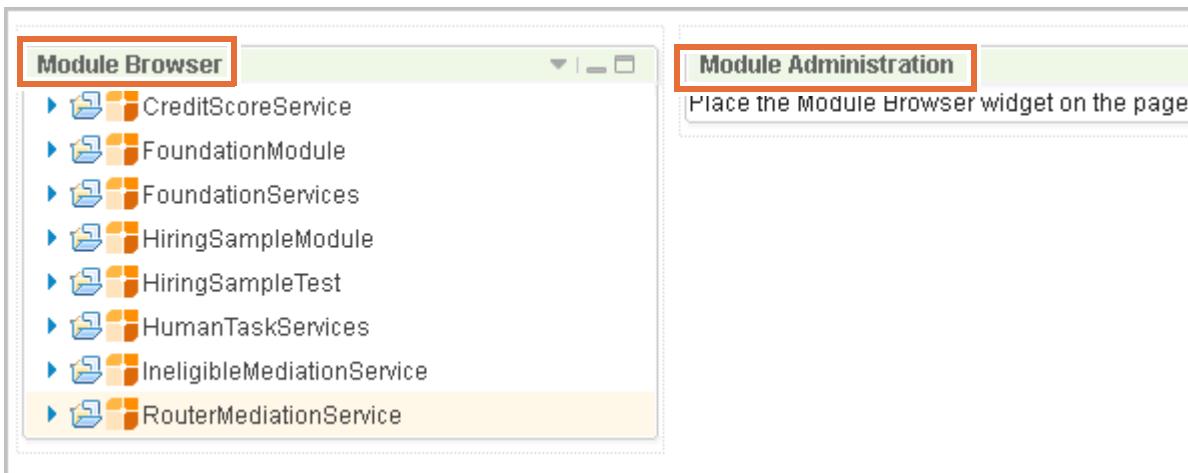
- \_\_ f. Verify that the **MyPage** page currently lists two widgets. The Module Administration widget currently displays no data.

A screenshot of the MyPage interface. At the top, there's a toolbar with tabs like 'Initiate Tasks and Processes', 'Work on tasks', 'Manage Processes', 'Manage Tasks', and 'Organizations'. Below the toolbar, a search bar says 'Filter Widgets' and a dropdown shows 'Solution Administration (7)'. A grid of icons represents different widgets: Business Calendars, Module Administration (with a plus sign icon), Module Assembly, Module Browser, Security Roles, Store and Forward, and Details. In the bottom half of the screen, there are two large, vertically stacked windows. The top window is titled 'Module Administration' and contains the instruction: 'Place the Module Browser widget on the page and use it to select a module name or module artifact to display.' The bottom window is titled 'Module Browser' and lists several module names: CreditScoreService, FoundationModule, FoundationServices, HiringSampleModule, HiringSampleTest, HumanTaskServices, IneligibleMediationService, and RouterMediationService. Both of these windows are highlighted with a red border.

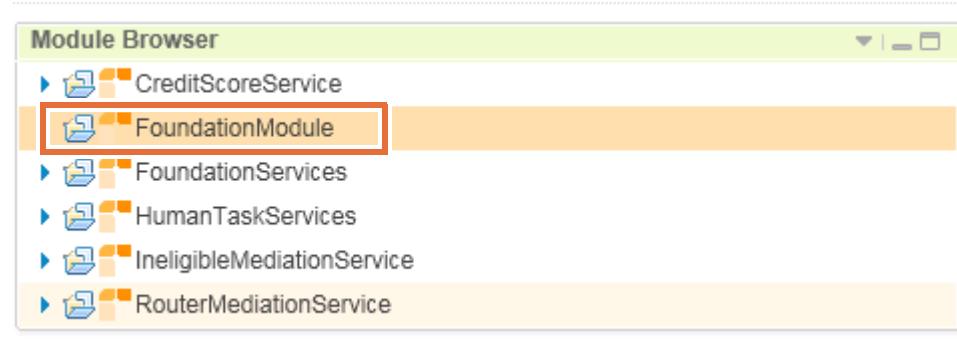
- g. You can rearrange the two widgets by clicking the down arrow icon in the upper-right corner of the Module Administration widget and clicking **Move Right**.



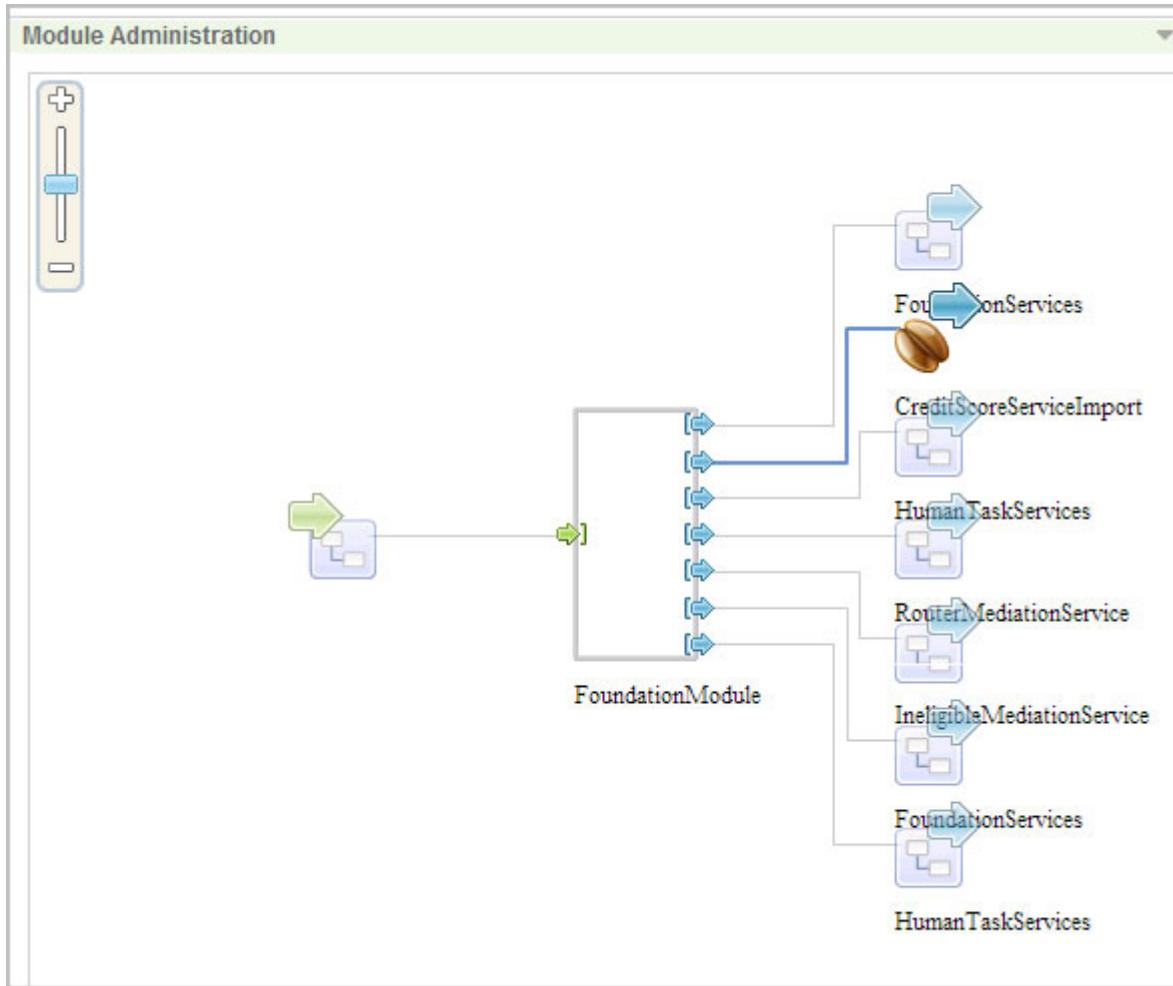
- h. The Module Administration widget is now displayed on the right of the page.



- i. Click **FoundationModule** in the Module Browser widget.



- \_\_ j. The details of FoundationModule are displayed in the Module Administration widget. You can hover over the objects in the widget to see more details.



- \_\_\_ k. Expand **FoundationModule > Processes** and click **AccountVerification** in the Module Browser widget. The process information is displayed in the Module Administration widget. Clicking **Explore Business Process Instances** starts Business Process Explorer in a separate browser. Do not click that link.

The screenshot shows the IBM Integration Designer interface. On the left, the **Module Browser** widget displays a tree view of artifacts and modules. The **FoundationModule** node is expanded, showing its **Processes** and **Module Policies** sub-nodes. The **AccountVerification** process is selected and highlighted in orange. On the right, the **Module Administration** widget provides details for the selected artifact. For the **AccountVerification** process, the details are:

| Module Administration |                                                       |
|-----------------------|-------------------------------------------------------|
| Name:                 | AccountVerification                                   |
| Type:                 | Business Process Template                             |
| Link:                 | <a href="#">Explore Business Process Instances...</a> |

- \_\_\_ l. Feel free to click other artifacts and modules in the Module Browser widget to view the details in the Module Administration widget.
- \_\_\_ m. Feel free to explore other available widgets by selecting different options in the widget list. When you are done exploring, click **Finish Editing**.
- \_\_\_ 9. Click the **Logout** link to log out of the Business Space.
- \_\_\_ 10. Close the browser window.
- \_\_\_ 11. Remove the applications from the server and (optionally) stop the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove Projects** from the menu.
- \_\_\_ b. Click **Remove All** and click **Finish**.
- \_\_\_ c. (Optional) Stop the server.
- \_\_\_ 12. Close IBM Integration Designer.

## End of exercise

## Exercise review and wrap-up

In this exercise, you created a business space and worked with several widgets to view displayed data.

# Exercise 14. Using component tests

## What this exercise is about

In this exercise, you use component tests to test applications. You also use cross-component tracing to inspect SCA components.

## What you should be able to do

After completing this exercise, you should be able to:

- Create a component test project with an operation-level test case
- Create a component test project with a scenario-based test case
- Run component test project test suites in the IBM Integration Designer integrated test environment

## Introduction

In IBM Integration Designer, the component test feature is the designated tool for setting up repeatable, predictable tests for both individual components and groups of components. You can use operation-level and scenario-based test cases. Component test projects contain test suites and are deployed to the IBM Process Server test environment as EAR files. When deployed, the integrated test environment or the Component Test Explorer can be used to run the component tests for performance and load testing.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Integration Designer test environment. Instructions for creating the lab environment are included in the exercise appendixes.

## Exercise instructions

### Part 1: Create a component test project with an operation-level test case

In IBM Integration Designer, you can test your modules by creating unit tests and component tests. In unit testing (the method that you used in the previous exercises), you choose the components and interfaces that contain the operations that you want to test. Then, you test the operations one at a time in the integrated test client. In component testing, you use the test suite editor and associated wizards to create test cases that comprise one or more operations. In this way, you can automate and simultaneously test the operations. Component test projects are SCA modules and are built and deployed like other SCA modules.

With the ability to test individual modules, operations, or components, the developer can prove that each of the various parts of an end-to-end solution works correctly. Then, when the entire solution is tested, it is easier to identify where problems originate.

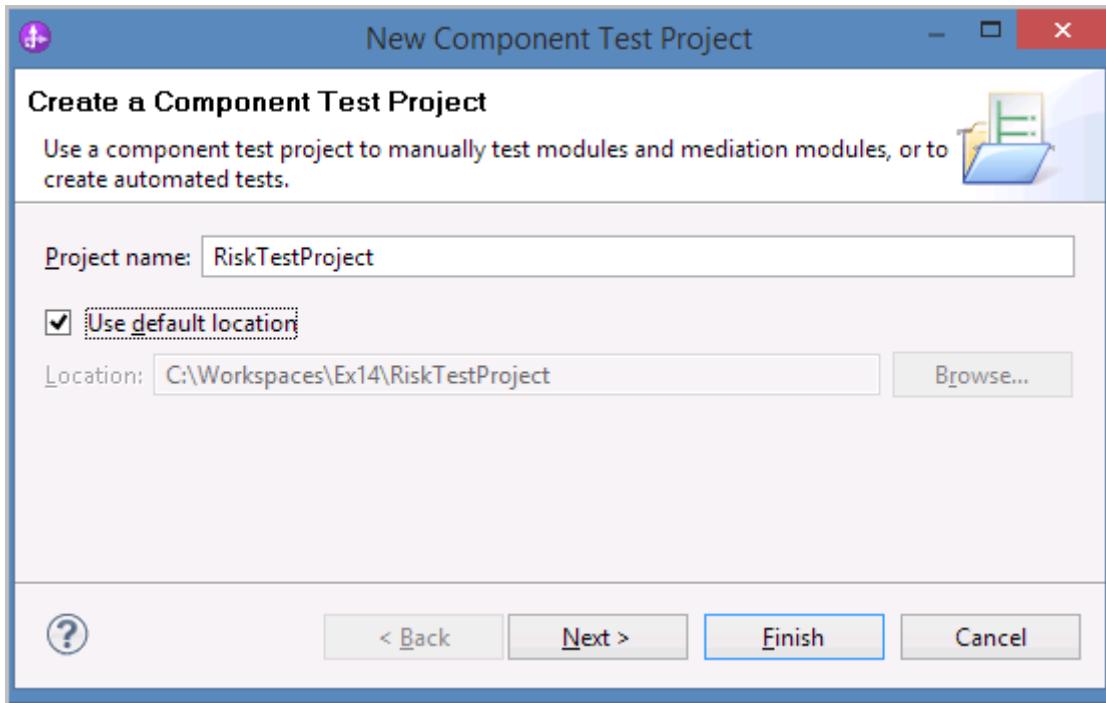
In this portion of the exercise, you create a component test project that contains a test suite, an operation-level test case, and variations of the test case. The test case that you create tests an individual operation: the `InputCriterion` operation of the `CreditRiskAssessment` interface. The `CreditRiskAssessment` service contains business rules that return predefined creditRisk values for each one of the four `companyName` test cases in your scenario. The creditRisk is `LOW` for IBM, `HIGH` for AbcCo and TestCo, and `MED` for ACME. The component test project uses IBM for the Default variation and AbcCo for the second variation. To save time, do not create variations for TestCo or ACME.

These tests require the right data inputs to match what the interfaces to the various modules expect. If you look again at the business objects and data maps, you can see the extensive data needed. In testing, it also includes expected values, allowing the tools to determine quickly and easily, whether the results are correct or not. To make testing easier and more automated, predefined data sets are made available here.

To create the component test project:

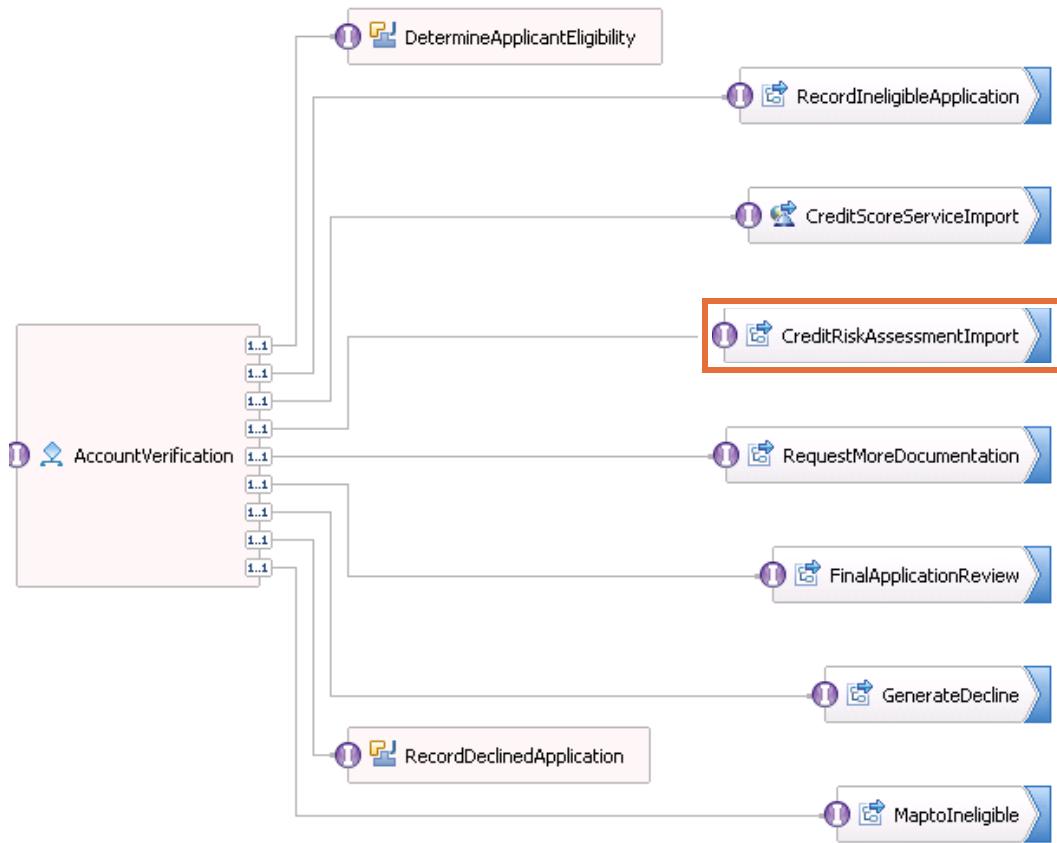
- \_\_\_ 1. Open the Exercise 14 workspace.
  - \_\_\_ a. On your desktop, open the **Exercise Shortcuts** folder.
  - \_\_\_ b. Double-click the **Exercise 14** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - \_\_\_ c. If you get a message that the server is already set to publish, then click **OK**.
  - \_\_\_ d. Close the **Getting Started** tab.
- \_\_\_ 2. Create a test project that is named: `RiskTestProject`
  - \_\_\_ a. Click **File > New > Component Test Project** from the menu options.

- \_\_ b. In the **Project name** field, type: RiskTestProject



- \_\_ c. Click **Finish**.

- \_\_\_ 3. Create an operation-level test suite that is named: RiskTestSuite  
The test suite tests the InputCriterion operation of the CreditRiskAssessment interface.



**Interface**

Configuration

|               |                                                                    |                                                                                           |
|---------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Name          | CreditRiskAssessment                                               | <a href="#">Refactor name</a>                                                             |
| Namespace     | http://FoundationLibrary/creditriskassessment/CreditRiskAssessment | <a href="#">Refactor namespace</a>                                                        |
| Binding Style | document literal wrapped                                           | <a href="#">Change binding style to document literal non-wrapped</a> <a href="#">More</a> |

**Operations** |

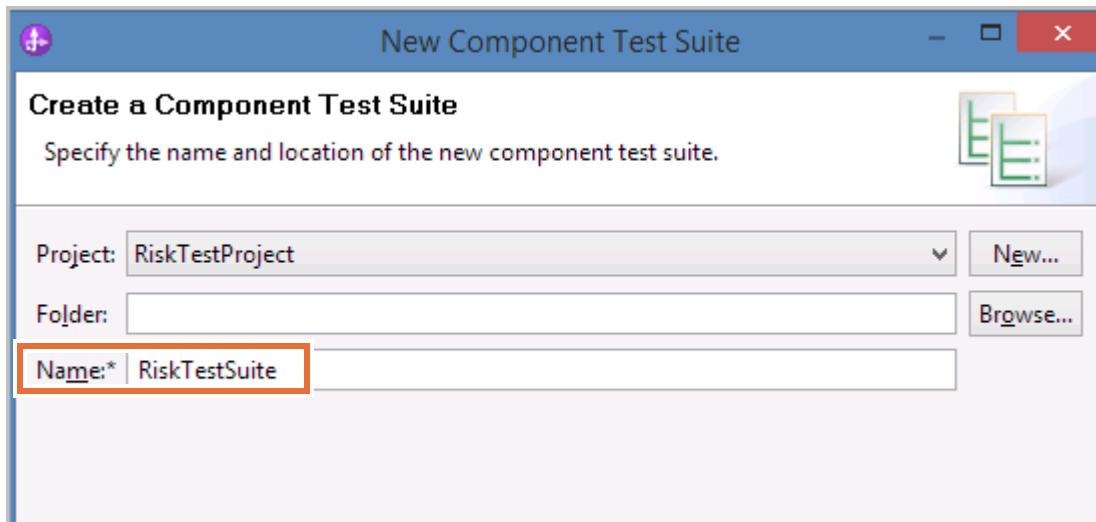
Operations and their parameters

| Name           | Type   |
|----------------|--------|
| InputCriterion |        |
| Inputs         | Input  |
| Outputs        | Output |

- \_\_ a. In the Business Integration view, expand **RiskTestProject**.
  - \_\_ b. Right-click **Test Suites** and click **New > Component Test Suite** from the menu.

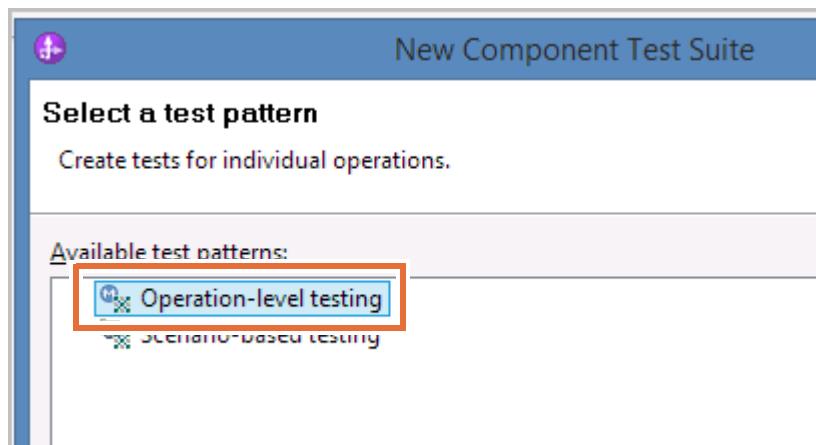
\_\_ c. At the “Create a Component Test Suite” panel, take the following actions:

- Leave the **Folder** field empty.
- In the **Name** field, type: RiskTestSuite



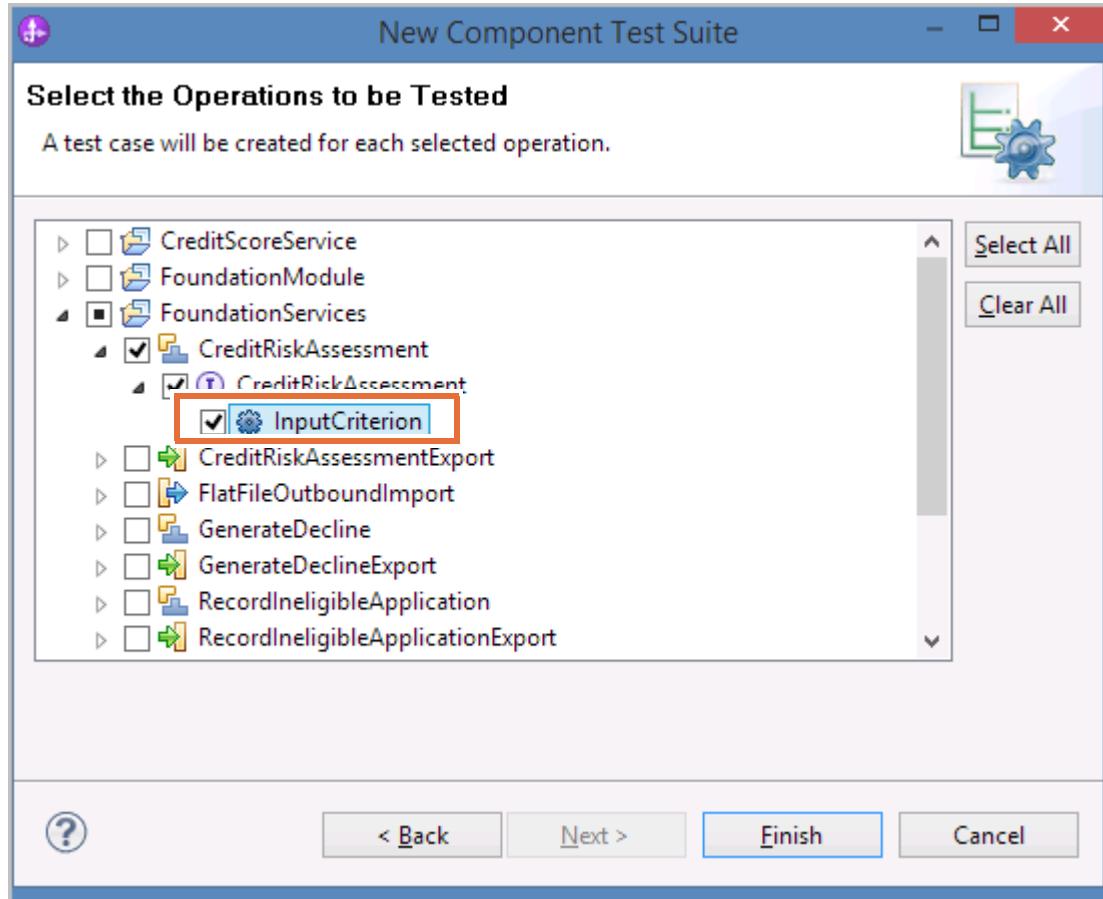
\_\_ d. Click **Next**.

\_\_ e. At the “Select a test pattern” panel, select **Operation-level testing**.



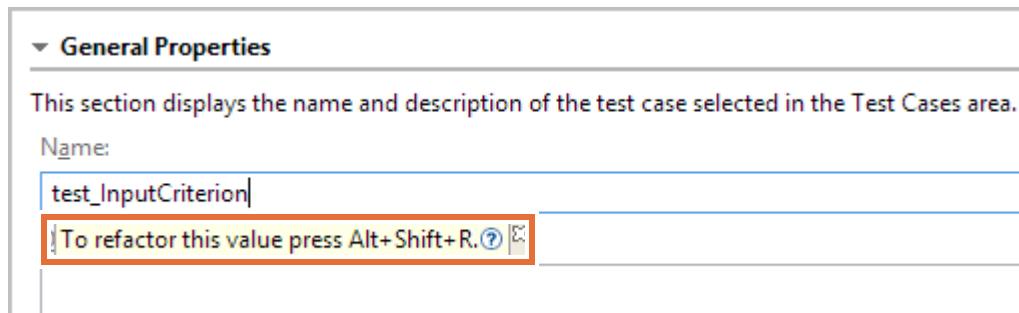
\_\_ f. Click **Next**.

- \_\_ g. At the “Select the Operations to be Tested” panel, take the following actions:
- Expand **FoundationServices > CreditRiskAssessment > CreditRiskAssessment**.
  - Select the **InputCriterion** operation in the **CreditRiskAssessment** interface.

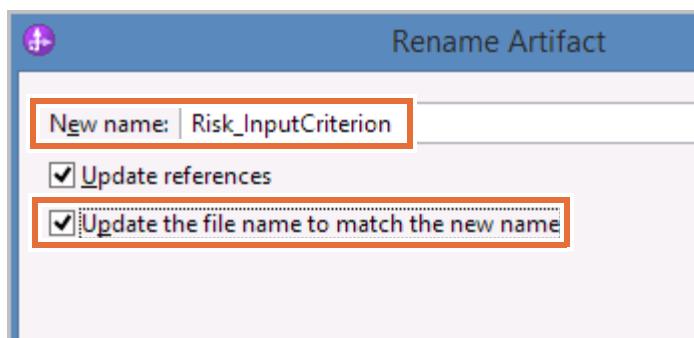


- \_\_ h. Click **Finish**.
- \_\_ 4. Refactor the name of the test case from `test_InputCriterion` to: `Risk_InputCriterion`
- \_\_ a. In the **Test Cases** section of the test suite editor, click the `test_InputCriterion` link.

- \_\_\_ b. In the **General Properties** section, place the cursor in the **Name** field and press Alt+Shift+R to refactor the name.



- \_\_\_ c. In the Rename Artifact dialog box, take the following actions:
- In the **New name** field, type: Risk\_InputCriterion
  - Select **Update the file name to match the new name**.



- \_\_\_ d. Click **Refactor**.
- \_\_\_ 5. Use the values in C:\labfiles\Support Files\Ex14\EX14\_Default.xml to enter the input data for the **Default** test variation.
- \_\_\_ a. At the bottom of the window, switch to the **Default** tab in the **Test Data Table** view.
  - \_\_\_ b. Right-click **Input** and click **Import from File** from the menu.
  - \_\_\_ c. Browse to C:\labfiles\Support Files\Ex14.
  - \_\_\_ d. Select **EX14\_Default.xml** and click **Open** to populate the XSet column with the required test data.

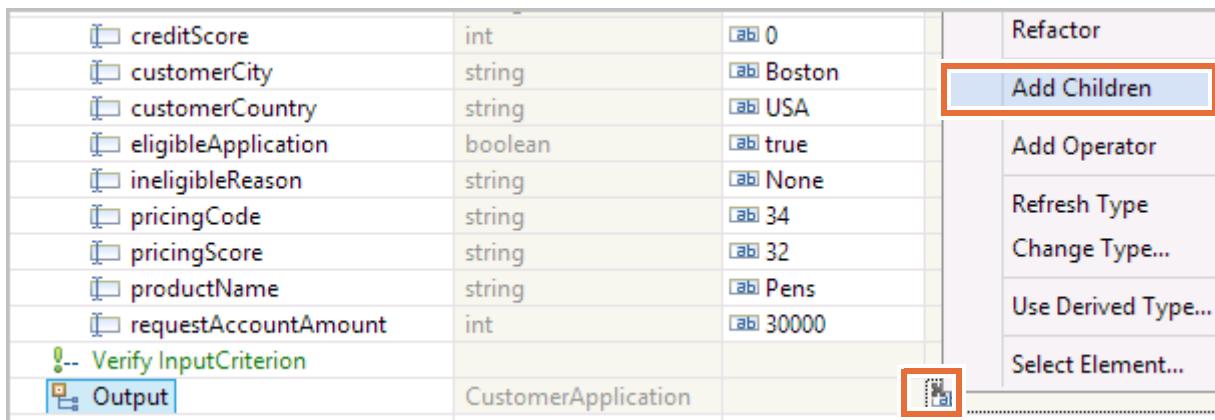
- \_\_\_ e. Alternatively, manually enter the following values on the **Default** tab (right-click the **Test Data Table** view and click **Maximize** to assist you with data entry).

- accountNumber: IBM007
- applicationDate: 06/10/2016
- applicationDecision: true
- comments: None
- companyName: IBM
- contactFirstName: Landon
- contactLastName: Donovan
- contactPhoneNumber: 547-555-3172
- creditRating: A++
- creditReportNeeded: true
- creditRisk: LOW
- creditScore: 0
- customerCity: Boston
- customerCountry: USA
- eligibleApplication: true
- ineligibleReason: None
- pricingCode: 34
- pricingScore: 32
- productName: Pens
- requestAccountAmount: 30000

The screenshot shows the 'Default' tab of a test data table for the 'RiskTestSuite : Risk\_InputCriterion' scenario. The table has columns for Name, Type, and Value. The 'Name' column lists parameters like accountNumber, applicationDate, applicationDecision, etc. The 'Type' column indicates the data type for each parameter. The 'Value' column shows the specific value assigned for each parameter.

| Name                                              | Type                | Value        |
|---------------------------------------------------|---------------------|--------------|
| Invoke CreditRiskAssessment:InputCriterion(Input) |                     |              |
| Input                                             | CustomerApplication |              |
| accountNumber                                     | string              | IBM007       |
| applicationDate                                   | string              | 06/10/2016   |
| applicationDecision                               | boolean             | true         |
| comments                                          | string              | None         |
| companyName                                       | string              | IBM          |
| contactFirstName                                  | string              | Landon       |
| contactLastName                                   | string              | Donovan      |
| contactPhoneNumber                                | string              | 547-555-3172 |
| creditRating                                      | string              | A++          |
| creditReportNeeded                                | boolean             | true         |
| creditRisk                                        | string              | LOW          |
| creditScore                                       | int                 | 0            |
| customerCity                                      | string              | Boston       |
| customerCountry                                   | string              | USA          |
| eligibleApplication                               | boolean             | true         |
| ineligibleReason                                  | string              | None         |
| pricingCode                                       | string              | 34           |
| pricingScore                                      | string              | 32           |
| productName                                       | string              | Pens         |
| requestAccountAmount                              | int                 | 30000        |

6. Use the values in C:\labfiles\Support Files\Ex14\EX14\_Default\_Out.xml to enter the output data that is expected from the **CreditRiskAssessment** service.
- In the **Test Data Table** view, on the **Default** tab, scroll to the **Output** section.
  - Right-click the **Expected** column value and click **Add Children** from the menu.



|                          |                     |        |                                                                                     |
|--------------------------|---------------------|--------|-------------------------------------------------------------------------------------|
| creditScore              | int                 | 0      | Refactor                                                                            |
| customerCity             | string              | Boston |                                                                                     |
| customerCountry          | string              | USA    |                                                                                     |
| eligibleApplication      | boolean             | true   |                                                                                     |
| ineligibleReason         | string              | None   |                                                                                     |
| pricingCode              | string              | 34     |                                                                                     |
| pricingScore             | string              | 32     |                                                                                     |
| productName              | string              | Pens   |                                                                                     |
| requestAccountAmount     | int                 | 30000  |                                                                                     |
| -- Verify InputCriterion |                     |        |                                                                                     |
| <b>Output</b>            | CustomerApplication |        |  |

- Right-click **Output** and click **Import from File**.
- Browse to C:\labfiles\Support Files\Ex14.
- Select **EX14\_Default\_Out.xml** and click **Open** to populate the **Expected** column with the output test data.

\_\_ f. Alternatively, manually enter the following values.

- accountNumber: IBM007
- applicationDate: [leave blank]
- applicationDecision: true
- comments: None
- companyName: IBM
- contactFirstName: Landon
- contactLastName: Donovan
- contactPhoneNumber: 547-555-3172
- creditRating: A++
- creditReportNeeded: true
- creditRisk: LOW
- creditScore: 0
- customerCity: Boston
- customerCountry: USA
- eligibleApplication: true
- ineligibleReason: [leave blank]
- pricingCode: 34
- pricingScore: 32
- productName: Pens
- requestAccountAmount: 30000

| Name                 | Type                | Expected         |
|----------------------|---------------------|------------------|
| Output               | CustomerApplication | (=)              |
| accountNumber        | string              | (=) IBM007       |
| applicationDate      | string              | (=)              |
| applicationDecision  | boolean             | (=) true         |
| comments             | string              | (=) None         |
| companyName          | string              | (=) IBM          |
| contactFirstName     | string              | (=) Landon       |
| contactLastName      | string              | (=) Donovan      |
| contactPhoneNumber   | string              | (=) 547-555-3172 |
| creditRating         | string              | (=) A++          |
| creditReportNeeded   | boolean             | (=) true         |
| creditRisk           | string              | (=) LOW          |
| creditScore          | int                 | (=) 0            |
| customerCity         | string              | (=) Boston       |
| customerCountry      | string              | (=) USA          |
| eligibleApplication  | boolean             | (=) true         |
| ineligibleReason     | string              | (=)              |
| pricingCode          | string              | (=) 34           |
| pricingScore         | string              | (=) 32           |
| productName          | string              | (=) Pens         |
| requestAccountAmount | int                 | (=) 30000        |

- g. Whether you imported the test data or not, you must manually set the **applicationDate** and **ineligibleReason** values in **Output** by taking the following actions:
- Right-click the empty **Expected** value for **applicationDate** and click **Set To > Not Equals Null** from the menu. It indicates that the result must contain some value, which can be anything.
  - Right-click the empty **Expected** value for **ineligibleReason** and click **Set To > Do Not Care** from the menu. It indicates that the result can be anything, including nothing.

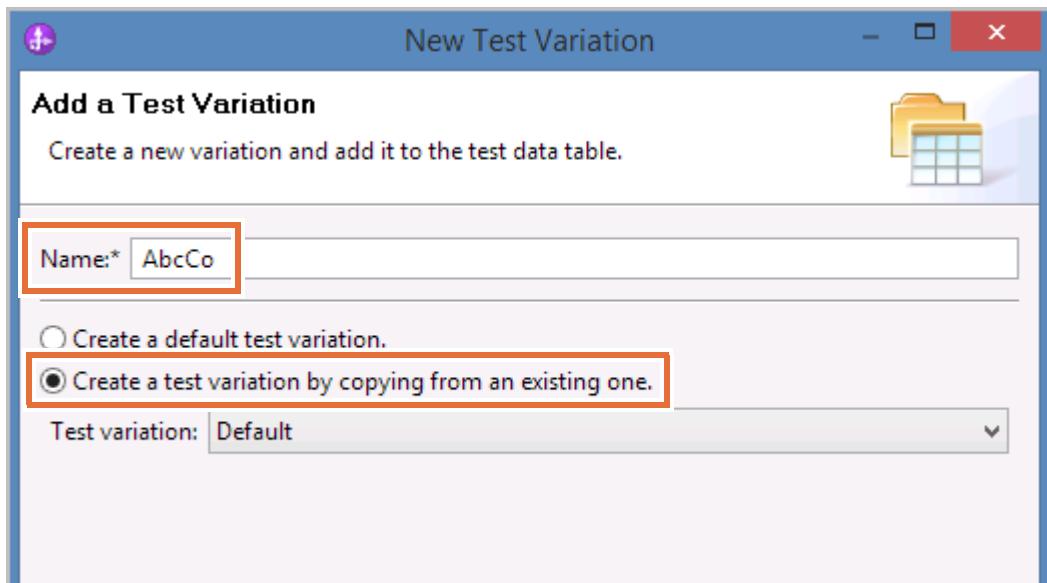
| Name                 | Type                | →X S... | X==Expected  |
|----------------------|---------------------|---------|--------------|
| Output               | CustomerApplication | ab (==) |              |
| accountNumber        | string              | ab (==) | IBM007       |
| applicationDate      | string              | ab (==) | null         |
| applicationDecision  | boolean             | ab (==) | true         |
| comments             | string              | t-->    | None         |
| companyName          | string              | ab (==) | IBM          |
| contactFirstName     | string              | ab (==) | Landon       |
| contactLastName      | string              | ab (==) | Donovan      |
| contactPhoneNumber   | string              | ab (==) | 547-555-3172 |
| creditRating         | string              | ab (==) | A++          |
| creditReportNeeded   | boolean             | ab (==) | true         |
| creditRisk           | string              | ab (==) | LOW          |
| creditScore          | int                 | ab (==) | 0            |
| customerCity         | string              | ab (==) | Boston       |
| customerCountry      | string              | ab (==) | USA          |
| elinibileApplication | boolean             | ab (==) | true         |
| ineligibleReason     | string              | ab      |              |
| pricingCode          | string              | t==>    | 34           |
| pricingScore         | string              | ab (==) | 32           |
| productName          | string              | ab (==) | Pens         |
| requestAccountAmount | int                 | ab (==) | 30000        |

- 7. Save your changes.
- 8. Use the values in `C:\labfiles\Support Files\Ex14\EX14_AbcCo.xml` and `EX14_AbcCo_Out.xml` to create another variation of the input and output data. Because the Default variation uses IBM, this test uses AbcCo for the `companyName`. IBM returns a `creditRisk` value of `LOW`. AbcCo returns a `creditRisk` value of `HIGH`.
- a. In the **Test Data Table** view, click the **Add Variation** icon (look far right to the **Test Data Table** view if your window is maximized).



- \_\_ b. In the **Add a Test Variation** pane, take the following actions:

- In the **Name** field, type: AbcCo
- Select **Create a test variation by copying from an existing one.**
- Verify that the **Test variation** field is set to **Default**.



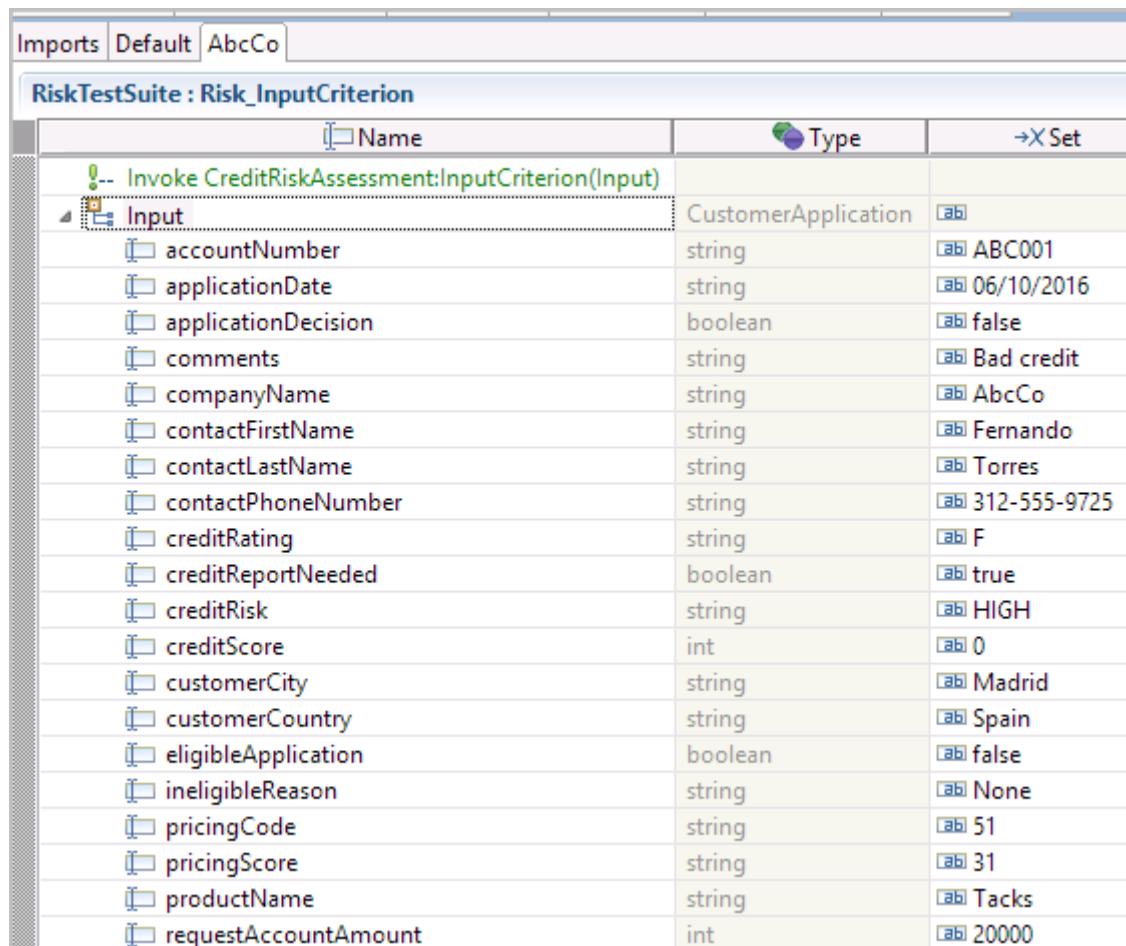
- \_\_ c. Click **Finish** to create a set of data based on the Default test variation.  
\_\_ d. You now see a new tab in the Test Data Table view that is named `AbcCo`.



- \_\_ e. Right-click **Input** and click **Import from File**.  
\_\_ f. Browse to `C:\labfiles\Support Files\Ex14`.  
\_\_ g. Select `EX14_AbcCo.xml`, and click **Open** to populate the XSet column with the required test data.

- \_\_\_ h. If you did not import the test data, manually change the following values:

- accountNumber: ABC001
- applicationDate: 06/10/2016
- applicationDecision: false
- comments: Bad credit
- companyName: AbcCo
- contactFirstName: Fernando
- contactLastName: Torres
- contactPhoneNumber: 312-555-9725
- creditRating: F
- creditRisk: HIGH
- customerCity: Madrid
- customerCountry: Spain
- eligibleApplication: false
- pricingCode: 51
- pricingScore: 31
- productName: Tacks
- requestAccountAmount: 20000

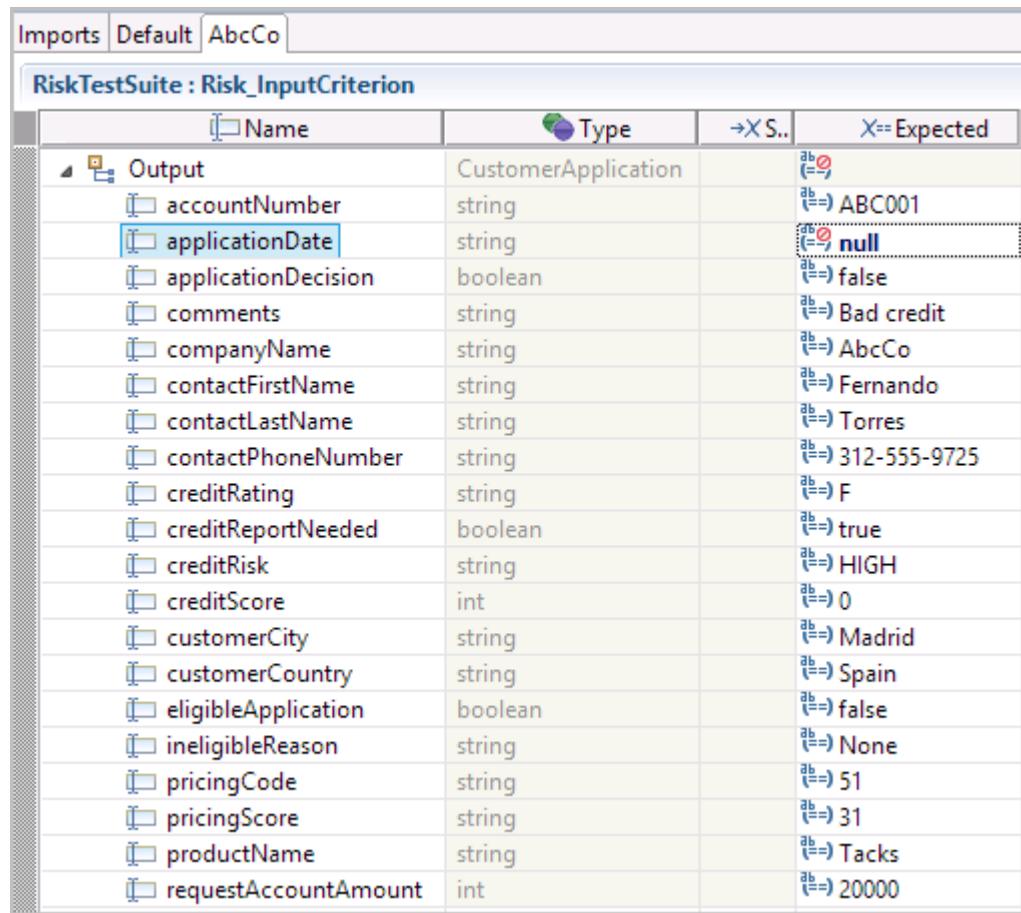


The screenshot shows the Rational Test Studio interface with the following details:

- Imports:** Default
- Current Project:** AbcCo
- Table Title:** RiskTestSuite : Risk\_InputCriterion
- Table Columns:**
  - Name
  - Type
  - Set
- Data Rows:**
  - Input** (CustomerApplication) - Type: CustomerApplication, Value: ABC001
  - accountNumber - Type: string, Value: ABC001
  - applicationDate - Type: string, Value: 06/10/2016
  - applicationDecision - Type: boolean, Value: false
  - comments - Type: string, Value: Bad credit
  - companyName - Type: string, Value: AbcCo
  - contactFirstName - Type: string, Value: Fernando
  - contactLastName - Type: string, Value: Torres
  - contactPhoneNumber - Type: string, Value: 312-555-9725
  - creditRating - Type: string, Value: F
  - creditReportNeeded - Type: boolean, Value: true
  - creditRisk - Type: string, Value: HIGH
  - creditScore - Type: int, Value: 0
  - customerCity - Type: string, Value: Madrid
  - customerCountry - Type: string, Value: Spain
  - eligibleApplication - Type: boolean, Value: false
  - ineligibleReason - Type: string, Value: None
  - pricingCode - Type: string, Value: 51
  - pricingScore - Type: string, Value: 31
  - productName - Type: string, Value: Tacks
  - requestAccountAmount - Type: int, Value: 20000

- \_\_\_ i. Right-click **Output** and click **Import from File**.  
 \_\_\_ j. Browse to C:\labfiles\Support Files\Ex14.

- \_\_ k. Select **EX14\_AbcCo\_Out.xml** and click **Open** to populate the **Expected** column with the required test data.
- \_\_ l. Right-click the **Expected** value for **applicationDate** and click **Set to > Not equals Null**.
- \_\_ m. If you did not import the test data, manually change the following values:
- accountNumber: ABC001
  - applicationDecision: false
  - comments: Bad credit
  - companyName: AbcCo
  - contactFirstName: Fernando
  - contactLastName: Torres
  - contactPhoneNumber: 312-555-9725
  - creditRating: F
  - creditRisk: HIGH
  - customerCity: Madrid
  - customerCountry: Spain
  - eligibleApplication: false
  - ineligibleReason: None
  - pricingCode: 51
  - pricingScore: 31
  - productName: Tacks
  - requestAccountAmount: 20000



The screenshot shows the IBM BPM RiskTestSuite interface with the 'RiskTestSuite : Risk\_InputCriterion' table. The table has columns for Name, Type, and Expected. The 'Name' column lists various application parameters, and the 'Type' column indicates their data type (e.g., string, boolean, int). The 'Expected' column contains the specific values assigned to each parameter. The 'applicationDate' row is highlighted with a red border, indicating it is the current field being edited.

| Name                 | Type    | Expected     |
|----------------------|---------|--------------|
| accountNumber        | string  | ABC001       |
| applicationDate      | string  | null         |
| applicationDecision  | boolean | false        |
| comments             | string  | Bad credit   |
| companyName          | string  | AbcCo        |
| contactFirstName     | string  | Fernando     |
| contactLastName      | string  | Torres       |
| contactPhoneNumber   | string  | 312-555-9725 |
| creditRating         | string  | F            |
| creditReportNeeded   | boolean | true         |
| creditRisk           | string  | HIGH         |
| creditScore          | int     | 0            |
| customerCity         | string  | Madrid       |
| customerCountry      | string  | Spain        |
| eligibleApplication  | boolean | false        |
| ineligibleReason     | string  | None         |
| pricingCode          | string  | 51           |
| pricingScore         | string  | 31           |
| productName          | string  | Tacks        |
| requestAccountAmount | int     | 20000        |

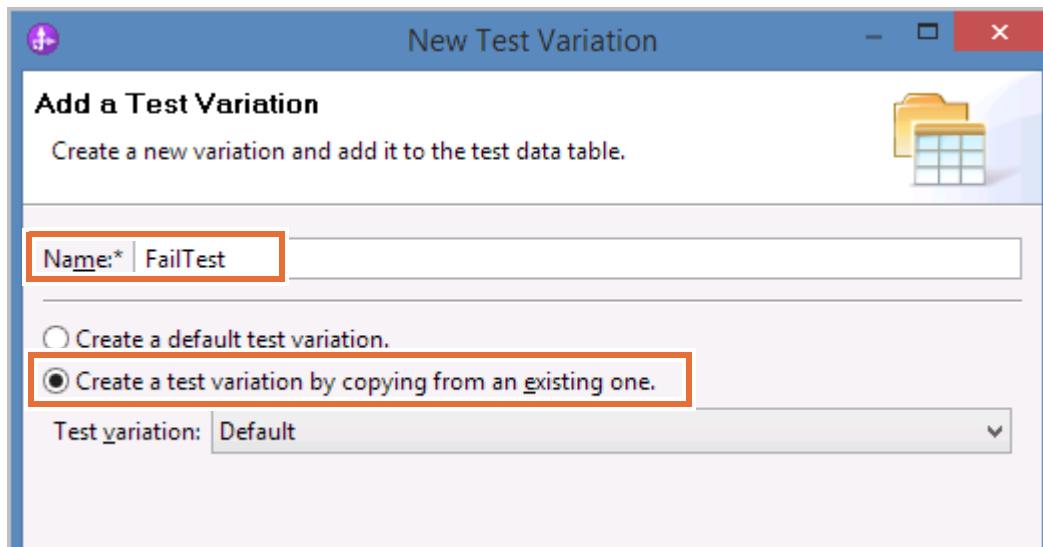
**Note**

Though not shown, you can also create variations for the remaining company names: TestCo and ACME. The expected input and output values can be found in C:\labfiles\Support Files\Ex8\DetermineApplicantEligibilityImpl\_InputCriterion.txt.

- \_\_\_ 9. Save your changes.
- \_\_\_ 10. Create a FailTest variation by using an improper companyName value: BeanCo  
This variation demonstrates test failure.
- \_\_\_ a. In the **Test Data Table** view, click the **Add Variation** icon (look far right to the **Test Data Table** view if your window is maximized).

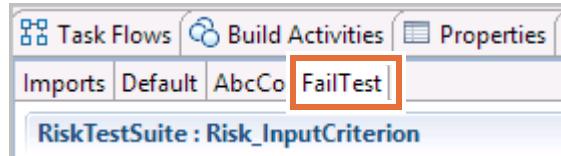


- \_\_\_ b. In the “Add a Test Variation” pane, enter the following information:
  - In **Name** field, type: FailTest
  - Select **Create a test variation by copying from an existing one**.
  - Verify that the **Test variation** field is set to **Default**.



- \_\_\_ c. Click **Finish**.

- \_\_ d. You now see a new tab in the **Test Data Table** that is named `FailTest`.



- \_\_ e. Change the `companyName` value in `Input` to `BeanCo` (a value that the `CreditRiskAssessment` rule set does not recognize).

| Name                                                 | Type                | Set               |
|------------------------------------------------------|---------------------|-------------------|
| -- Invoke CreditRiskAssessment:InputCriterion(Input) |                     |                   |
| Input                                                | CustomerApplication | [ab]              |
| accountNumber                                        | string              | [ab] IBM007       |
| applicationDate                                      | string              | [ab] 06/10/2016   |
| applicationDecision                                  | boolean             | [ab] true         |
| comments                                             | string              | [ab] None         |
| companyName                                          | string              | [ab] BeanCo       |
| contactFirstName                                     | string              | [ab] Landon       |
| contactLastName                                      | string              | [ab] Donovan      |
| contactPhoneNumber                                   | string              | [ab] 547-555-3172 |
| creditRating                                         | string              | [ab] A++          |

- \_\_ 11. Save your changes. Each one of your validation cases is listed in the **Detailed Properties** section of the test suite editor.

▼ **Detailed Properties**

This section displays the links to the test configuration and test variations of the test case that is selected in the Test Cases area. [More...](#)

Test configuration: [RiskTestSuite](#)

Test variations

- [Default](#)
- [AbcCo](#)
- [FailTest](#)

- \_\_ 12. Close the **RiskTestSuite** tab.

## Running a test suite with operation-level test case

In this portion of the exercise, you run **RiskTestSuite** in the IBM Integration Designer test environment. The Default and AbcCo variations are expected to pass. As expected, the FailTest variation fails.

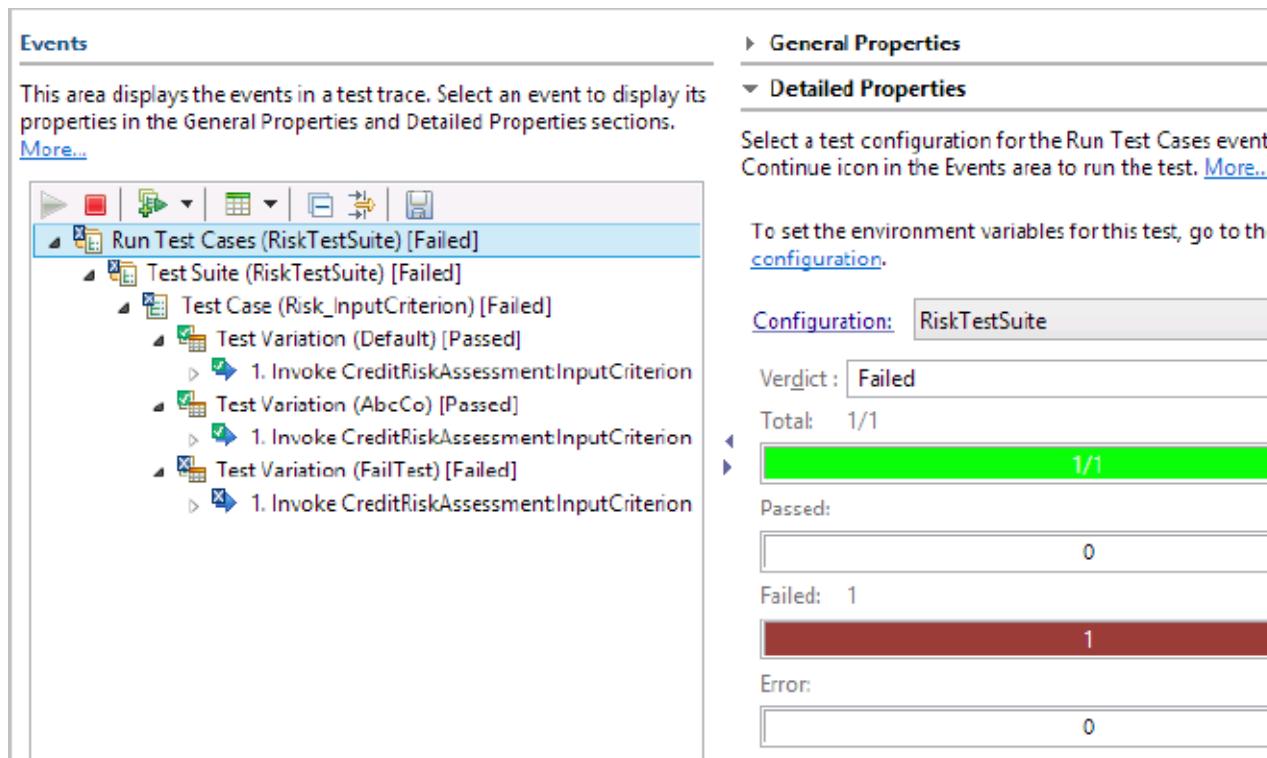
To run the test suite:

- \_\_\_ 1. Run **RiskTestSuite** in the IBM Integration Designer test environment.
  - \_\_\_ a. (If necessary) In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu. Wait for the startup process to complete before continuing.

The server is started when the message `Server server1 open for e-business` is displayed in the **Server Logs** view. The server status also changes to **Started** in the **Servers** view.

- \_\_\_ b. In the Business Integration view, expand **RiskTestProject**.
- \_\_\_ c. Right-click **RiskTestSuite** and click **Run Test Cases** from the menu.
- \_\_\_ d. Click the **Continue** icon in the Event window action bar.
- \_\_\_ e. At the “Select a Deployment Location” dialog box, click **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_\_ f. At the User Login dialog box, accept the default values for **User ID** and **Password** and click **OK**. These fields are set to `admin` and `websphere` during product installation.
- \_\_\_ g. When you run the suite, the modules that are required to do the operation-level tests are published and started: **FoundationModuleApp** and **RiskTestProjectApp**. It can take a few minutes depending on your system resources.
- \_\_\_ h. After the modules are deployed and started, all three test cases run automatically.

2. Examine the results from the test cases.
- a. In the **Events** list, you see two test cases with a verdict of **[Passed]** and one with a verdict of **[Failed]**.



The screenshot shows the Rational Test Studio interface. On the left, there's a tree view under the 'Events' tab. The root node is 'Run Test Cases (RiskTestSuite) [Failed]'. It has three children: 'Test Suite (RiskTestSuite) [Failed]', 'Test Case (Risk\_InputCriterion) [Failed]', and 'Test Variation (FailTest) [Failed]'. The 'Test Case (Risk\_InputCriterion) [Failed]' node has two children: 'Test Variation (Default) [Passed]' and 'Test Variation (AbcCo) [Passed]'. The 'Test Variation (Default) [Passed]' node has one child: '1. Invoke CreditRiskAssessment\InputCriterion'. The 'Test Variation (AbcCo) [Passed]' node also has one child: '1. Invoke CreditRiskAssessment\InputCriterion'. On the right side, there are two sections: 'General Properties' and 'Detailed Properties'. Under 'General Properties', it says 'Select a test configuration for the Run Test Cases event, Continue icon in the Events area to run the test.' Under 'Detailed Properties', it shows the configuration as 'RiskTestSuite', the verdict as 'Failed', and a summary table with 'Total: 1/1', 'Passed: 0', 'Failed: 1', and 'Error: 0'.



## Troubleshooting

If you do not see results similar to the screen capture above, you might want to check whether all the steps were followed correctly. Also, make sure that you save your changes.

- \_\_ b. In the **Events** list, select **Test Variation (FailTest) [Failed]**.

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)

- Run Test Cases (RiskTestSuite) [Failed]
  - Test Suite (RiskTestSuite) [Failed]
    - Test Case (Risk\_InputCriterion) [Failed]
      - Test Variation (Default) [Passed]
        - 1. Invoke CreditRiskAssessment:InputCriterion
      - Test Variation (AbcCo) [Passed]
        - 1. Invoke CreditRiskAssessment:InputCriterion
      - Test Variation (FailTest) [Failed]**
        - 1. Invoke CreditRiskAssessment:InputCriterion

- \_\_ c. Examine the **Results** section. The cause of the test failure (the expected companyName versus the actual companyName) is highlighted, and the error is listed in the **Exception message** window. An **Exception trace** is also provided.

| Name                | Type                | Actual               | Expected |
|---------------------|---------------------|----------------------|----------|
| Output              | CustomerApplication | BusinessO... (!=)    |          |
| accountNumber       | string              | IBM007 (==) IBM007   |          |
| applicationDate     | string              | 06/10/2016 (!=) null |          |
| applicationDecision | boolean             | true (==) true       |          |
| comments            | string              | None (==) None       |          |
| <b>companyName</b>  | string              | BeanCo (!=) IBM      |          |
| contactFirstName    | string              | Landon (==) Landon   |          |
| contactLastName     | string              | Donovan (==) Donovan |          |

Exception message:

```
Variation:[FailTest] Variable:[Output] Path:[companyName] FAIL(Input:[BeanCo]) != Expected:[IBM])
```

- \_\_ d. As time permits, explore the test variations that passed.
- \_\_ e. Close the test suite tab and click **No** when you are prompted to save the test trace.
- \_\_ 3. Remove the applications from the server. Applications that are deployed during a component test run remain deployed until you manually remove them.
- \_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.

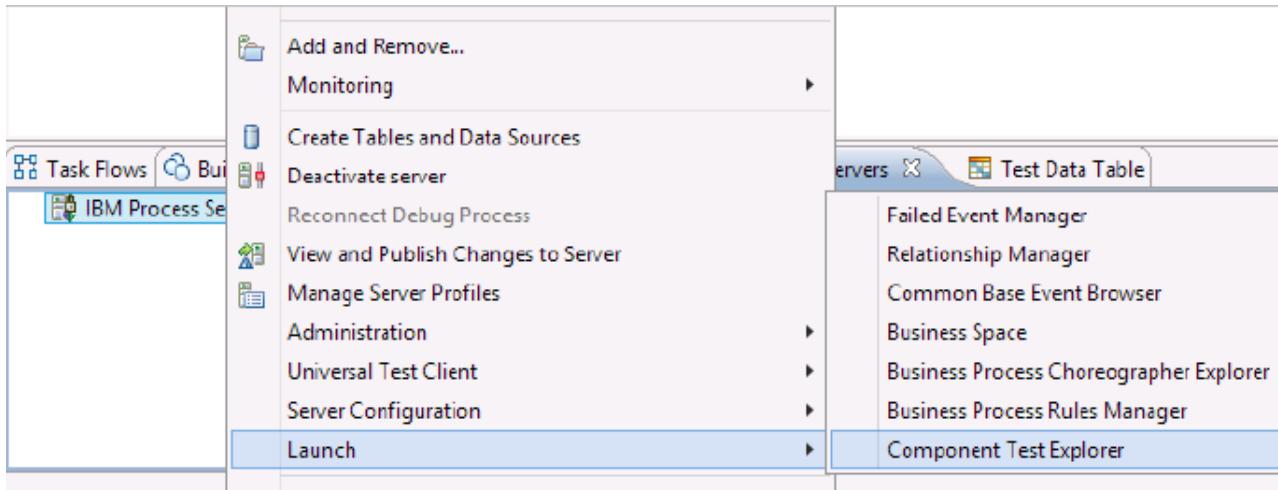
- \_\_\_ b. Click **Remove All** and click **Finish**. It takes a few minutes until the applications are removed from the server. Wait for the server status to change to **Started**, **Synchronized**.



### Information

Though not used in this exercise, IBM Integration Designer also includes a web-based application that is named Component Test Explorer. You can use the Component Test Explorer to manage and run test suites that are deployed to an IBM Integration Designer test environment server. Using the Component Test Explorer, you can:

- Query and display deployed component test projects, test suites, and test cases
- Select and run one or more component test projects, test suites, or test cases and then display and automatically save the results
- Schedule specific times to automatically run one or more component test projects, test suites, or test cases



## **Part 2: Create a component test project with a scenario-based test case**

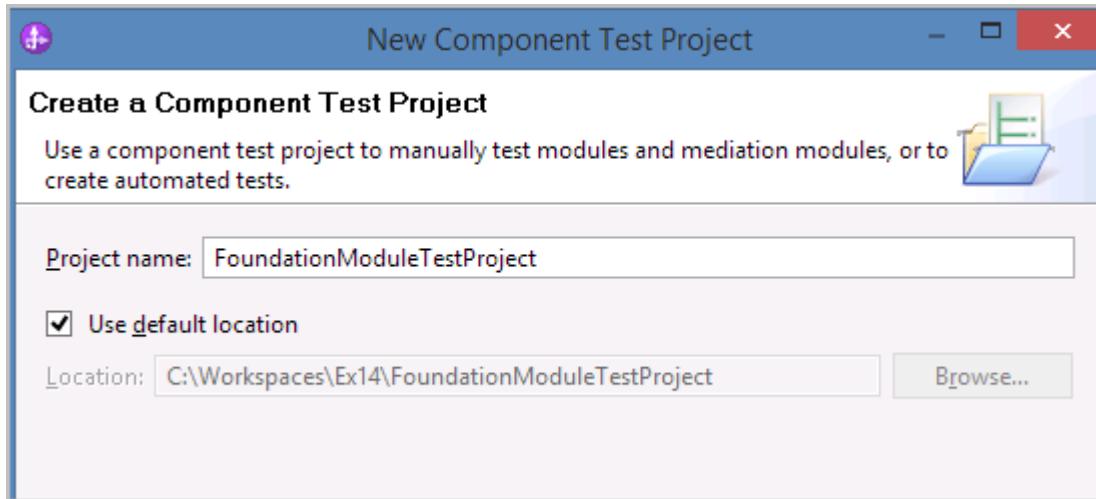
In this portion of the exercise, you create a component test project with a scenario-based test case. Scenario-based test cases invoke a series of interface operations. The test case that you create simulates one of the process paths in your solution. After creating the project, you run the test suite in the IBM Integration Designer integrated test client.

The path you test is for companyName ACME. Because the creditScore returned for ACME is 6, ACME has a creditRisk of MED. An application with a MED creditRisk takes this path through the AccountVerification process:

**Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment > Final Application Review > Create Output > Account Verification Reply.**

Because this path involves a human task, you create a human task emulator to simulate human input.

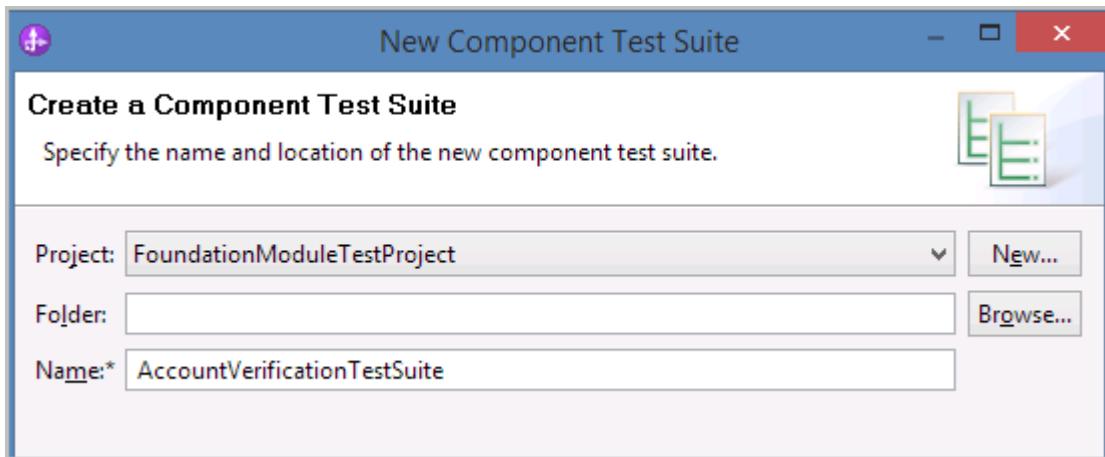
- 1. Create a test project named `FoundationModuleTestProject`.
  - a. Click **File > New > Component Test Project** from the menu options.
  - b. In the **Project name** field, type: `FoundationModuleTestProject`



- c. Click **Finish**.
- 2. Create a component test suite in **FoundationModuleTestProject** that is named `AccountVerificationTestSuite`. The test suite is scenario-based and contains a test case that is named `AccountVerification_MED`. The project includes the following operations:
  - InputCriterion operation in the `AccountVerification` interface
  - InputCriterion operation in the `DetermineApplicantEligibility` interface
  - calculateCreditScore operation in the `creditScoreService` interface
  - InputCriterion operation in the `CreditRiskAssessment` interface
  - InputCriterion operation in the `FinalApplicationReview` interface
- a. In the Business Integration view, expand **FoundationModuleTestProject**.
- b. Right-click **Test Suites** and click **New > Component Test Suite** from the menu.

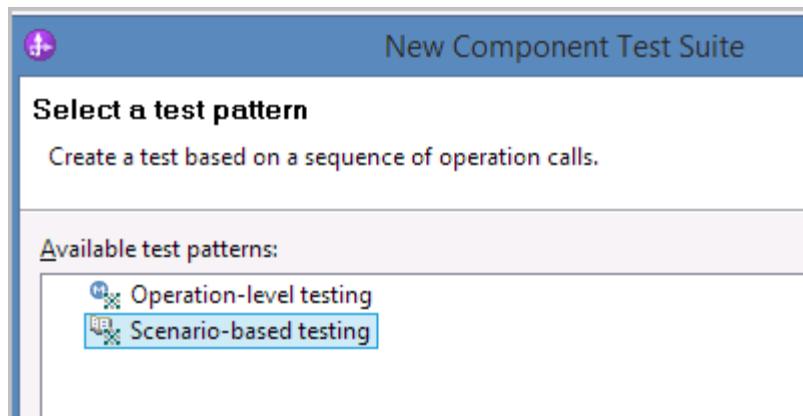
\_\_ c. At the “Create a Component Test Suite” pane, take the following actions:

- Leave the **Folder** field empty.
- In the **Name** field, type: AccountVerificationTestSuite



\_\_ d. Click **Next**.

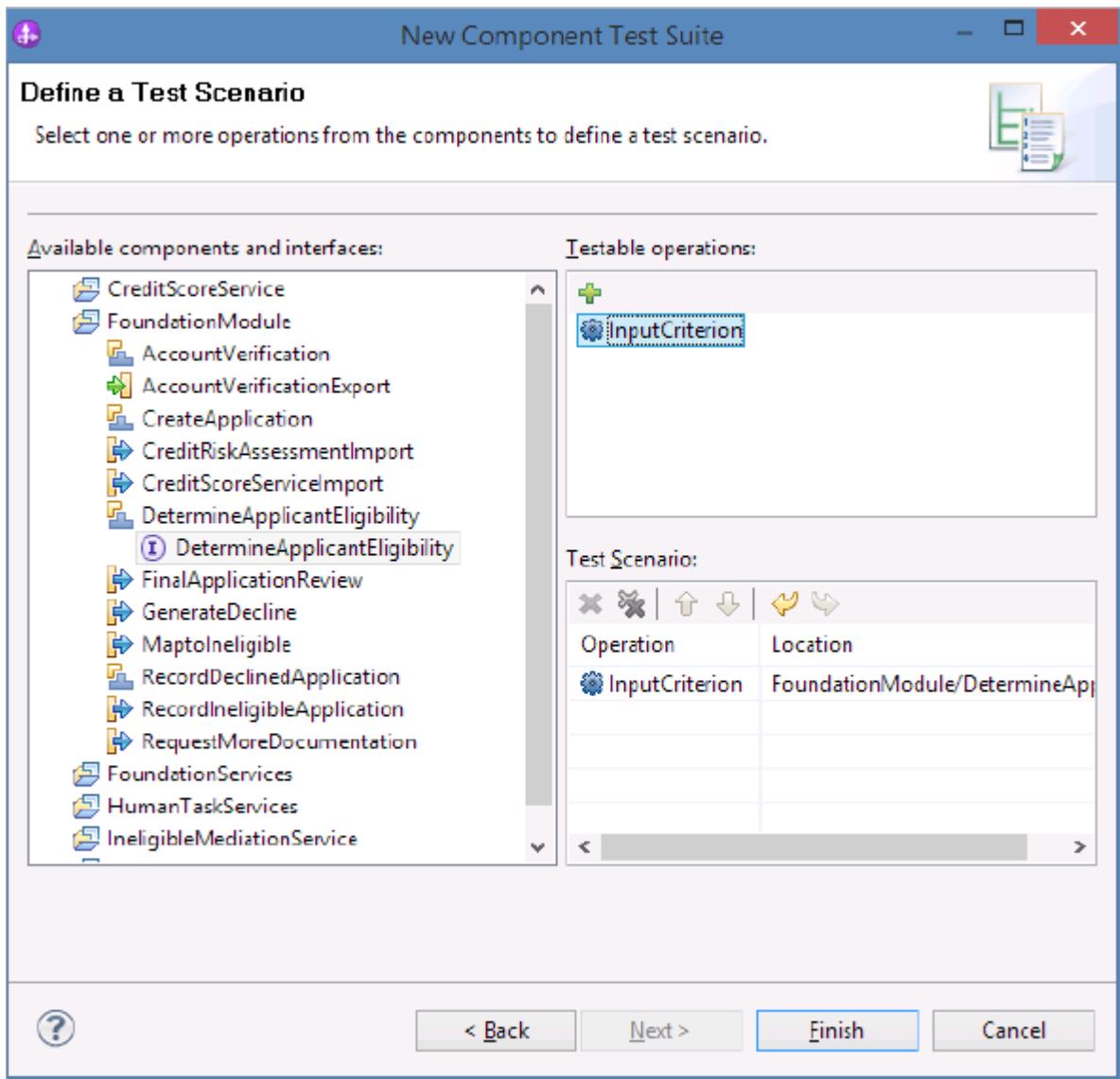
\_\_ e. In the “Select a test pattern” pane, select **Scenario-based testing**.



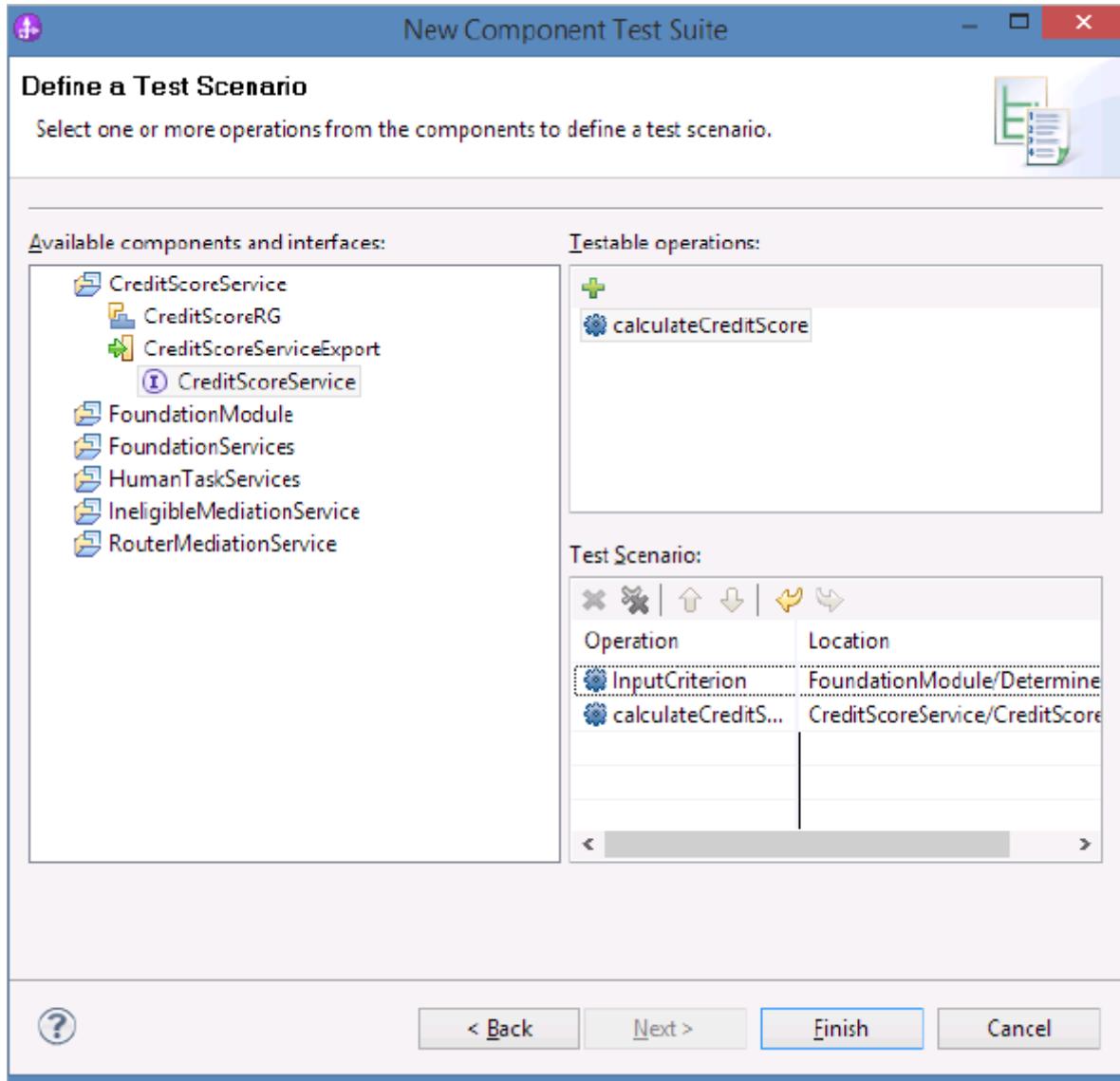
\_\_ f. Click **Next**.

\_\_ g. In the “Define a Test Scenario” pane, take the following actions:

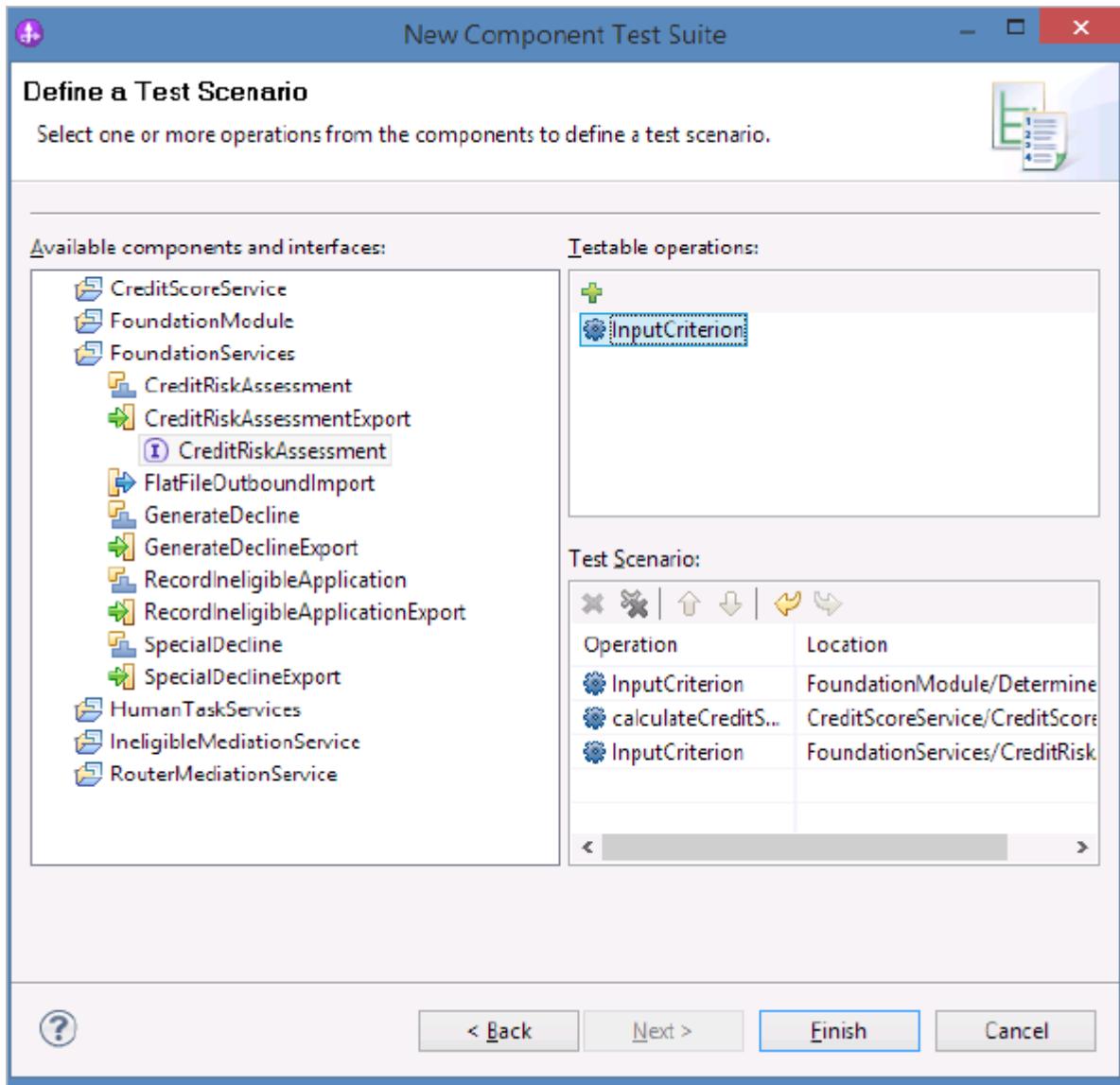
- In the **Test case name** field, type: AccountVerification\_MED
- In the “Available components and interfaces” window, expand **FoundationModule > DetermineApplicantEligibility** and select the **DetermineApplicantEligibility** interface.
- In the “Testable operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.



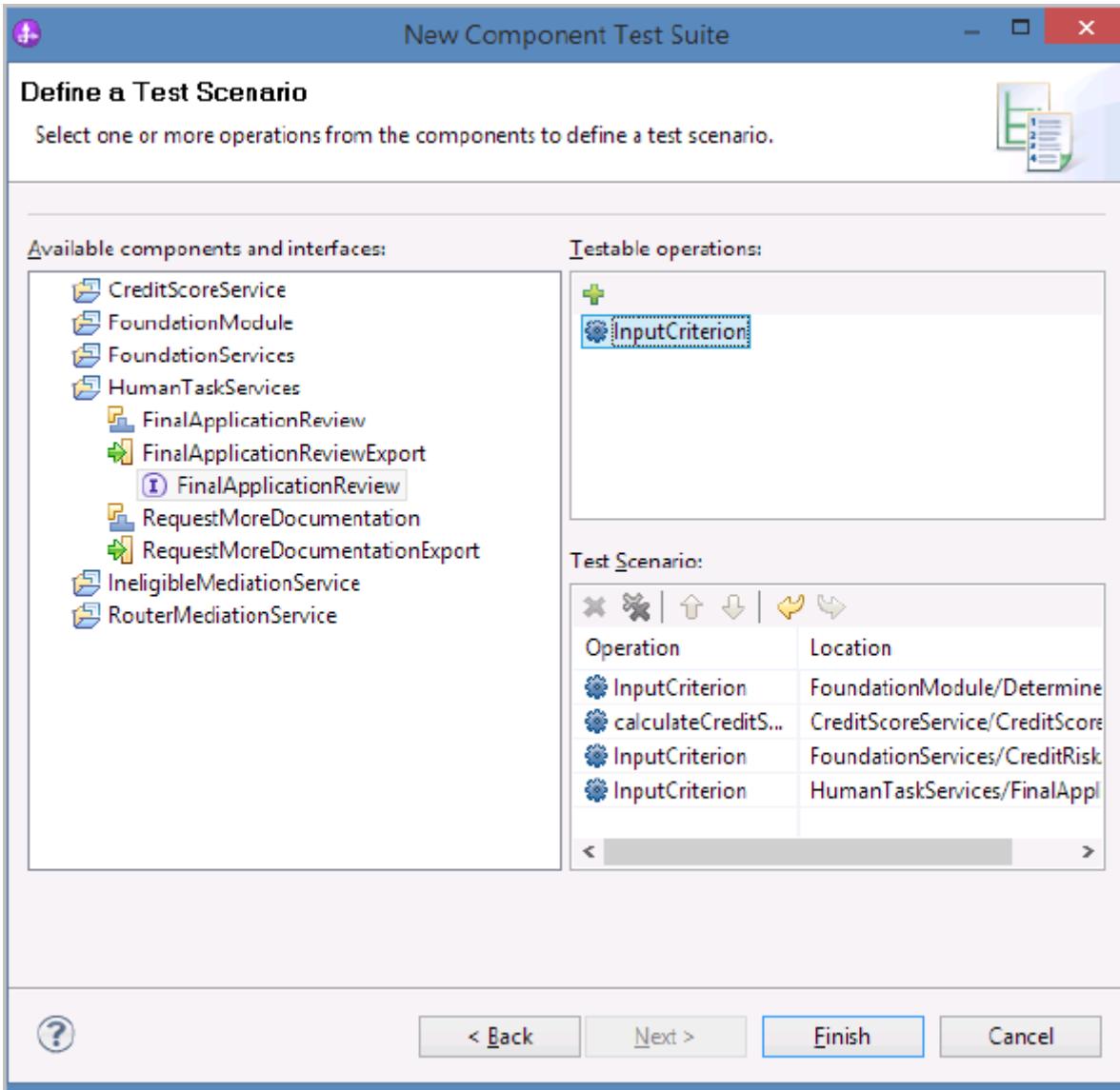
- In the “Available components and interfaces” window, expand **CreditScoreService** > **CreditScoreServiceExport** and select the **CreditScoreService** interface.
- In the “Testable operations” window, double-click **calculateCreditScore** to add the operation to the “Test Scenario” window.



- In the “Available components and interfaces” window, expand **FoundationServices > CreditRiskAssessmentExport** and select the **CreditRiskAssessment** interface.
- In the “Testable operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.



- In the “Available components and interfaces” window, expand **HumanTaskServices > FinalApplicationReviewExport** and select the **FinalApplicationReview** interface.
- In the “Testable Operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.



- Verify that the completed “Test Scenario” window contains the following four operations:

| Test Scenario:       |                                          |
|----------------------|------------------------------------------|
| Operation            | Location                                 |
| InputCriterion       | FoundationModule/DetermineApplica...     |
| calculateCreditScore | CreditScoreService/CreditScoreService... |
| InputCriterion       | FoundationServices/CreditRiskAssessm...  |
| InputCriterion       | HumanTaskServices/FinalApplicationR...   |

- h. Click **Finish**.
- 3. Populate the credappin and credappout values for Invoke DetermineApplicantEligibility:InputCriterion in the Default test variation. Use the credappin values in `c:\labfiles\Support Files\Ex14\EX14_Scenario_credappin.xml` and the credappout values in `EX14_Scenario_credappout.xml`. Set applicationDate in credappout to **Do Not Care**.
  - a. In the Test Data Table tab, go to **Invoke DetermineApplicantEligibility:InputCriterion**. You can double-click the test Data Table tab to maximize the view.
  - b. Right-click **credappin** and click **Import from File**.
  - c. Browse to `C:\labfiles\Support Files\Ex14\`.

- \_\_ d. Select **EX14\_Scenario\_credappin.xml** and click **Open**.

| Name                                     | Type                | Set               |
|------------------------------------------|---------------------|-------------------|
| !-- Invoke DetermineApplicantEligibility |                     |                   |
| credappin                                | CustomerApplication | [ab]              |
| accountNumber                            | string              | [ab] ACM002       |
| applicationDate                          | string              | [ab] 06/10/2016   |
| applicationDecision                      | boolean             | [ab] true         |
| comments                                 | string              | [ab] None         |
| companyName                              | string              | [ab] ACME         |
| contactFirstName                         | string              | [ab] Torsten      |
| contactLastName                          | string              | [ab] Frings       |
| contactPhoneNumber                       | string              | [ab] 905-555-7234 |
| creditRating                             | string              | [ab] C+           |
| creditReportNeeded                       | boolean             | [ab] true         |
| creditRisk                               | string              | [ab] MED          |
| creditScore                              | int                 | [ab] 0            |
| customerCity                             | string              | [ab] Berlin       |
| customerCountry                          | string              | [ab] Germany      |
| eligibleApplication                      | boolean             | [ab] true         |
| ineligibleReason                         | string              | [ab] None         |
| pricingCode                              | string              | [ab] 23           |
| pricingScore                             | string              | [ab] 17           |
| productName                              | string              | [ab] Labels       |
| requestAccountAmount                     | int                 | [ab] 10000        |

- \_\_ e. Scroll to the bottom of the input data and locate the `credappout` section under `VerifyInputCriterion`.
- \_\_ f. In the `credappout` row, right-click the value in the **Expected** column and click **Add Children** from the menu.
- \_\_ g. Right-click `credappout` and click **Import from File** from the menu.
- \_\_ h. Browse to `C:\labfiles\Support Files\Ex14\`.
- \_\_ i. Select **EX14\_Scenario\_credappout.xml** and click **Open**.

- \_\_ j. When the values are imported, right-click the **Expected** value for `applicationDate` and click **Set To > Do Not Care** from the menu.

| !-- Verify InputCriterion |                     |                     |  |
|---------------------------|---------------------|---------------------|--|
| credappout                | CustomerApplication | ab=0                |  |
| accountNumber             | string              | ab=(=) ACM002       |  |
| applicationDate           | string              | ab=                 |  |
| applicationDecision       | boolean             | ab=(=) true         |  |
| comments                  | string              | ab=(=) None         |  |
| companyName               | string              | ab=(=) ACME         |  |
| contactFirstName          | string              | ab=(=) Torsten      |  |
| contactLastName           | string              | ab=(=) Frings       |  |
| contactPhoneNumber        | string              | ab=(=) 905-555-7234 |  |
| creditRating              | string              | ab=(=) C+           |  |
| creditReportNeeded        | boolean             | ab=(=) true         |  |
| creditRisk                | string              | ab=(=) MED          |  |
| creditScore               | int                 | ab=(=) 0            |  |

- \_\_ 4. Populate the request and calculateCreditScoreReturn values for Invoke CreditScoreServiceExport:calculateCreditScore in the Default test variation. Use the request values in `C:\labfiles\Support Files\Ex14\EX14_Scenario_request.xml` and the calculateCreditScoreReturn values in `EX14_Scenario_return.xml`. Set the dateRequested value in calculateCreditScoreReturn to **Do Not Care**.
- Scroll to **Invoke CreditScoreServiceExport:calculateCreditScore**.
  - Expand **request > Envelope > Body > any > calculateCreditScore > request**.
  - Right-click **request** and click **Import from File** from the menu.
  - Browse to `C:\labfiles\Support Files\Ex14\`.
  - Select `EX14_Scenario_request.xml` and click **Open**.

| Name                                                              | Type                     | Set                    |
|-------------------------------------------------------------------|--------------------------|------------------------|
| !-- Invoke CreditScoreServiceExport:calculateCreditScore(request) |                          |                        |
| serviceAddress                                                    | String                   | ab http://localhost... |
| request                                                           | SoapMessage12            | ab                     |
| Envelope *                                                        | Envelope                 | ab                     |
| Header                                                            | Header                   | ab                     |
| Body *                                                            | Body                     | ab                     |
| any *                                                             | anyType[]                | ab                     |
| calculateCreditScore *                                            | calculateCreditScore_... | ab                     |
| request *                                                         | CreditCheckRequest       | ab                     |
| accountNumber                                                     | string                   | ab ACM002              |
| companyName                                                       | string                   | ab ACME                |
| creditScore                                                       | int                      | ab 0                   |
| dateRequested                                                     | string                   | ab 06/10/2016          |

To verify the values, you can expand **request** again after doing the import.

- \_\_\_ f. Alternatively, enter the following values:
- accountNumber: ACM002
  - companyName: ACME
  - creditScore: 0
  - dateRequested: 06/10/2016
- \_\_\_ g. Scroll to **Verify calculateCreditScore**.
- \_\_\_ h. In the **response** row, right-click the value in the **Expected** column and click **Add Children** from the menu.
- \_\_\_ i. Expand **response > Envelope > Body > any > calculateCreditScoreResponse > calculateCreditScoreReturn**.
- \_\_\_ j. Right-click **calculateCreditScoreReturn** and click **Import from File** from the menu.
- \_\_\_ k. Browse to C:\labfiles\Support Files\Ex14\.
- \_\_\_ l. Select **EX14\_Scenario\_return.xml** and click **Open**.
- \_\_\_ m. If necessary, expand **calculateCreditScoreReturn**.
- \_\_\_ n. In the **dateRequested** row, right-click the value in the **Expected** column and click **Set To > Do Not Care** from the menu.

| !.. Verify calculateCreditScore |                          |                                   |
|---------------------------------|--------------------------|-----------------------------------|
| response                        | SoapMessage12            | ab <sup>0</sup> null<br>(-) null  |
| Envelope *                      | Envelope                 | ab <sup>0</sup> null<br>(-) null  |
| Header                          | Header                   | ab <sup>0</sup>                   |
| Body *                          | Body                     | ab <sup>0</sup> null<br>(-) null  |
| any *                           | anyType[]                | ab <sup>0</sup> null<br>(-) null  |
| calculateCreditScoreResponse *  | calculateCreditScoreR... | ab <sup>0</sup> null<br>(-) null  |
| calculateCreditScoreReturn *    | CreditCheckRequest       | ab <sup>0</sup>                   |
| accountNumber                   | string                   | ab <sup>0</sup> ) ACM002          |
| companyName                     | string                   | ab <sup>0</sup> ) ACME            |
| creditScore                     | int                      | ab <sup>0</sup> ) 6               |
| dateRequested                   | string                   | ab <sup>0</sup> .....<br>(-) null |
| anyAttribute                    | anySimpleType[]          | ab <sup>0</sup>                   |
| anyAttribute                    | anySimpleType[]          | ab <sup>0</sup>                   |
| attachments                     | UnreferencedAttach...    | ab <sup>0</sup>                   |

- \_\_\_ o. Alternatively, enter the following values:
- accountNumber: ACM002
  - companyName: ACME
  - creditScore: 6
  - dateRequested: (set to **Do Not Care**)
- \_\_\_ p. Save your changes.

5. Populate the Input and Output values for Invoke CreditRiskAssessmentExport:InputCriterion in the Default test variation. Use the Input values in C:\labfiles\Support Files\Ex14\EX14\_Scenario\_input.xml and the Output values in EX14\_Scenario\_output.xml.
- \_\_ a. Scroll to **Invoke CreditRiskAssessmentExport:InputCriterion(Input)**.
  - \_\_ b. Right-click **Input** and click **Import from File**.
  - \_\_ c. Browse to C:\labfiles\Support Files\Ex14\.
  - \_\_ d. Select **EX14\_Scenario\_input.xml** and click **Open**.
  - \_\_ e. Alternatively, enter the values. If you enter the values manually, change the **creditScore** value to 6.

| Name                                                        | Type                | Value             |
|-------------------------------------------------------------|---------------------|-------------------|
| !-- Invoke CreditRiskAssessmentExport:InputCriterion(Input) |                     |                   |
| Input                                                       | CustomerApplication | [ab] ACM002       |
| accountNumber                                               | string              | [ab] 06/10/2016   |
| applicationDate                                             | string              | [ab] true         |
| applicationDecision                                         | boolean             | [ab] None         |
| comments                                                    | string              | [ab] ACME         |
| companyName                                                 | string              | [ab] Torsten      |
| contactFirstName                                            | string              | [ab] Frings       |
| contactLastName                                             | string              | [ab] 905-555-7234 |
| contactPhoneNumber                                          | string              | [ab] C+           |
| creditRating                                                | string              | [ab] true         |
| creditReportNeeded                                          | boolean             | [ab] MED          |
| creditRisk                                                  | string              | [ab] 6            |
| creditScore                                                 | int                 | [ab] Berlin       |
| customerCity                                                | string              | [ab] Germany      |
| customerCountry                                             | string              | [ab] true         |
| eligibleApplication                                         | boolean             | [ab] None         |
| ineligibleReason                                            | string              | [ab] 23           |
| pricingCode                                                 | string              | [ab] 17           |
| pricingScore                                                | string              | [ab] Labels       |
| productName                                                 | string              | [ab] 10000        |
| requestAccountAmount                                        | int                 |                   |

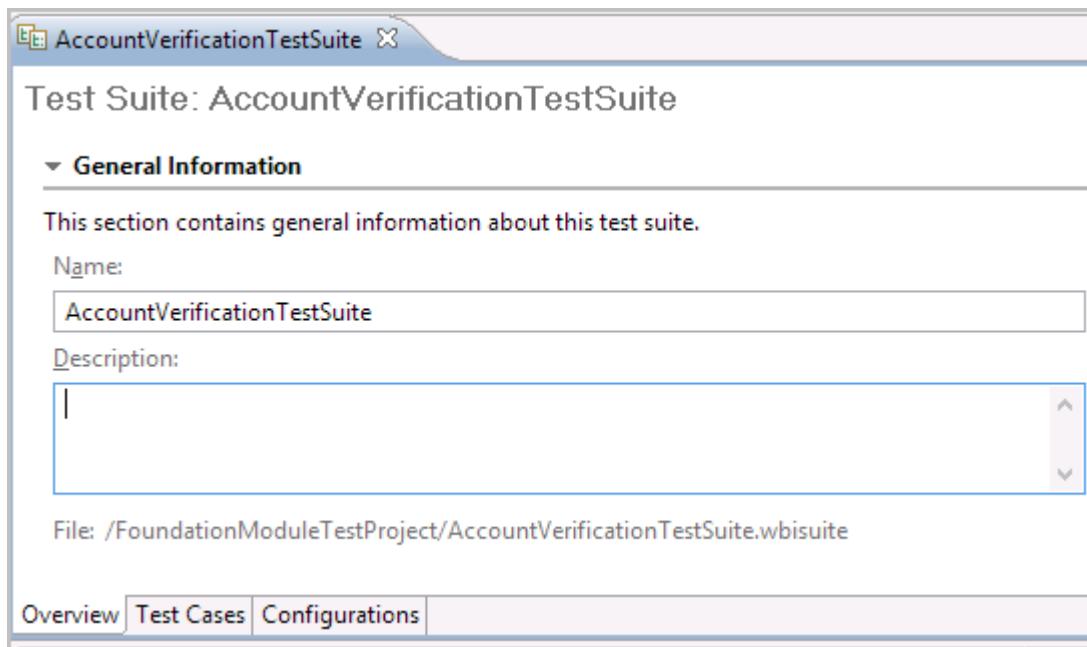
- \_\_ f. Scroll to **Verify InputCriterion**.
- \_\_ g. In the **Output** row, right-click the value in the **Expected** column and click **Add Children** from the menu.
- \_\_ h. Right-click **Output** and click **Import from File** from the menu.
- \_\_ i. Browse to C:\labfiles\Support Files\Ex14\.
- \_\_ j. Select **EX14\_Scenario\_output.xml** and click **Open**.

- \_\_ k. Alternatively, enter the values. If you enter the values manually, change the creditScore value to 6.

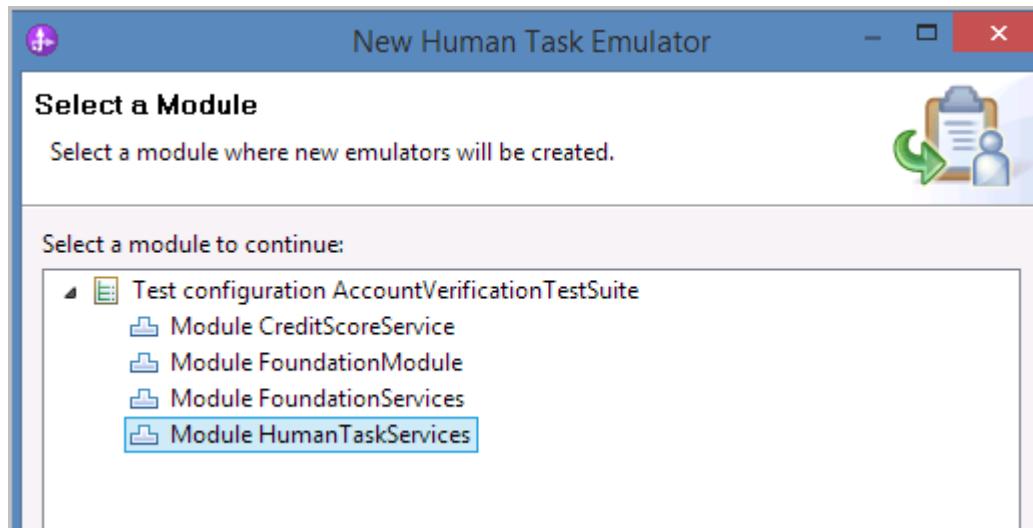
| Name                 | Type                | Expect...          |
|----------------------|---------------------|--------------------|
| Output               | CustomerApplication | ab(=) 0            |
| accountNumber        | string              | ab(=) ACM002       |
| applicationDate      | string              | ab(=) 06/10/2016   |
| applicationDecision  | boolean             | ab(=) true         |
| comments             | string              | ab(=) None         |
| companyName          | string              | ab(=) ACME         |
| contactFirstName     | string              | ab(=) Torsten      |
| contactLastName      | string              | ab(=) Frings       |
| contactPhoneNumber   | string              | ab(=) 905-555-7234 |
| creditRating         | string              | ab(=) C+           |
| creditReportNeeded   | boolean             | ab(=) true         |
| creditRisk           | string              | ab(=) MED          |
| creditScore          | int                 | ab(=) 6            |
| customerCity         | string              | ab(=) Berlin       |
| customerCountry      | string              | ab(=) Germany      |
| eligibleApplication  | boolean             | ab(=) true         |
| ineligibleReason     | string              | ab(=) None         |
| pricingCode          | string              | ab(=) 23           |
| pricingScore         | string              | ab(=) 17           |
| productName          | string              | ab(=) Labels       |
| requestAccountAmount | int                 | ab(=) 10000        |

- \_\_ l. Save your changes.

- 6. Create a human task emulator for the final operation in the test case: the InputCriterion operation in the FinalApplicationReview interface.
- a. On the **AccountVerificationTestSuite** tab, switch to the **Configurations** tab.

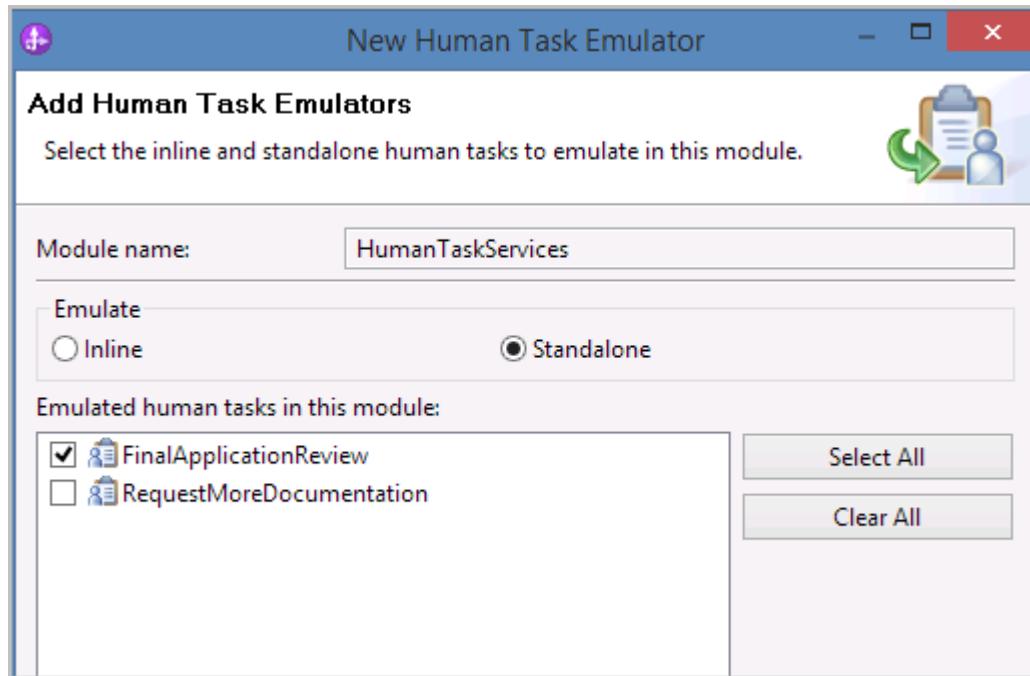


- b. Under **Module HumanTaskServices**, right-click **Human Task Emulators** and click **Add > Human Task Emulator**.
- c. At the “Select a Module” panel, select **Module HumanTaskServices**.

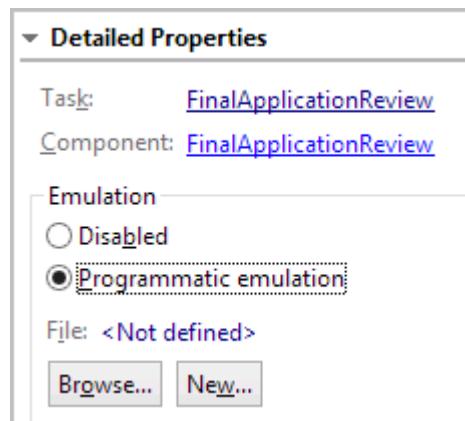


- d. Click **Next**.

- \_\_ e. At the Add Human Task Emulators pane:
- In the **Emulate** section, select the **Standalone** option.
  - In the “Emulated human tasks in this module” window, select **FinalApplicationReview**.



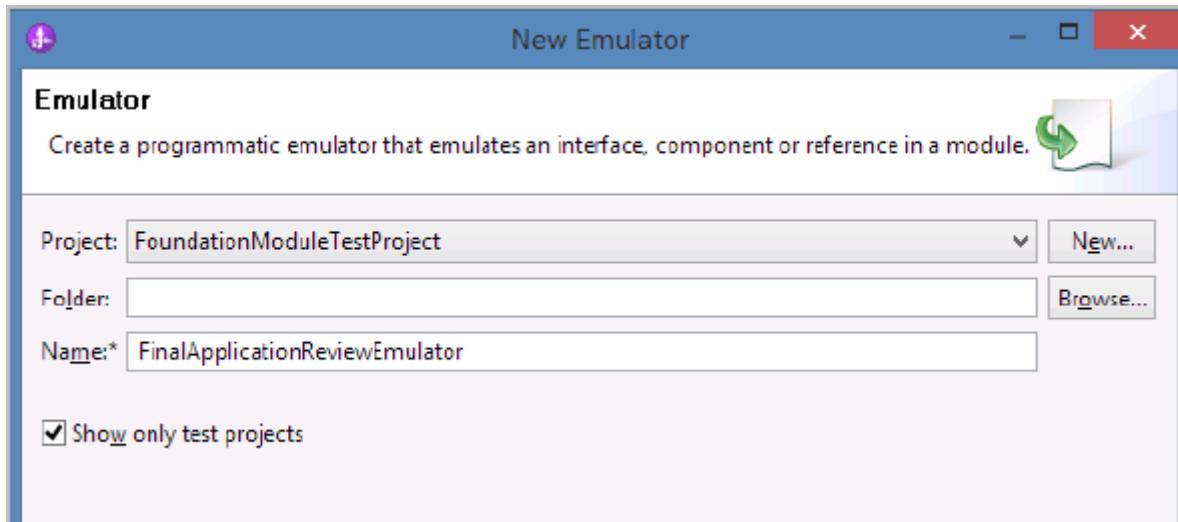
- \_\_ f. Click **Finish**.
- \_\_ g. In the **Detailed Properties** section, in the **Emulation** box, click **Programmatic emulation**.



- \_\_ h. Click **New**.

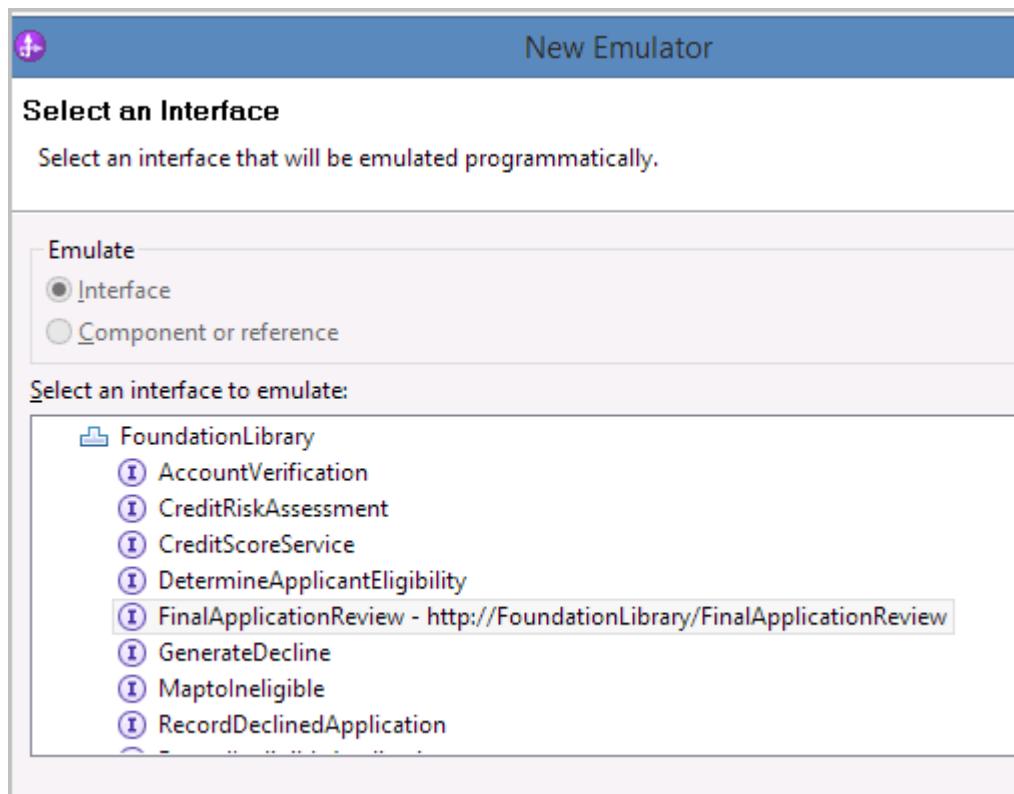
\_\_ i. In the New Emulator dialog box, on the Emulator pane, take the following actions:

- Verify that the **Project** field is set to: FoundationModuleTestProject
- Leave the **Folder** field empty.
- In the **Name** field, type: FinalApplicationReviewEmulator



\_\_ j. Click **Next**.

\_\_ k. At the “Select an Interface” pane, verify that the **FinalApplicationReview** interface is selected.



\_\_ l. Click **Next**.

\_\_ m. At the **Java Class** pane, accept the default options.

- \_\_ n. Click **Finish**.
- \_\_ o. In the **Interface Summary** section, click the **CustomerApplication\_InputCriterion(CustomerApplication\_input)** link.

The screenshot shows a user interface for managing operations in an interface. At the top, there is a header labeled "Interface Summary". Below it, a message says "A list of operations defined in the interface:". Underneath, there is a section titled "Operations" which contains a single link: "[CustomerApplication\\_InputCriterion\(CustomerApplication\\_input\)](#)". This link is highlighted with a blue border, indicating it is selected or being interacted with.

- \_\_ p. Select **Java snippet editor**.

The screenshot shows a dialog box titled "Define Emulation". It contains a message: "An emulation for operation InputCriterion has not been defined. Select one of the editors below to start a new emulation." Below this, there is a section titled "Emulate using" with two options: "Visual snippet editor" and "Java snippet editor". The "Java snippet editor" option is selected, indicated by a checked radio button. At the bottom of the dialog is a "Define Emulation" button.

- \_\_ q. Click **Define Emulation**.
- \_\_ r. In Windows Explorer, browse to `C:\labfiles\Support Files\EX14\`.
- \_\_ s. Open `FinalApplicationReviewEmulatorCode.txt` in a text editor such as Notepad.
- \_\_ t. Copy the code and paste it in the Java Editor over the existing text. Be sure to remove `return null;`

#### Java Editor

The screenshot shows a Java code editor window. The code is as follows:

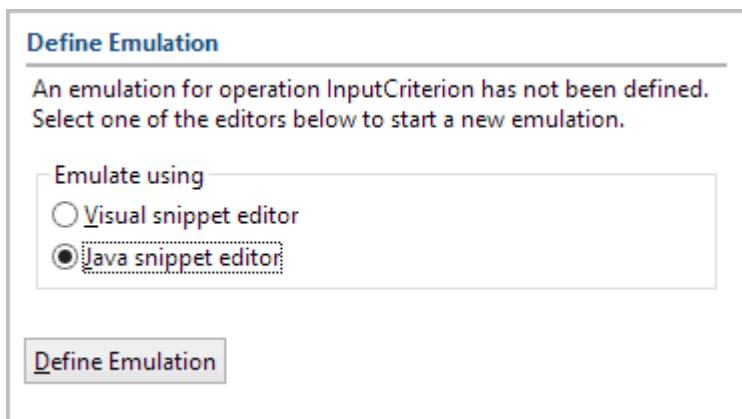
```
System.out.println(" Start - FinalApplicationReview Emulation route ...");
System.out.println(" End - FinalApplicationReview Emulation route ...");
return input;
```

- \_\_ u. Alternatively, enter the following Java code:

```
System.out.println
(" Start - FinalApplicationReview Emulation route ...");
System.out.println
(" End - FinalApplicationReview Emulation route ...");
return input;
```

- \_\_ v. Save your changes and close the **FinalApplicationReviewEmulator** tab.
- \_\_ w. Close `FinalApplicationReviewEmulatorCode.txt` and Windows Explorer.

- \_\_\_ x. On the test suite editor tab, in the **Detailed Properties** section, verify that the programmatic emulator file is listed in the **Emulation** box. Also, confirm that **Immediately** is selected in the **Claim** box.



- \_\_\_ y. Save your changes and close the **AccountVerificationTestSuite** tab.

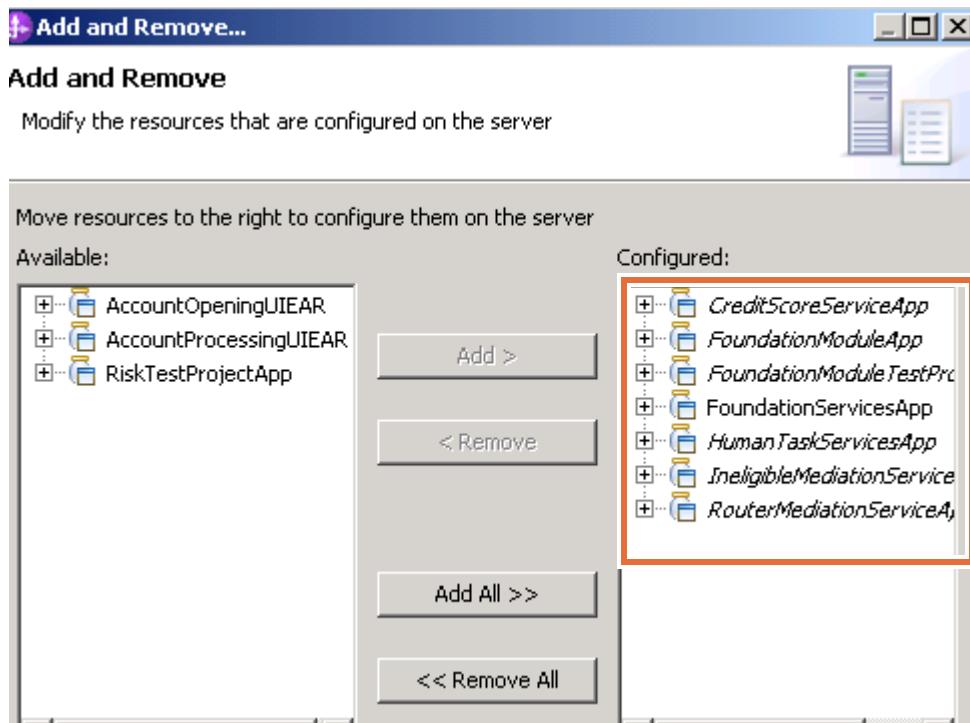
### **Part 3: Run component test project test suites in the IBM Integration Designer integrated test environment**

In this portion of the exercise, you run the AccountVerificationTestSuite in IBM Integration Designer.

To run AccountVerificationTestSuite:

- \_\_\_ 1. Start the server (if it is not already running) and deploy all of the modules.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Start** from the menu (if necessary). Wait for the startup process to complete before continuing.  
The server is started when the message `Server server1 open for e-business` is displayed in the Server Logs view. The server status also changes to **Started** in the Servers view.
- \_\_\_ b. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove**.

- \_\_\_ c. Click **Add All** to include the projects in the **Configured** list. If system resources are an issue, do *not* publish AccountOpeningUIEAR, AccountProcessingUIEAR, or RiskTestProjectApp.



- \_\_\_ d. Click **Finish**.
- \_\_\_ e. Wait until the modules are published and started. It can take several minutes, depending on your system resources. To see the status of the modules, expand **IBM Process Server v8.5.7 at localhost** in the **Servers** view.

If any module is not started, then right-click the module and click **Restart**. If the status of the server is **Publishing**, then right-click the server and click **View and Publish Changes to Server**.

You can also look for `Application started: [ApplicationName]` messages in the **Server Logs** view.

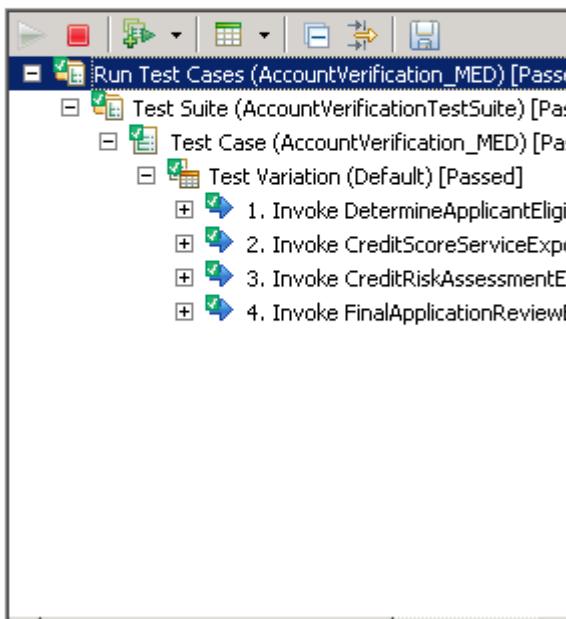
- \_\_\_ 2. Run the test suite in IBM Integration Designer.
- \_\_\_ a. In the Business Integration view, expand **FoundationModuleTestProject > Test Suites > AccountVerificationTestSuite**.
- \_\_\_ b. Right-click **AccountVerification\_MED** and click **Run Test Cases** from the menu.
- \_\_\_ c. Click **Continue** on the **Events** toolbar.
- \_\_\_ d. When the **Deployment Location** dialog box is displayed, select **IBM Process Server v8.5.7 at localhost** and click **Finish**.
- \_\_\_ e. When the **User Login** prompt is displayed, accept the default options for **User Name** and **Password**; click **OK**.
- \_\_\_ f. Wait for the test to complete.

- \_\_\_ g. When the test run is complete, you see four events that represent the invocation of the interface operations. If you receive an error, you can redo the component test scenario and confirm that all the entries were done correctly.

## Integration Test Client: Run\_AccountVerification\_MED

### Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



### General Properties

### Detailed Properties

Select a test configuration for the Run Test Cases event, then click the Continue icon in the Events area to run the test. [More...](#)

To set the environment variables for this test, go to the selected [test configuration](#).

**Configuration:** AccountVerification\_MED

|           |        |
|-----------|--------|
| Verdict : | Passed |
| Total:    | 1/1    |
|           |        |
| Passed:   | 1      |
|           |        |
| Failed:   | 0      |
|           |        |
| Error:    | 0      |
|           |        |



### Troubleshooting

If you do not see the results similar to the screen capture, then make sure that you completed steps 3j and 4n. Also, make sure that you save your changes.

- \_\_\_ h. As time permits, explore the test results.
  - \_\_\_ i. Close the test tab and do not save the test trace.
- \_\_\_ 3. Remove all of the projects from the server.
- \_\_\_ a. In the **Servers** view, right-click **IBM Process Server v8.5.7 at localhost** and click **Add and Remove** from the menu.
  - \_\_\_ b. Click **Remove All** and click **Finish**.
  - \_\_\_ c. Optionally, stop the server.
- \_\_\_ 4. Close IBM Integration Designer.

## End of exercise

## Exercise review and wrap-up

In this exercise, you created and ran component test projects in the IBM Integration Designer test environment.

# Exercise 15.Bringing the UTE Process Server online

## What this exercise is about

In this exercise, you bring the UTE Process Server online and connect with Process Center.

## What you should be able to do

After completing this exercise, you should be able to:

- Verify the status of the connected Process Server
- Run wsadmin commands for environment configuration
- Exchange SSL certificates between Process Center and Process Server

## Introduction

After you install and configure IBM Business Process Manager, you can use the wsadmin command to change the Process Server settings that are used for connecting to the Process Center. To deploy snapshots to the online Process Server from the Process Center Console, you need to bring the Process Server online. In this exercise, you bring the UTE Process Server online and connect with the Process Center.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Process Server test environment. Instructions for creating the lab environment are included in the exercise appendixes.

This exercise is a prerequisite for Exercise 16, Exploring Process Center.

## Exercise instructions

### Part 1: Start IBM Process Center and the Process Server test environment

You can check the state of the Process Center environment by double-clicking the **Deployment Manager Administrative Console** shortcut on the desktop. If the web server does not return an error, the Process Center is running. If the Process Center environment is already running, then go to Step 5 to start the UTE server.

- 1. Start the Process Center deployment manager.
  - a. On your Windows desktop in the lab environment, select the shortcut that is titled: **Start Process Center deployment manager**. It might be necessary to select the shortcut to completely view its description. Double-click the shortcut or press Enter to start the server.



A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts.

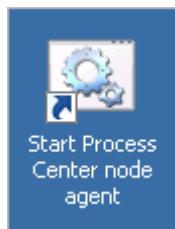
- b. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.

```

Start Process Center deployment manager
CWUPO0001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
 C:\IBM\BPM\v8.5\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 2820
Press any key to continue . . . =

```

- 2. Start the Process Center node agent.
  - a. On your Windows desktop, select the shortcut that is titled **Start Process Center node agent**. Double-click the shortcut or press Enter to start the server.



- \_\_\_ b. A DOS command window is displayed. Press any key to continue when prompted.

```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
C:\IBM\BPM\v8.5\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 3220
Press any key to continue . . .
```

- \_\_\_ 3. Start the single cluster.

- \_\_\_ a. On your Windows desktop, select the **Start Process Center Cluster** shortcut. Double-click the shortcut or press Enter to start the cluster member.



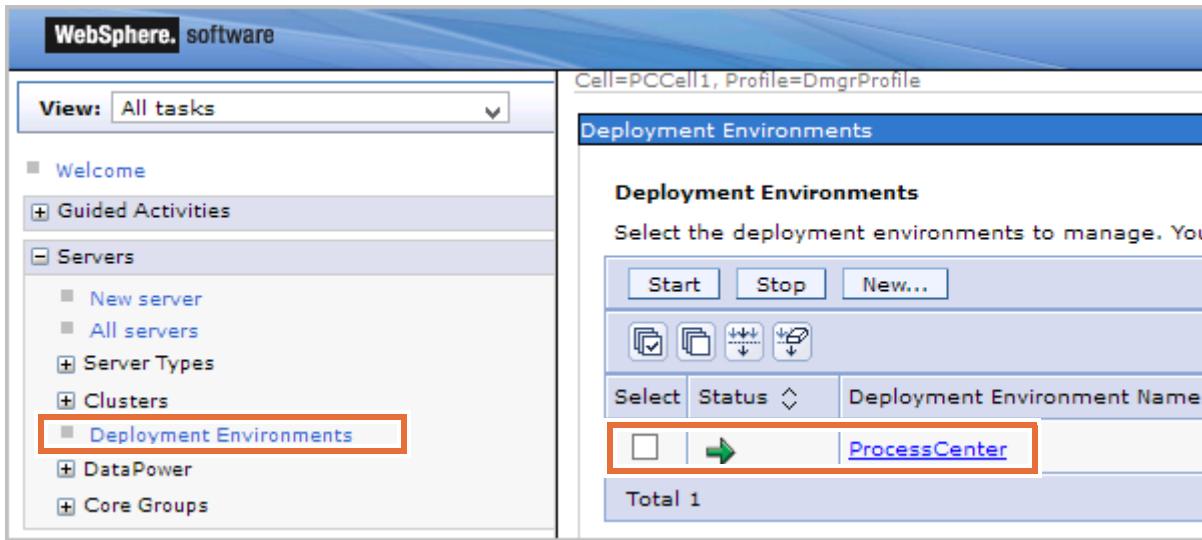
- \_\_\_ b. A DOS command window is displayed. Press any key to continue when prompted.

```
CWUP00001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
C:\IBM\BPM\v8.5\profiles\Node1Profile\logs\SingleClusterMember1\startServer.log
ADMU0128I: Starting tool with the Node1Profile profile
ADMU3100I: Reading configuration for server: SingleClusterMember1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server SingleClusterMember1 open for e-business; process id is 3964
Press any key to continue . . .
```

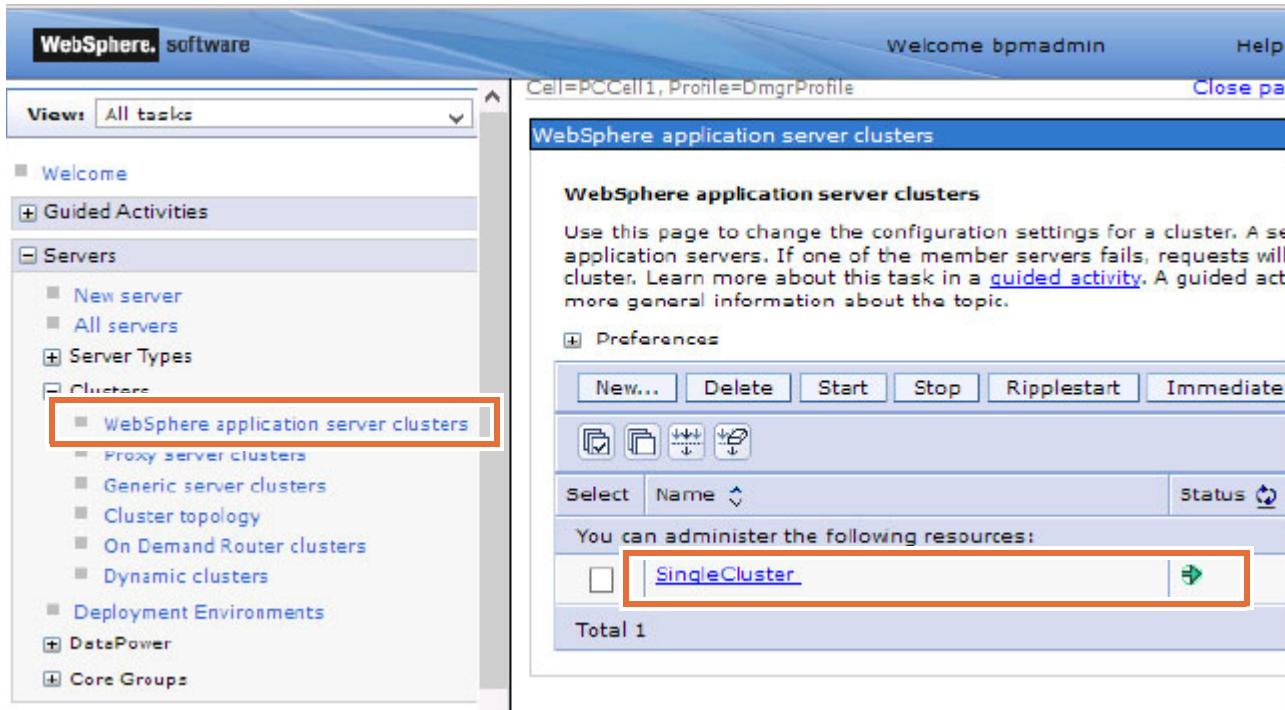
- \_\_\_ 4. Verify that the Process Center Single cluster is started.

- \_\_\_ a. Double-click the **Deployment Manager Administrative Console** shortcut on the desktop.
- \_\_\_ b. Log in by entering `bpmadmin` in the **User ID** field and `websphere` in the **Password** field.

- \_\_ c. Click **Servers > Deployment Environments**.

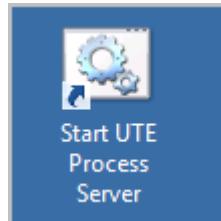


- \_\_ d. Verify that the deployment environment is started and the status is green.  
 \_\_ e. Click **Servers > Clusters > WebSphere application server clusters** to verify that the single cluster is running.



- \_\_ f. Click **Logout** to exit.  
 \_\_ g. Close the browser.

- 5. If the UTE Process Server is already running, then skip to the Part 2. Otherwise, complete the following steps:
  - a. On your desktop, double-click the **Start UTE Process Server** shortcut.

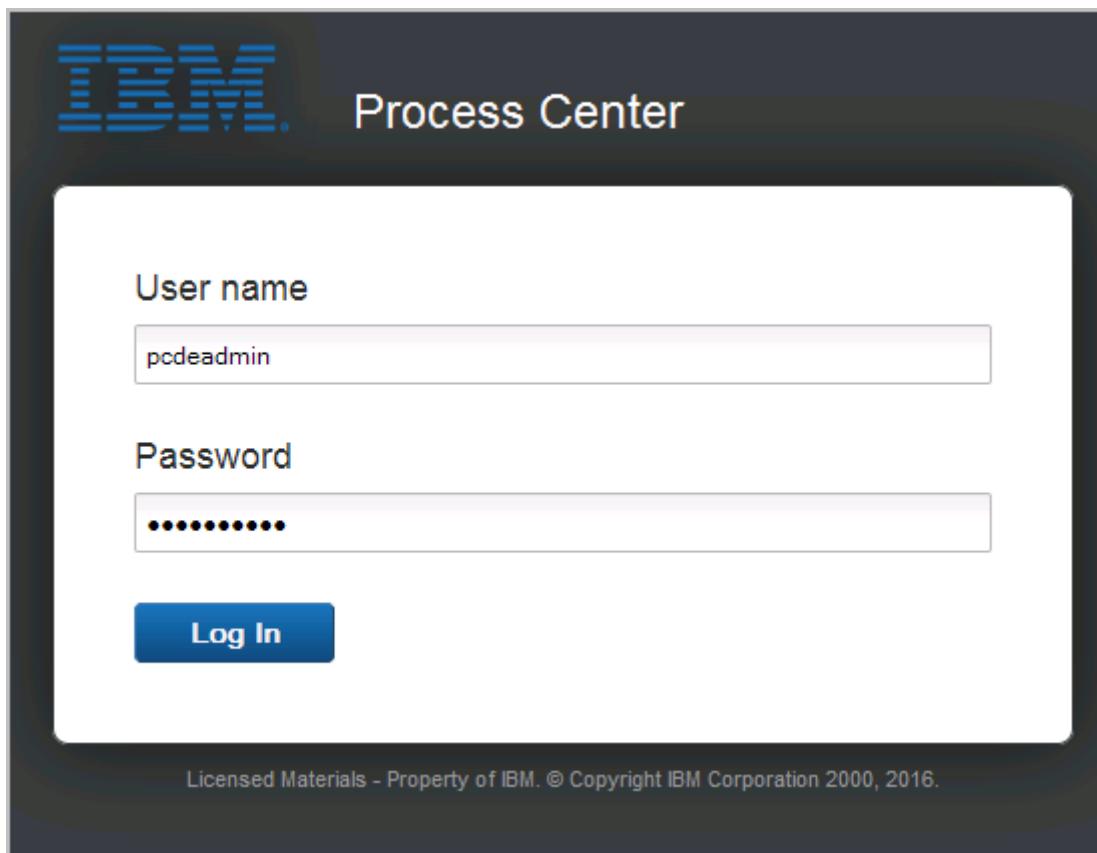


A DOS command window is displayed, and the UTE Process Center server instance starts. The command window closes when the server is running.

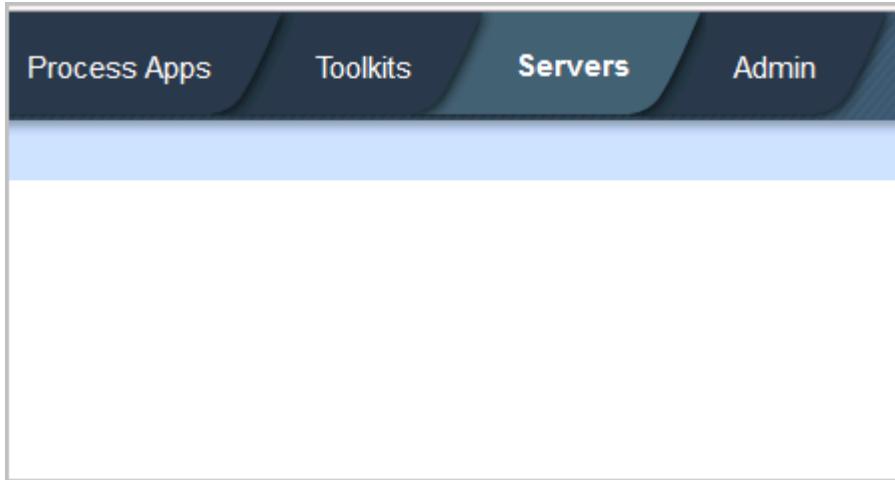
## **Part 2: Verify the status of the UTE Process Server**

In this section, you check whether the UTE Process Server is online and connected to Process Center.

- 1. Start the Process Center console.
  - a. On your desktop, double-click the **IBM BPM Process Center** shortcut.
  - b. Log in to the Process Center console by entering `pcdeadmin` in the **User name** field and `web1sphere` in the **Password** field.
  - c. Click **Log In**.



- \_\_\_ d. If prompted, close the “Getting Started with IBM Process Center 8.5.7.0” window.
- \_\_\_ 2. Check the status of the online UTE Process Server.
  - \_\_\_ a. Click the **Servers** tab.
  - \_\_\_ b. Verify that no server is listed in the Servers tab, which means that the UTE Process Server is not connected (offline or online) with the Process Center.



- \_\_\_ c. Minimize the browser. Do not log out of the Process Center console because you are going to check the status again later.

The UTE Process Server must be connected for you to install and deploy snapshots from the Process Center console. You bring the UTE server online in the remaining portion of this exercise.

### ***Part 3: Examine the connection properties***

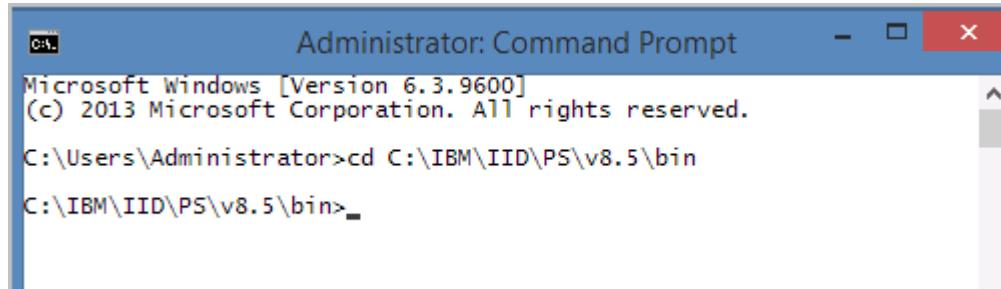
In this section, you use the wsadmin scripts to create and update some of the key properties of the UTE Process Server that are required for the connection.

- \_\_\_ 1. Open a command prompt window, and change the directory to C:\IBM\IID\PS\v8.5\bin.
  - \_\_\_ a. Double-click the command prompt icon in the taskbar.



- \_\_\_ b. Type the following command and press the Enter key to change the directory to the Process Server deployment manager bin folder:

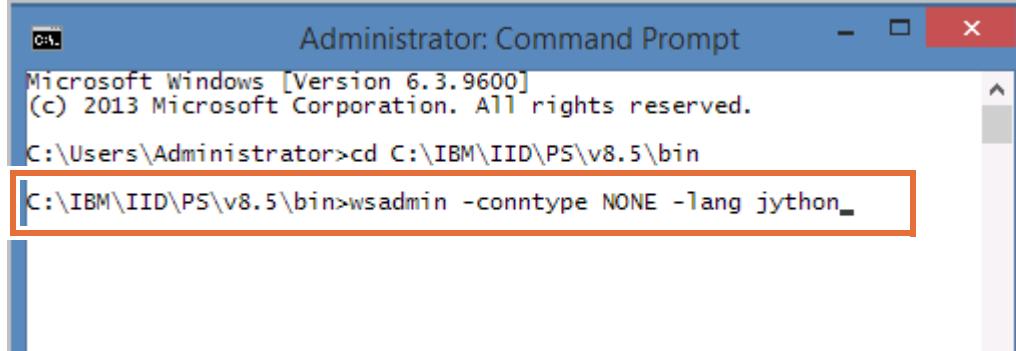
```
cd C:\IBM\IID\PS\v8.5\bin
```



The screenshot shows an 'Administrator: Command Prompt' window. The title bar says 'Administrator: Command Prompt'. The window content shows the following text:  
 Microsoft Windows [Version 6.3.9600]  
 (c) 2013 Microsoft Corporation. All rights reserved.  
 C:\Users\Administrator>cd C:\IBM\IID\PS\v8.5\bin  
 C:\IBM\IID\PS\v8.5\bin>

- \_\_\_ 2. Start the wsadmin tool so that it runs in the local mode and is not connected to any running servers, and so it uses Jython commands.
- \_\_\_ a. Enter the following command and press the Enter key:

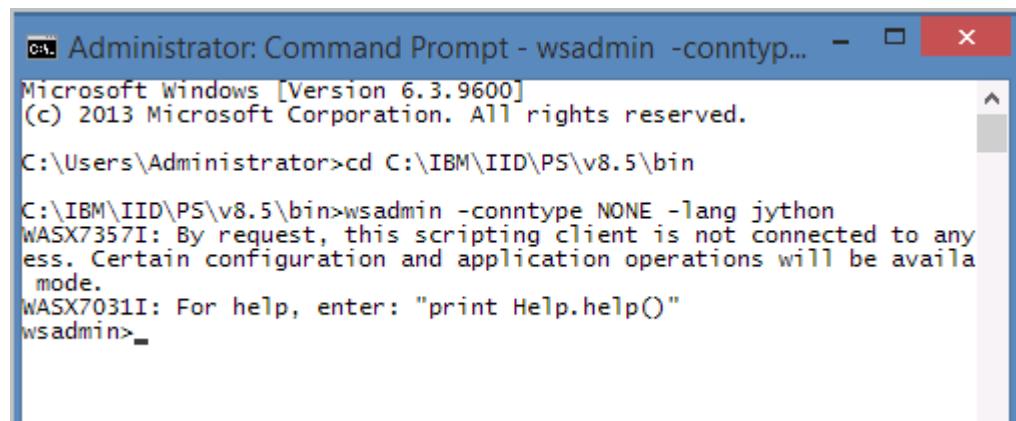
```
wsadmin -conntype NONE -lang jython
```



The screenshot shows an 'Administrator: Command Prompt' window. The title bar says 'Administrator: Command Prompt'. The window content shows the following text:  
 Microsoft Windows [Version 6.3.9600]  
 (c) 2013 Microsoft Corporation. All rights reserved.  
 C:\Users\Administrator>cd C:\IBM\IID\PS\v8.5\bin  
 C:\IBM\IID\PS\v8.5\bin>wsadmin -conntype NONE -lang jython

The wsadmin tool starts, and you see messages in the command prompt window.

- \_\_\_ b. Wait for the start process to complete and you see the wsadmin> prompt.



The screenshot shows an 'Administrator: Command Prompt - wsadmin -conntyp...' window. The title bar says 'Administrator: Command Prompt - wsadmin -conntyp...'. The window content shows the following text:  
 Microsoft Windows [Version 6.3.9600]  
 (c) 2013 Microsoft Corporation. All rights reserved.  
 C:\Users\Administrator>cd C:\IBM\IID\PS\v8.5\bin  
 C:\IBM\IID\PS\v8.5\bin>wsadmin -conntype NONE -lang jython  
 WASX7357I: By request, this scripting client is not connected to any  
 ess. Certain configuration and application operations will be availa  
 mode.  
 WASX7031I: For help, enter: "print Help.help()  
 wsadmin>



## Note

For more information about wsadmin, see the product documentation at [http://www.ibm.com/support/knowledgecenter/SSFPJS\\_8.5.7/com.ibm.wbpm.main.doc/kc-homepage-bpm.html](http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbpm.main.doc/kc-homepage-bpm.html) and search for wsadmin.

- \_\_\_ 3. Display the list of UTE Process Server configuration properties.

- \_\_\_ a. Type the following command, and press Enter:

```
ps=AdminConfig.getId("/BPMProcessServer:/")
```

```
Administrator: Command Prompt - wsadmin -conntyp...
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\IBM\IID\PS\v8.5\bin
C:\IBM\IID\PS\v8.5\bin>wsadmin -conntype NONE -lang jython
WASX7357I: By request, this scripting client is not connected to any
ess. Certain configuration and application operations will be availa
mode.

wsadmin>ps=AdminConfig.getId("/BPMProcessServer:/")
```

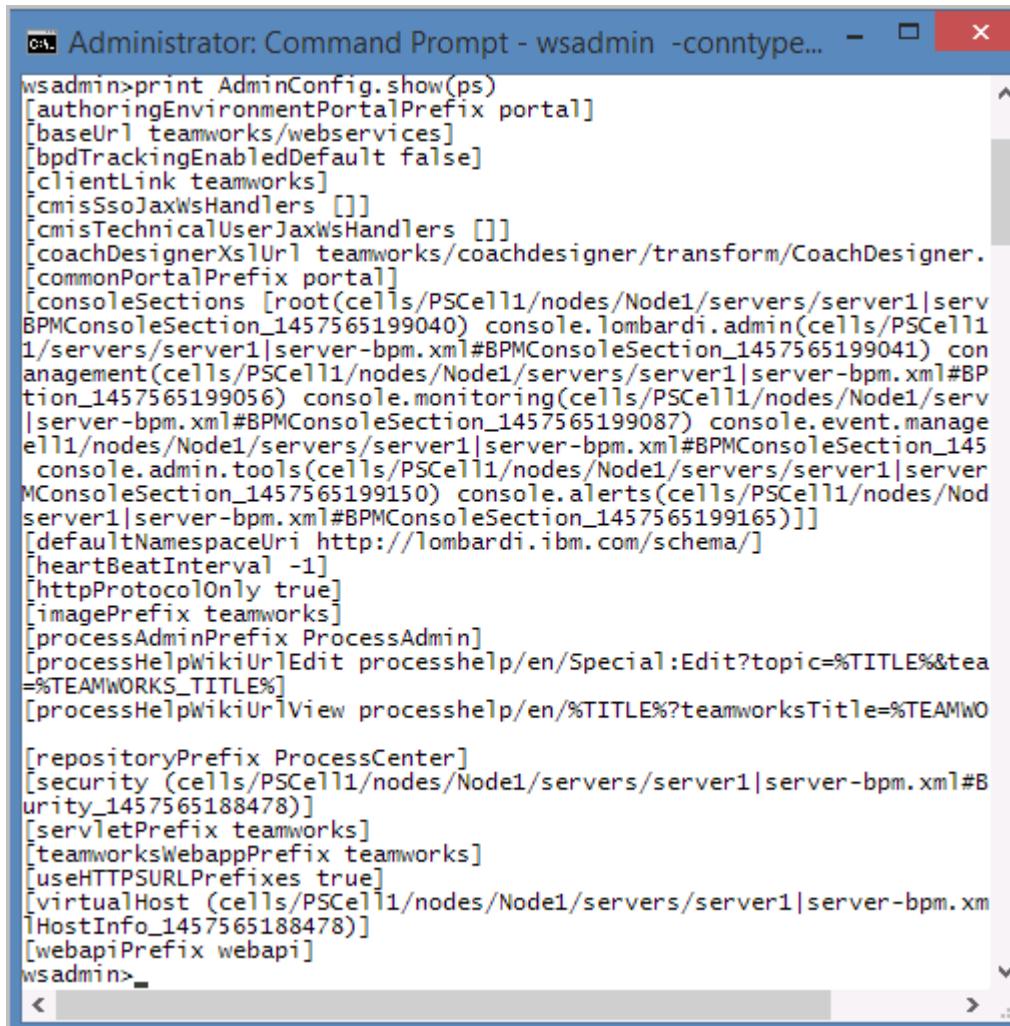
- \_\_\_ b. When you see the wsadmin > prompt, type the following command and press Enter:

```
print AdminConfig.show(ps)
```

```
Administrator: Command Prompt - wsadmin -conntyp...
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\IBM\IID\PS\v8.5\bin
C:\IBM\IID\PS\v8.5\bin>wsadmin -conntype NONE -lang jython
WASX7357I: By request, this scripting client is not connected to any
ess. Certain configuration and application operations will be availa
mode.
WASX7031I: For help, enter: "print Help.help()"
wsadmin>print AdminConfig.show(ps)
```

- \_\_\_ c. Review the properties that are listed.



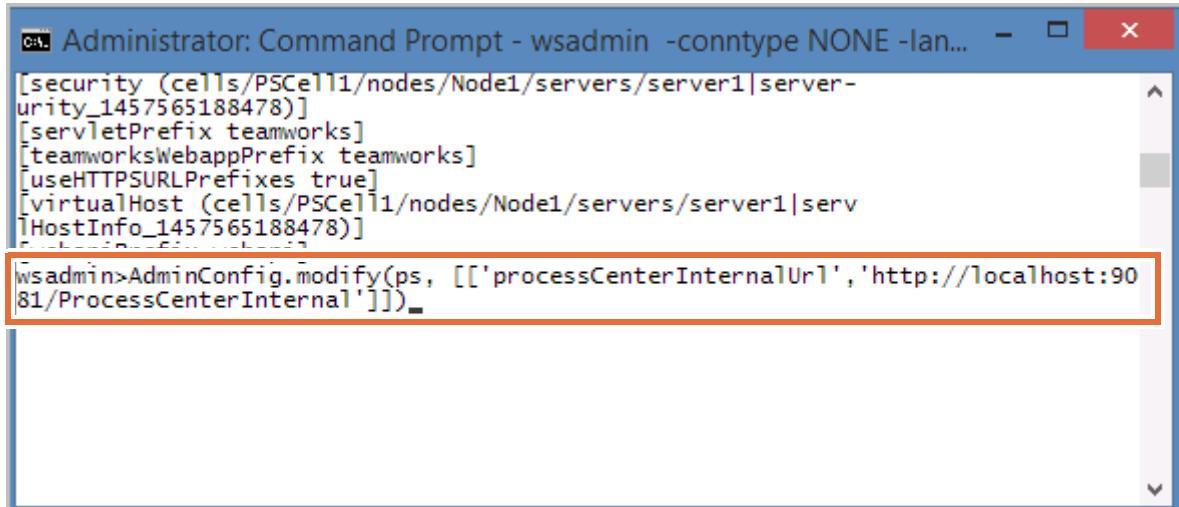
The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - wsadmin -conntype...". The window displays the output of the command "wsadmin>print AdminConfig.show(ps)". The output is a large block of text representing a Java-style configuration object. It includes sections for "authoringEnvironmentPortal", "baseUrl\_teamworks/webservices", "bpdTrackingEnabledDefault", "clientLink\_teamworks", "cmisSsoJaxWsHandlers", "cmisTechnicalUserJaxWsHandlers", "coachDesignerXslUrl", "commonPortalPrefix", "consoleSections", "consoleMonitoring", "consoleEventManagement", "consoleAdminTools", "consoleAlerts", "defaultNamespaceUri", "heartBeatInterval", "httpProtocolOnly", "imagePrefix", "processAdminPrefix", "processHelpWikiUrlEdit", "processHelpWikiUrlView", "repositoryPrefix", "security", "servletPrefix", "teamworksWebappPrefix", "useHTTPSURLPrefixes", "virtualHost", and "webapiPrefix". The text is color-coded for readability, with various parts in blue, purple, and red.

```
wsadmin>print AdminConfig.show(ps)
[authoringEnvironmentPortal]prefix portal]
[baseUrl_teamworks/webservices]
[bpdTrackingEnabledDefault false]
[clientLink_teamworks]
[cmisSsoJaxWsHandlers []]
[cmisTechnicalUserJaxWsHandlers []]
[coachDesignerXslUrl teamworks/coachdesigner/transform/CoachDesigner.
[commonPortalPrefix portal]
[consoleSections [root(cells/PSCell11/nodes/Node1/servers/server1|serv
BPMConsoleSection_1457565199040) console.lombardi.admin(cells/PSCell11
1/servers/server1|server-bpm.xml#BPMConsoleSection_1457565199041) con
anagement(cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml#BP
tion_1457565199056) console.monitoring(cells/PSCell11/nodes/Node1/serv
|server-bpm.xml#BPMConsoleSection_1457565199087) console.event.manage
ell1/nodes/Node1/servers/server1|server-bpm.xml#BPMConsoleSection_145
console.admin.tools(cells/PSCell11/nodes/Node1/servers/server1|server
MConsoleSection_1457565199150) console.alerts(cells/PSCell11/nodes/Nod
server1|server-bpm.xml#BPMConsoleSection_1457565199165)]]
[defaultNamespaceUri http://lombardi.ibm.com/schema/]
[heartBeatInterval -1]
[httpProtocolOnly true]
[imagePrefix teamworks]
[processAdminPrefix ProcessAdmin]
[processHelpWikiUrlEdit processhelp/en/Special>Edit?topic=%TITLE%&tea
=%TEAMWORKS_TITLE%]
[processHelpWikiUrlView processhelp/en/%TITLE%?teamworksTitle=%TEAMWO
[repositoryPrefix ProcessCenter]
[security (cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml#B
urity_1457565188478)]
[servletPrefix teamworks]
[teamworksWebappPrefix teamworks]
[useHTTPSURLPrefixes true]
[virtualHost (cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml
HostInfo_1457565188478)]
[webapiPrefix webapi]
wsadmin>
```

4. Update the ProcessCenterInternalUrl property.

a. Type the following command, and press Enter:

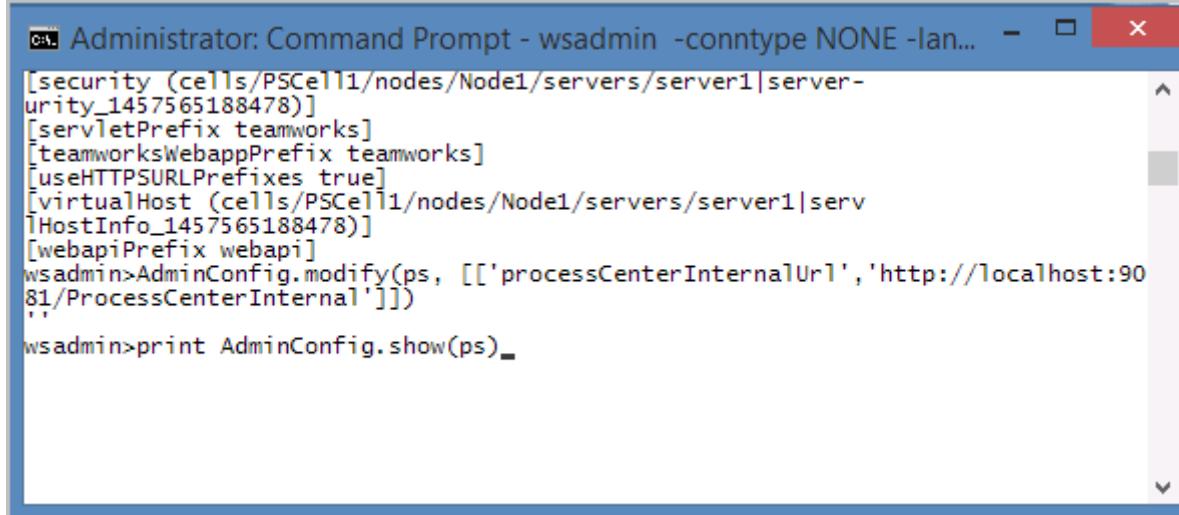
```
AdminConfig.modify(ps,
 [['processCenterInternalUrl','http://localhost:9081/ProcessCenterInternal
 ']])
```



The screenshot shows an Administrator Command Prompt window titled "Administrator: Command Prompt - wsadmin -conntype NONE -lan...". The command entered is "AdminConfig.modify(ps, [['processCenterInternalUrl','http://localhost:9081/ProcessCenterInternal']])" which is highlighted with a red box. The output shows various server configuration details like security, servletPrefix, and virtualHost settings.

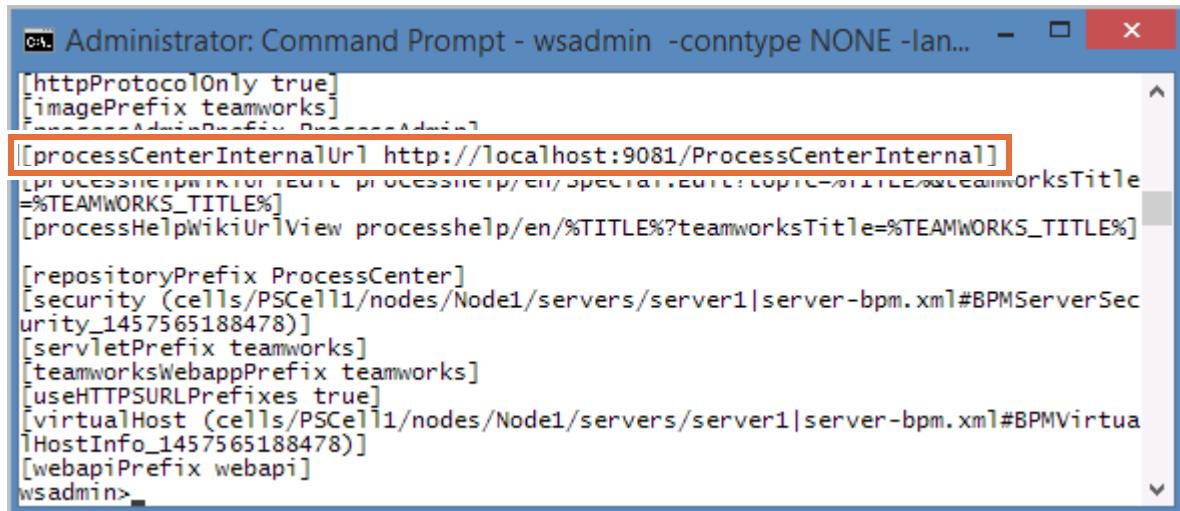
b. When you see the wsadmin > prompt, type the following command and press Enter:

```
print AdminConfig.show(ps)
```



The screenshot shows an Administrator Command Prompt window titled "Administrator: Command Prompt - wsadmin -conntype NONE -lan...". The command entered is "print AdminConfig.show(ps)" which is highlighted with a red box. The output shows the updated configuration including the modified processCenterInternalUrl setting.

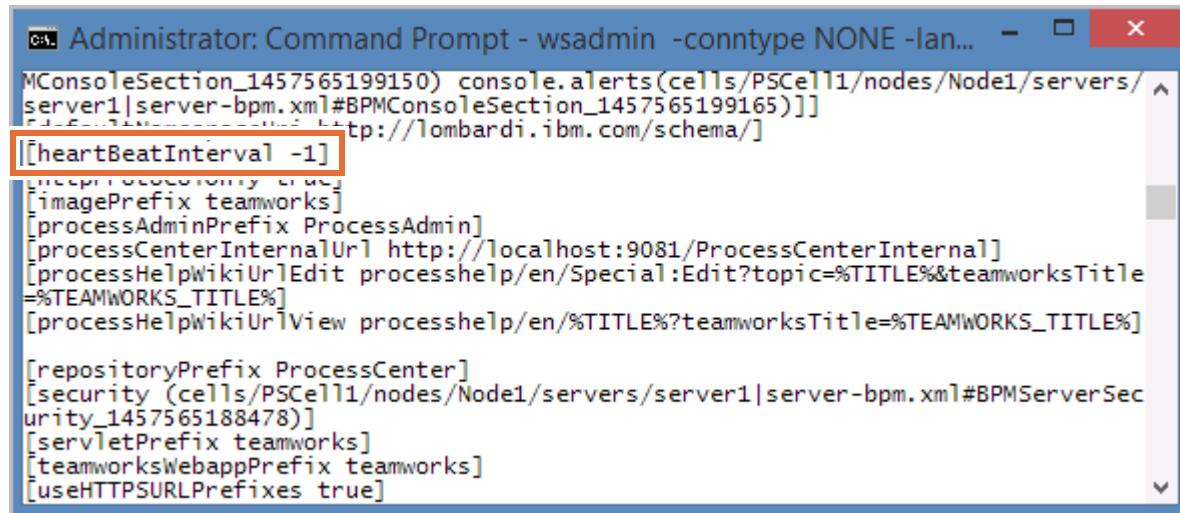
- \_\_\_ c. Verify that the `processCenterInternalUrl` property is added.



```
[httpProtocolOnly true]
[imagePrefix teamworks]
[processCenterInternalUrl http://localhost:9081/ProcessCenterInternal]
[processAdminPrefix ProcessAdmin]
[processHelpWikiUrlEdit processhelp/en/Special>Edit?topic=%TITLE%&teamworksTitle=%TEAMWORKS_TITLE%]
[processHelpWikiUrlView processhelp/en/%TITLE%?teamworksTitle=%TEAMWORKS_TITLE%]

[repositoryPrefix ProcessCenter]
[security (cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml#BPMServerSecurity_1457565188478)]
[servletPrefix teamworks]
[teamworksWebappPrefix teamworks]
[useHTTPSURLPrefixes true]
[virtualHost (cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml#BPMVirtualHostInfo_1457565188478)]
[webapiPrefix webapi]
wsadmin>
```

- \_\_\_ d. Find the `heartBeatInterval` property.



```
MConsoleSection_1457565199150) console.alerts(cells/PSCell11/nodes/Node1/servers/
server1|server-bpm.xml#BPMConsoleSection_1457565199165)]]
http://lombardi.ibm.com/schema/]
[[heartBeatInterval -1]]
[imagePrefix teamworks]
[processAdminPrefix ProcessAdmin]
[processCenterInternalUrl http://localhost:9081/ProcessCenterInternal]
[processHelpWikiUrlEdit processhelp/en/Special>Edit?topic=%TITLE%&teamworksTitle=%TEAMWORKS_TITLE%]
[processHelpWikiUrlView processhelp/en/%TITLE%?teamworksTitle=%TEAMWORKS_TITLE%]

[repositoryPrefix ProcessCenter]
[security (cells/PSCell11/nodes/Node1/servers/server1|server-bpm.xml#BPMServerSecurity_1457565188478)]
[servletPrefix teamworks]
[teamworksWebappPrefix teamworks]
[useHTTPSURLPrefixes true]
```

The `heartBeatInterval` value is listed at `-1`. The heartbeat interval is the polling interval, in seconds, that is used by the UTE Process Server to communicate its location and characteristics to the Process Center. To change the state of an offline UTE Process Server to online, you need to update the `heartBeatInterval` value to a number that is larger than `0` (zero).

- \_\_\_ e. Type the following command, and press Enter:

```
AdminConfig.modify(ps, [['heartBeatInterval','30']])
```

- \_\_\_ f. When you see the `wsadmin >` prompt, type the following command and press Enter:

```
print AdminConfig.show(ps)
```

- \_\_ g. Verify that the heartBeatInterval property is updated to 30.

```

Administrator: Command Prompt - wsadmin -conntype NONE -lan...
server1|server-bpm.xml#BPMConsoleSection_1457565199165)]]
[http://lombardi.ibm.com/schema/]
[heartBeatInterval 30]
[imagePrefix teamworks]
[processAdminPrefix ProcessAdmin]
[processCenterInternalUrl http://localhost:9081/ProcessCenterInternal]
[processHelpWikiUrlEdit processhelp/en/Special>Edit?topic=%TITLE%&teamworksTitle=%TEAMWORKS_TITLE%]
[processHelpWikiUrlView processhelp/en/%TITLE%?teamworksTitle=%TEAMWORKS_TITLE%]

[repositoryPrefix ProcessCenter]
[security (cells/PSCell1/nodes/Node1/servers/server1|server-bpm.xml#BPMServerSecurity_1457565188478)]
[servletPrefix teamworks]
[teamworksWebappPrefix teamworks]
[useHTTPSURLPrefixes true]
[virtualHost (cells/PSCell1/nodes/Node1/servers/server1|server-bpm.xml#BPMVirtualHost_1457565188478)]

```

- \_\_ h. Type AdminConfig.save() and press Enter. Your changes are saved.

- \_\_ i. Exit wsadmin by entering the following command and pressing Enter:

exit

- \_\_ j. Close the command prompt window.

For the wsadmin changes to take effect, the UTE Process Server must be restarted. However, you need to make some additional changes before you restart the server, which is done in the next section.

#### **Part 4: Define the authentication alias in Process Server**

- \_\_ 1. Log in to the UTE Process Server administrative console.

- \_\_ a. Open a Firefox web browser window and go to the following URL:  
`http://localhost:9060/ibm/console`
- \_\_ b. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.
- \_\_ c. On the next screen, accept the defaults and click **Confirm Security Exception**.
- \_\_ d. In the WebSphere Integrated Solutions Console login window, enter the following credentials and click **Log in**.

**User ID:** admin

**Password:** web1sphere

- \_\_ e. In the administration console, expand **Security** and click **Global security**.

- \_\_ f. In the Authentication section, expand **Java Authentication and Authorization Service**.

The screenshot shows the 'Authentication' configuration page. At the top, there's a heading 'Authentication mechanisms and expiration'. Below it is a list of options:

- [LTPA](#)
- [Kerberos and LTPA](#)  
[Kerberos configuration](#)
- [SWAM \(deprecated\): No authenticated communication between servers](#)  
[Authentication cache settings](#)

Below these are sections for 'Web and SIP security', 'RMI/IOP security', and 'Java Authentication and Authorization Service'. The 'Java Authentication and Authorization Service' section is highlighted with a red box. Underneath it are three checkboxes:

- [Enable Java Authentication SPI \(JASPI\)](#)  
[Providers](#)
- [Use realm-qualified user names](#)

At the bottom, there are four links:

- [Security domains](#)
- [External authorization providers](#)
- [Programmatic session cookie configuration](#)
- [Custom properties](#)

- \_\_ g. Click **J2c authentication data**.  
\_\_ h. Verify that the option for **Prefix new alias names with the node name of the cell** is not selected.

The screenshot shows the 'Global security > JAAS - J2C authentication data' page. It has a blue header bar with the title. Below it is a descriptive text: 'Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.' Underneath is a checkbox:

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

At the bottom is a 'Apply' button.

- \_\_ 2. Create an authentication alias for the Process Center user.  
\_\_ a. In the **J2c authentication data** page, click **New**.

- \_\_ b. Enter the following information:

- **Alias:** ProcessCenterUserAlias
- **User ID:** pcdeadmin
- **Password:** web1sphere

**Global security**

[Global security > JAAS - J2C authentication data > New...](#)

Specifies a list of user identities and passwords for Java(TM) 2 Platform, Enterprise Edition (J2EE) applications.

**General Properties**

|             |                        |
|-------------|------------------------|
| * Alias     | ProcessCenterUserAlias |
| * User ID   | pcdeadmin              |
| * Password  | *****                  |
| Description |                        |

Apply    OK    Reset    Cancel

- \_\_ c. Click **OK**.

- \_\_ d. Click **Save**.



#### Information

ProcessCenterUserAlias must use a valid user name and password from the Process Center environment. The user does not need any special authorization in Process Center.

- \_\_ e. Expand **Servers** and click **Deployment Environments**.

- \_\_ f. Click the **ProcessServer** link.

| Deployment Environments                               |                             |
|-------------------------------------------------------|-----------------------------|
| Select the deployment environments to manage. You can |                             |
|                                                       |                             |
| Status                                                | Deployment Environment Name |
|                                                       | ProcessServer               |
| Total                                                 | 1                           |

- \_\_ g. In the **Related Items** section, click **Authentication Aliases**.
- \_\_ h. Examine the roles and the aliases that are associated with them.

The screenshot shows a software interface titled "Authentication Alias". On the left, there is a list of "Role"s, and on the right, there is a list of "Alias" entries. Both lists have a red border around them. The "Role" list contains 15 items: BPCUser, BPMAdminJobUser, BPMAuthor, BPMUser, BPMWebserviceUser, DeAdmin, EmbeddedECMTechicalUser, EventManagerUser, MMAadmin, PerformanceDWUser, ProcessCenterUser, ProcessServerUser, RALUser, SCADeploymentUser, and SCAUser. The "Alias" list contains 15 entries, each corresponding to a role and set to "DeAdminAlias": DeAdminAlias, and DeAdminAlias. At the bottom of the interface, there is a message: "You can specify multiple aliases for role SystemLaneUser. Hold the CTRL key to select." A "Total 15" label is also present at the bottom.

- \_\_ i. Notice the alias for the `ProcessCenterUser` role. It is listed as `DeAdminAlias`.  
The `ProcessCenterUserAlias` that exists in the environment for access to the Process Center is not associated anywhere here with any role, specifically the `ProcessCenterUser` role. You make that association in the next step.

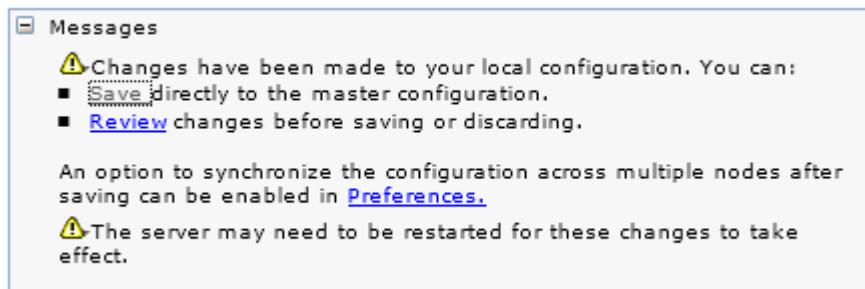
3. Associate the alias with the ProcessCenterUser role.
- a. Change the value of the **Alias** for ProcessCenterUser from DeAdminAlias to: ProcessCenterUserAlias



- b. Verify that the modified value is listed for the alias. Click **OK**.

| Authentication Alias    |                        |
|-------------------------|------------------------|
| Role                    | Alias                  |
| BPCUser                 | DeAdminAlias           |
| BPMAdminJobUser         | DeAdminAlias           |
| BPMAuthor               | DeAdminAlias           |
| BPMUser                 | DeAdminAlias           |
| BPMWebserviceUser       | DeAdminAlias           |
| DeAdmin                 | DeAdminAlias           |
| EmbeddedECMTechicalUser | DeAdminAlias           |
| EventManagerUser        | DeAdminAlias           |
| MMAadmin                | DeAdminAlias           |
| PerformanceDWUser       | DeAdminAlias           |
| ProcessCenterUser       | ProcessCenterUserAlias |
| ProcessServerUser       | DeAdminAlias           |
| RALUser                 | DeAdminAlias           |
| SCADeploymentUser       | DeAdminAlias           |
| SCAUser                 | DeAdminAlias           |
| Total 15                |                        |

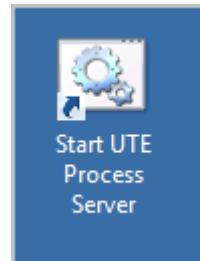
- \_\_\_ c. Click **Save**.



- \_\_\_ d. Log out of the browser and close it.
- \_\_\_ 4. Restart the UTE Process Server so that the new configuration is in effect.
- \_\_\_ a. On your desktop, double-click **Stop UTE Process Server**.



- \_\_\_ b. A DOS command window is displayed, and the UTE Process Center server instance stops. The command window closes when the server stops.
- \_\_\_ c. On your desktop, double-click **Start UTE Process Server**.



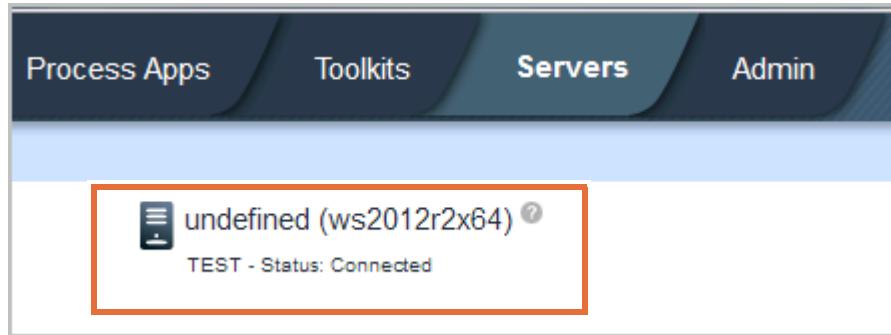
- \_\_\_ d. A DOS command window is displayed, and the UTE Process Center server instance starts. The command window closes when the server is running.

## **Part 5: Verify the status of the online Process Server**

Now that the modifications are done, you examine the status of the online server.

- \_\_\_ 1. Examine the Process Center Console and verify the status of the online Process Server.
- \_\_\_ a. Go back to the Process Center console that you minimized earlier. If you are logged out, then log back in by entering `pcdeadmin` in the **User name** field and `web1sphere` in the **Password** field.
- \_\_\_ b. Click the **Servers** tab.

- \_\_\_ c. Notice that the status of the online Process Server is now connected.



- \_\_\_ 2. Because the online UTE server is listed as undefined, you might want to modify the name to something more specific.
- \_\_\_ a. Open Windows Explorer and go to:  
C:\IBM\IID\PS\v8.5\profiles\qbpmaps\config\cells\PSCell1\nodes\Node1\servers\server1\process-server\config
  - \_\_\_ b. Right-click **100Custom.xml** and click **Copy**.
  - \_\_\_ c. Right-click any empty space in the folder and click **Paste**. A copy of the file is made in the directory.
  - \_\_\_ d. Right-click the Copy of the **100Custom.xml** file and click **Rename**.
  - \_\_\_ e. Rename the file to **100Custom.bak** and press Enter.
  - \_\_\_ f. Click **Yes** to change the file name extension.
  - \_\_\_ g. Right-click **100Custom.xml** and click **Open with > Notepad**.
  - \_\_\_ h. Scroll to the bottom of the **100Custom.xml** file to the last line. You see the **</properties>** tag.
  - \_\_\_ i. Place the cursor right before the **</properties>** tag and press Enter twice to create an empty space.

```
<keystore-type>JKS</keystore-type>
<certificate>C:\ibm\kf\rtc\deploy2\App
Also contains server1 cert
</webservice-security>
</server>
-->
</properties>
```

The image shows a Notepad window displaying XML code. The code includes tags for keystore-type, certificate, webservice-security, server, and properties. The properties tag is highlighted with an orange rectangular box.

- \_\_\_ j. Go to the **C:\labfiles\Support Files\Ex15** folder and open the **RenameOnlineUTEServer.txt** file in Notepad.

- \_\_\_ k. Copy the text from `RenameOnlineUTEserver.txt` and paste it in `100Custom.xml` in the newly created empty space right above the `</properties>` tag.

```

<server>
 <server-name merge="replace">UTEserver</server-name>
</server>
 <keystore-type>JKS</keystore-type>
 <certificate>C:\ibm\kf\rtc\deploy2\AppServer\profiles\Star
 Also contains server1 cert
 </webservice-security>
</server>
-->
<server>
<server-name merge="replace">UTEserver</server-name>
</server>
</properties>

```

- \_\_\_ l. Save the `100Custom.xml` file and close it.

The server must be restarted for the changes to take effect. However, you need to do one more task before starting the server. You need to exchange the security keys before you can deploy snapshots.

## **Part 6: Configure SSL communication between Process Center and test environment Process Server**

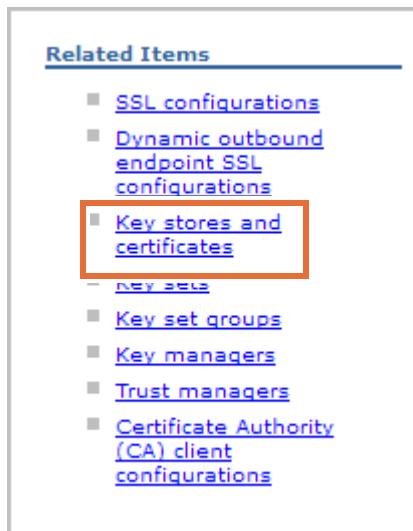
Now that the UTE Process Server is connected online, you must verify that the Process Server root signer SSL certificate is imported into Process Center because Process Center connects to Process Server through HTTPS by default. This step is needed to successfully deploy the process application from Process Center to the Process Server. In this section, you exchange the security keys.

- \_\_\_ 1. Import the Process Server root SSL certificate into Process Center.
  - \_\_\_ a. Double-click the **Deployment Manager Administrative Console** shortcut on the desktop.
  - \_\_\_ b. Log in by entering `bpmadmin` in the **User ID** field and `web1sphere` in the **Password** field.

- \_\_ c. In the left section, expand **Security** and click **SSL certificate and key management**.



- \_\_\_ d. In the **SSL certificate and key management** window, in the **Related Items** section, click **Key stores and certificates**.



- \_\_ e. Click **CellDefaultTrustStore**.

SSL certificate and key management

SSL certificate and key management > Key stores and certificates

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM),

Keystore usages

SSL keystores

[+] Preferences

New... Delete Change password... Exchange signers...

Select Name Description

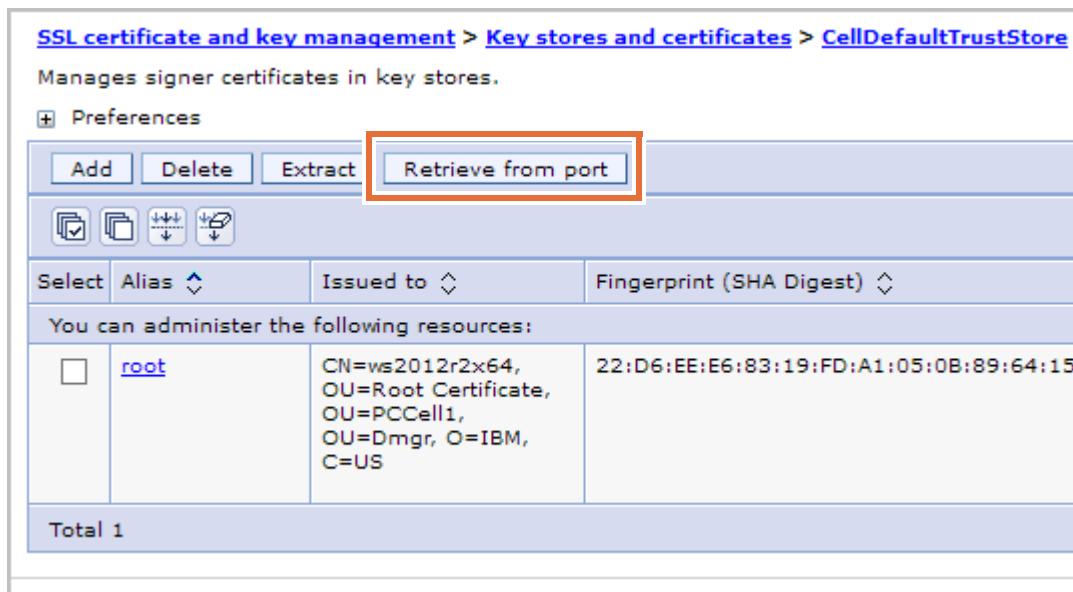
You can administer the following resources:

|                          |                                         |                                 |
|--------------------------|-----------------------------------------|---------------------------------|
| <input type="checkbox"/> | <a href="#">BlueWorksLiveTrustStore</a> | Trust store for BlueWorksLive   |
| <input type="checkbox"/> | <a href="#">CellDefaultKeyStore</a>     | Default key store for PCCell1   |
| <input type="checkbox"/> | <a href="#">CellDefaultTrustStore</a>   | Default trust store for PCCell1 |
| <input type="checkbox"/> | <a href="#">NodeDefaultKeyStore</a>     | Default key store for Node1     |
| <input type="checkbox"/> | <a href="#">NodeDefaultTrustStore</a>   | Default trust store for Node1   |

Total 5

- \_\_ f. In the **Additional Properties** section, click **Signer certificates**.

- \_\_ g. Click **Retrieve from port**.



- \_\_ h. Enter the following information in the **General Properties** section

- **Host:** localhost
- **Port:** 9443
- **Alias:** ProcessServerAlias

The screenshot shows a 'General Properties' dialog box. It has a title bar 'General Properties'. Inside, there are three input fields with asterisks: '\* Host' containing 'localhost', '\* Port' containing '9443', and '\* Alias' containing 'ProcessServerAlias'. These three fields are all enclosed within a large red rectangular box. Below these fields is a button 'Retrieve signer information'. At the bottom of the dialog are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

- \_\_ i. Click **Retrieve signer information**. Click **OK**.

**General Properties**

\* Host  
localhost

\* Port  
9443

SSL configuration for outbound connection  
CellDefaultSSLSettings ▾

\* Alias  
ProcessServerAlias

**Retrieve signer information**

**Retrieved signer information**

Serial number  
6573864856643

Issued to  
CN=ws2012r2x64, OU=Root Certificate, OU=PSCell1, OU=Node1, O=IBM, C=US

Issued by  
CN=ws2012r2x64, OU=Root Certificate, OU=PSCell1, OU=Node1, O=IBM, C=US

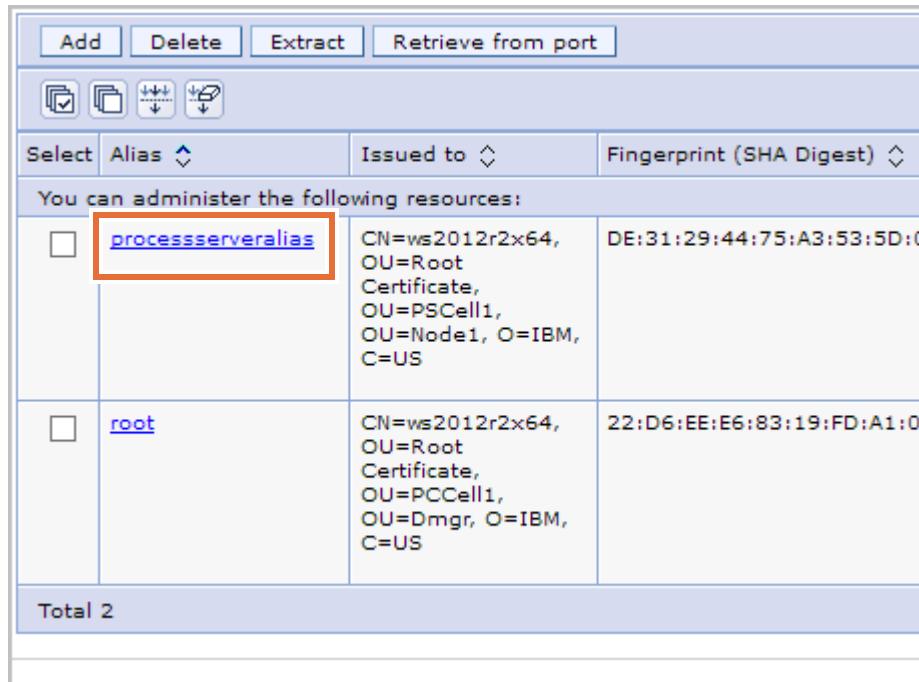
Fingerprint (SHA digest)  
DE:31:29:44:75:A3:53:5D:0F:0D:D2:B8:38:EB:D1:B3:7A:BE:53:9A

Validity period  
Mar 5, 2031

**Apply** **OK** **Reset** **Cancel**

- \_\_ j. Click **Save** to save directly to the master configuration.

- \_\_\_ k. Verify that the newly created certificate is listed.



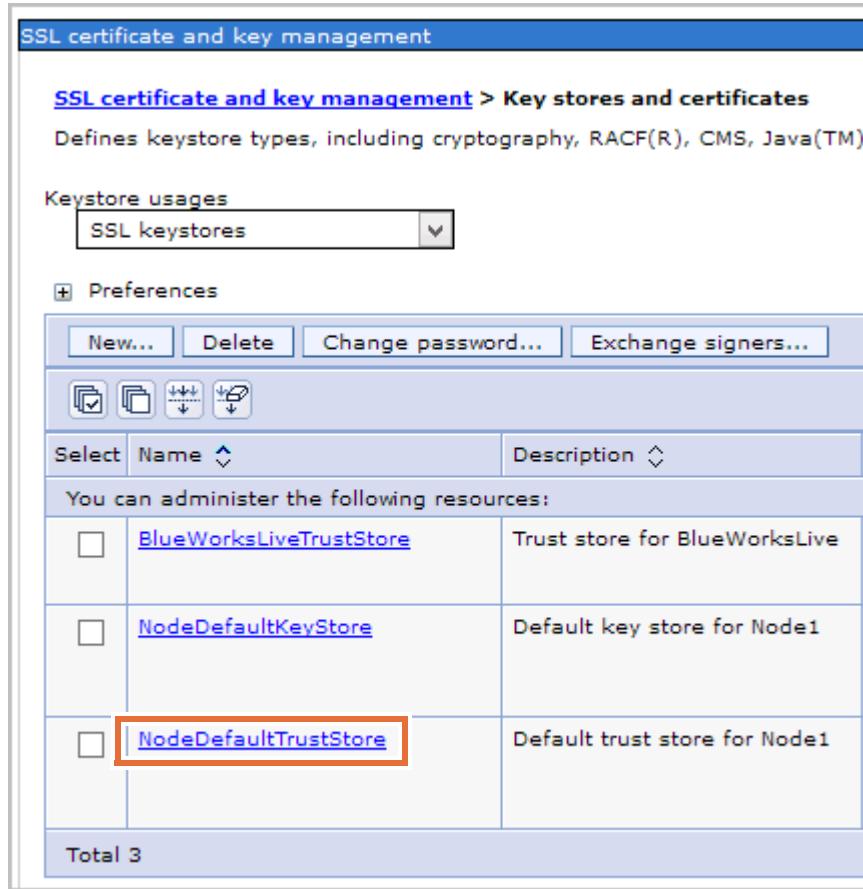
The screenshot shows a software interface for managing certificates. At the top, there are buttons for 'Add', 'Delete', 'Extract', and 'Retrieve from port'. Below these are icons for selecting, deleting, and extracting. A toolbar includes 'Select' and 'Alias' dropdowns, and 'Issued to' and 'Fingerprint (SHA Digest)' dropdowns. The main area displays a table titled 'You can administer the following resources:' with two rows:

|                          | Alias                              | Issued to                                                                             | Fingerprint (SHA Digest)  |
|--------------------------|------------------------------------|---------------------------------------------------------------------------------------|---------------------------|
| <input type="checkbox"/> | <a href="#">processserveralias</a> | CN=ws2012r2x64,<br>OU=Root<br>Certificate,<br>OU=PSCell1,<br>OU=Node1, O=IBM,<br>C=US | DE:31:29:44:75:A3:53:5D:0 |
| <input type="checkbox"/> | <a href="#">root</a>               | CN=ws2012r2x64,<br>OU=Root<br>Certificate,<br>OU=PCCell1,<br>OU=Dmgr, O=IBM,<br>C=US  | 22:D6:EE:E6:83:19:FD:A1:0 |

A total count of '2' is shown at the bottom left of the table area.

- \_\_\_ l. Log out of the administrative console and close the browser.
- \_\_\_ 2. Import the Process Center root SSL certificate into UTE Process Server.
- \_\_\_ a. Open a Firefox web browser window and go to the following URL:  
`http://localhost:9060/ibm/console`
- \_\_\_ b. Log in by entering **admin** in the **User ID** field and **web1sphere** in the **Password** field.
- \_\_\_ c. In the left section, expand **Security** and click **SSL certificate and key management**.
- \_\_\_ d. In the **SSL certificate and key management** window, in the **Related Items** section, click **Key stores and certificates**.

- \_\_ e. Click **NodeDefaultTrustStore**.



- \_\_ f. In the **Additional Properties** section, click **Signer certificates**.  
 \_\_ g. Click **Retrieve from port**.  
 \_\_ h. Enter the following information in the **General Properties** section

- **Host:** localhost
- **Port:** 9444
- **Alias:** ProcessCenterAlias

**General Properties**

|                                                                     |
|---------------------------------------------------------------------|
| * Host<br>localhost                                                 |
| * Port<br>9444                                                      |
| SSL configuration for outbound connection<br>NodeDefaultSSLSettings |
| * Alias<br>ProcessCenterAlias                                       |
| <a href="#">Retrieve signer information</a>                         |

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

- \_\_ i. Click **Retrieve signer information**. Click **OK**.

The screenshot shows the 'General Properties' dialog box. At the top, there are fields for 'Host' (localhost) and 'Port' (9444). Below these is a dropdown for 'SSL configuration for outbound connection' set to 'NodeDefaultSSLSettings'. A section titled 'Retrieved signer information' displays the following details:

|                          |                                                                       |
|--------------------------|-----------------------------------------------------------------------|
| Serial number            | 62845550811740                                                        |
| Issued to                | CN=ws2012r2x64, OU=Root Certificate, OU=PCCell1, OU=Dmgr, O=IBM, C=US |
| Issued by                | CN=ws2012r2x64, OU=Root Certificate, OU=PCCell1, OU=Dmgr, O=IBM, C=US |
| Fingerprint (SHA digest) | 22:D6:EE:E6:83:19:FD:A1:05:0B:89:64:15:46:D3:4F:29:46:F5:6F           |
| Validity period          | Mar 6, 2031                                                           |

At the bottom of the dialog are buttons for **Apply**, **OK**, **Reset**, and **Cancel**.

- \_\_ j. Click **Save** to save directly to the master configuration.

- \_\_ k. Verify that the newly created certificate is listed.

| Select                   | Alias                              | Issued to                                                                             | Fingerprint (SHA Digest) |
|--------------------------|------------------------------------|---------------------------------------------------------------------------------------|--------------------------|
| <input type="checkbox"/> | <a href="#">processcenteralias</a> | CN=ws2012r2x64,<br>OU=Root<br>Certificate,<br>OU=PCCell1,<br>OU=Dmgr, O=IBM,<br>C=US  | 22:D6:EE:E6:83:19:FD:A1: |
| <input type="checkbox"/> | <a href="#">root</a>               | CN=ws2012r2x64,<br>OU=Root<br>Certificate,<br>OU=PSCell1,<br>OU=Node1, O=IBM,<br>C=US | DE:31:29:44:75:A3:53:5D  |
| Total 2                  |                                    |                                                                                       |                          |

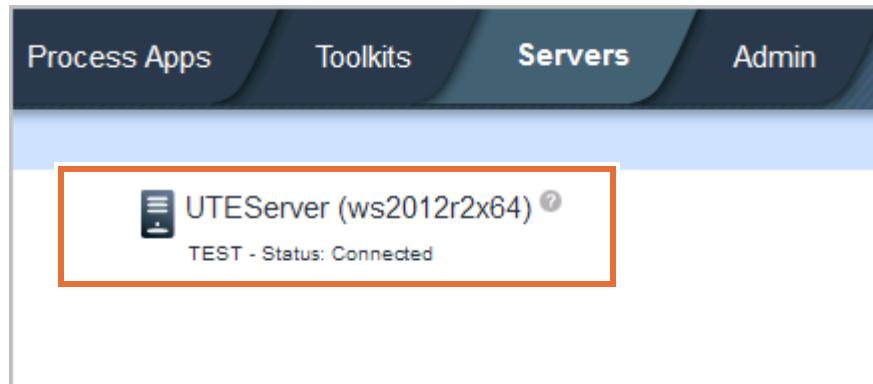
- \_\_ l. Log out of the administrative console and close the browser.

The environment configuration is now complete. You can now deploy snapshots successfully to an online UTE Process Server. Do not stop the servers as you need them running for the next exercise.

## Part 7: Restart the server environment and verify the UTE server status and name change

- \_\_ 1. Stop the IBM Process Center Cell Environment.
  - \_\_ a. Double-click the **Stop Process Center cluster** icon. Wait until it stops successfully. Press any key to close the command window.
  - \_\_ b. Double-click the **Stop Process Center node agent** icon. Wait until it stops successfully. Press any key to close the command window.
  - \_\_ c. Double-click the **Stop Process Center deployment manager** icon. Wait until it stops successfully. Press any key to close the command window.
- \_\_ 2. Stop the UTE server.
  - \_\_ a. Double-click the desktop shortcut labeled **Stop UTE Process Server**. A command window opens and it closes when the UTE server stops.
- \_\_ 3. Start the IBM Process Center Cell Environment.
  - \_\_ a. Double-click the **Start Process Center deployment manager** icon. Wait until it starts successfully. Press any key to close the command window.

- \_\_\_ b. Double-click the **Start Process Center node agent** icon. Wait until it starts successfully.  
Press any key to close the command window.
  - \_\_\_ c. Double-click the **Start Process Center cluster** icon. Wait until it starts successfully.  
Press any key to close the command window.
- \_\_\_ 4. Start the UTE server.
- \_\_\_ a. Double-click the desktop shortcut labeled **Start UTE Process Server**. A command window opens and it closes when the UTE server starts.
- \_\_\_ 5. Start Process Center console.
- \_\_\_ a. Open a web browser, and enter the following web address:  
`http://localhost:9081/ProcessCenter`
  - \_\_\_ b. Log in to the Process Center console by entering `pcdeadmin` in the **User name** field and `websphere` in the **Password** field.
  - \_\_\_ c. Click **Login**.
- \_\_\_ 6. Check the status of the online UTE Process Server.
- \_\_\_ a. Click the **Servers** tab.
  - \_\_\_ b. Verify that `UTEServer` is listed in the **Servers** tab. This step confirms that the environment is configured successfully.



If you do not see the server listed online, then go back and check the earlier steps, and also check whether the `100Custom.xml` file was edited correctly.

## End of exercise

## Exercise review and wrap-up

In this exercise, you brought the UTE Process Server online and ready for deployment.

# Exercise 16.Exploring IBM Process Center

## What this exercise is about

In this exercise, you use the IBM Process Center repository to manage your applications.

## What you should be able to do

After completing this exercise, you should be able to:

- Explore the IBM Process Center repository
- Create a toolkit in IBM Process Center
- Associate IBM Integration Designer artifacts with the toolkit
- Generate access to process applications and toolkits
- Archive and delete process applications

## Introduction

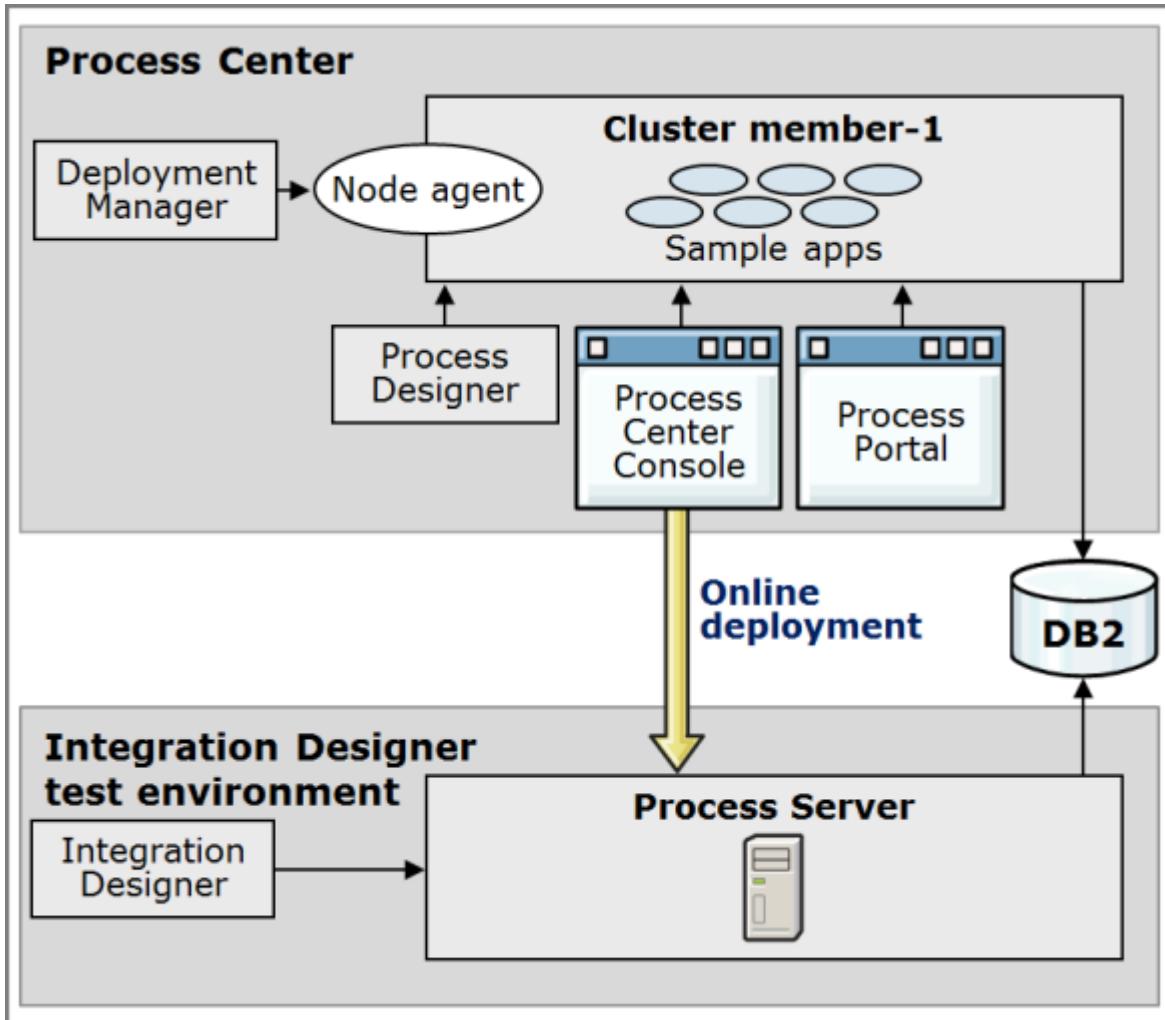
IBM Process Center is a runtime environment where IBM Process Designer and IBM Integration Designer share assets. It allows the development of business processes cooperatively in a highly interactive manner. IBM Process Center includes a server and performance data warehouse. IT employees who work in the authoring environments run processes and store performance data for testing and playback purposes.

The IBM Process Center console provides a convenient location in which to create and maintain high-level containers such as process applications and toolkits. Administrators can use the IBM Process Center console to provide a framework in which BPM analysts and developers can build their processes and underlying implementations. Another primary task for administrators is managing access to the IBM Process Center repository by setting up the appropriate authorization for users and groups.

From the Process Center console, you can:

- Create process applications and toolkits and grant other users access to those process applications and toolkits
- Create process models, services, and other assets within process applications
- Install process applications that are ready for testing or production on the process servers in those environments
- Manage running instances of process applications in configured environments

Process Center is installed and configured as a single cluster topology, whereas Process Server is installed as a test server (as a stand-alone single server) while installing integration designer. The following diagram depicts different processes that you start and their relationship with Process Center and Process server Cells.



## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and IBM Process Server test environment. Instructions for creating the lab environment are included in the exercise appendixes.

Successful completion of Exercise 15, Bringing the UTE Process Server online, is a prerequisite for this exercise.

## Exercise instructions

### **Part 1: Explore IBM Process Center repository**

One way of accessing the IBM Process Center Console is through a perspective within the IBM Integration Designer.

If the Process Center environment is running from the previous exercise, then go to step 5 to open the Exercise 16 workspace. Otherwise, complete the following steps to start the IBM Process Center server:

- 1. On your Windows desktop in your lab environment, select the **Start Process Center deployment manager** shortcut. Double-click the shortcut or press Enter to start the server.



- a. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.
- 2. On your Windows desktop, select the **Start Process Center node agent** shortcut. Double-click the shortcut or press Enter to start the server.

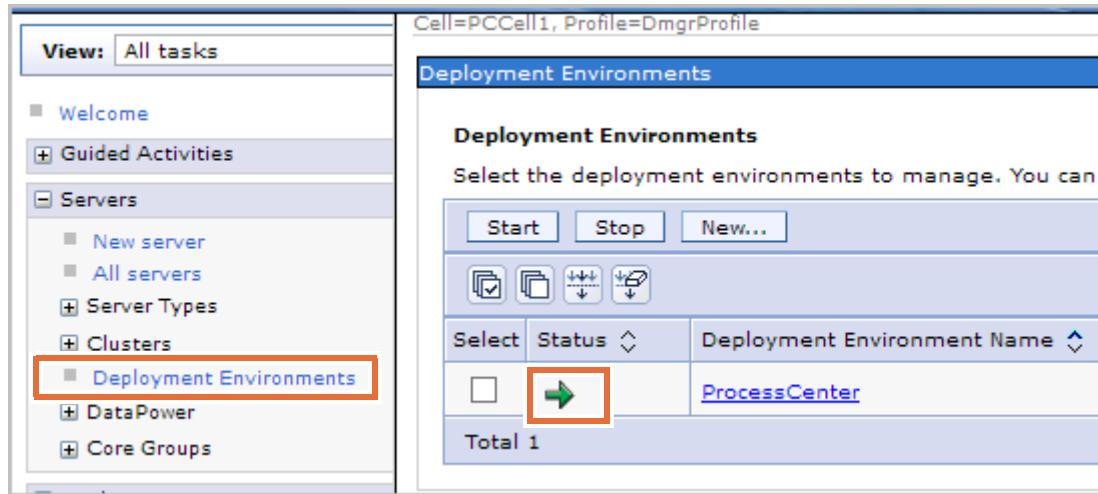


- a. A DOS command window is displayed; press any key to continue when prompted.
- 3. Start the process center single cluster.
- a. On your Windows desktop, select the **Start Process Center Cluster** shortcut. Double-click the shortcut or press Enter to start the cluster member.



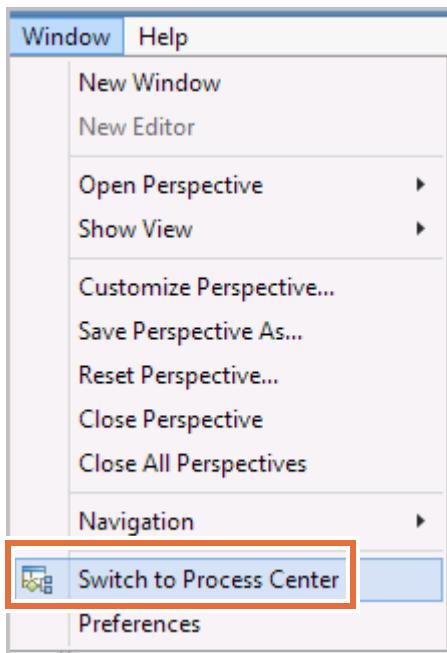
- b. A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts. Press any key to continue when prompted.

4. Verify that the Process Center Single Cluster is started.
- \_\_ a. On your Windows desktop, select the **Deployment Manager Administrative Console** shortcut. Double-click the shortcut or press Enter to go to the console
  - \_\_ b. Log in by entering `bpmadmin` and `web1sphere` in the **User ID** and **Password** fields.
  - \_\_ c. Select **Servers > Deployment Environments**.

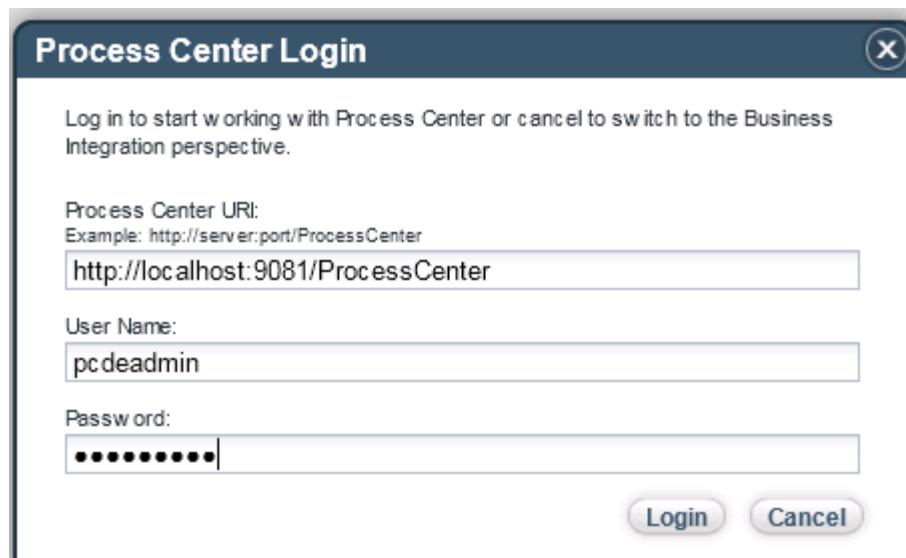


- \_\_ d. Verify that the deployment environment is started. When it starts, the status turns green. It might be necessary to refresh the browser or click the **Deployment Environments** link to view the status change.
  - \_\_ e. Click **Servers > Clusters > WebSphere application server clusters** to verify that the single cluster is running.
  - \_\_ f. Click **Logout** to exit.
  - \_\_ g. Close the browser.
5. Open the Exercise 16 workspace.
- \_\_ a. On your desktop, open the **Exercise Shortcuts** folder.
  - \_\_ b. Double-click the **Exercise 16** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
  - \_\_ c. Close the **Getting Started** tab.

- 6. Switch to the Process Center perspective by clicking **Window > Switch to Process Center** in the menu options.



- 7. In the Process Center Login window, enter the following credentials:
- **Process Center URI:** `http://localhost:9081/ProcessCenter`
  - **User Name:** `pcdeadmin`
  - **Password:** `web1sphere`



- 8. Click **Login**.
- 9. When the Secure Storage window opens, click **Cancel**.
- 10. If a Security Alert window is displayed asking to proceed, then click **Yes** each time.
- 11. When the Secure Storage Warning window opens, click **OK**.

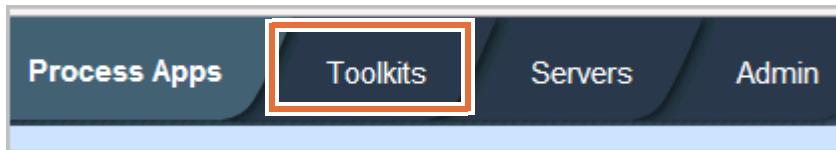
- \_\_\_ 12. Close the “Getting Started with IBM Process Center 8.5.7.0” welcome screen, by clicking the X at the upper-right corner of the window.

## **Part 2: Create a Toolkit in IBM Process Center**

You can create toolkits to enable IBM Process Designer users to share library items across process applications. The Process Center perspective is the way to establish and maintain relationships between toolkits, process applications, and SCA services. You can do tasks such as:

- Create a process application or toolkit
- Associate SCA services with a process application or toolkit
- Import process application and toolkit artifacts into the IBM Integration Designer
- Publish process applications and toolkits to the Process Center

- \_\_\_ 1. In the Process Center perspective, click the **Toolkits** tab.



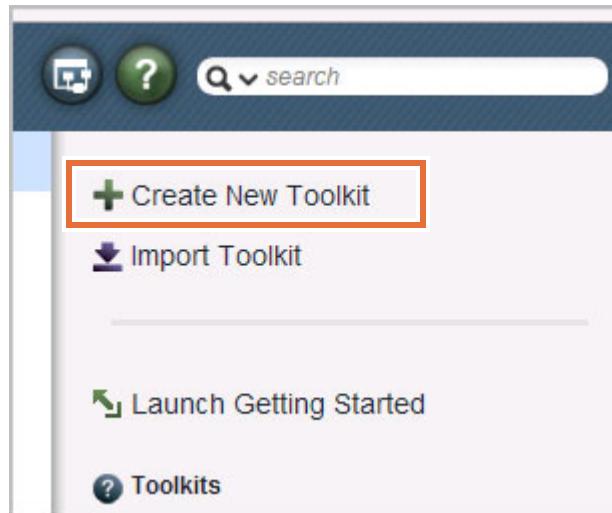
2. Verify that one toolkit is named **System Data**, which is listed under the Toolkits tab. This toolkit is the default, which is created during installation and is imported into the Process Center repository. Each process application and toolkit that you create automatically includes a System Data toolkit dependency. You have access to the assets that all IBM Business Process Manager projects require, such as standard variable types and standard charts for reports.

The screenshot shows the 'Toolkits' tab selected in the navigation bar. Below the tabs, there is a list of toolkits:

- Coaches (deprecated) (SYSC) ★ ?  
Last updated on 3/10/16 by pdeadmin
- System Data (TWSYS)** ★ ?  
Last updated on 3/10/16 by pdeadmin
- Responsive Portal Components (SYSRPC) ★ ?  
Last updated on 3/10/16 by pdeadmin
- SAP Guided Workflow (SGW) ★ ?  
Last updated on 3/10/16 by pdeadmin
- System Governance (TWSYSG) ★ ?  
Last updated on 3/10/16 by pdeadmin
- Dashboards (SYSD) ★ ?  
Last updated on 3/10/16 by pdeadmin
- Content Management (SYSCM) ★ ?  
Last updated on 3/10/16 by pdeadmin
- Responsive Coaches (SYSRC) ★ ?  
Last updated on 3/10/16 by pdeadmin

The 'System Data (TWSYS)' toolkit is highlighted with a red border.

- \_\_\_ 3. In the upper-right corner of the Process Center perspective, click **Create New Toolkit**.



- \_\_\_ 4. In the Create New Toolkit window, enter the following information:

- Enter Account Verification Services in the **Toolkit Name** field.
- Enter AVS101 in the **Acronym** field. The acronym must be unique in the repository. You can change this value in case this acronym is in use in the repository.
- Enter Account Integration Service for Customer in the **Description** field.

The 'Create New Toolkit' dialog box is shown. It has fields for 'Toolkit Name' (Account Verification Services), 'Acronym' (AVS101), and 'Documentation' (Account Integration Service for Customer). A rich text toolbar is visible above the documentation field.

Note the rich text formatting options available in the toolbar for the **Description** field. You can apply different font options, font sizes, alignment, and indentation to the text. Feel free to explore the different formatting options.

- \_\_\_ 5. Click **Create** to create the toolkit. When it is created, the **Create New Toolkit** window closes.

- \_\_\_ 6. Verify that the newly created toolkit is listed in the **Toolkit** tab.

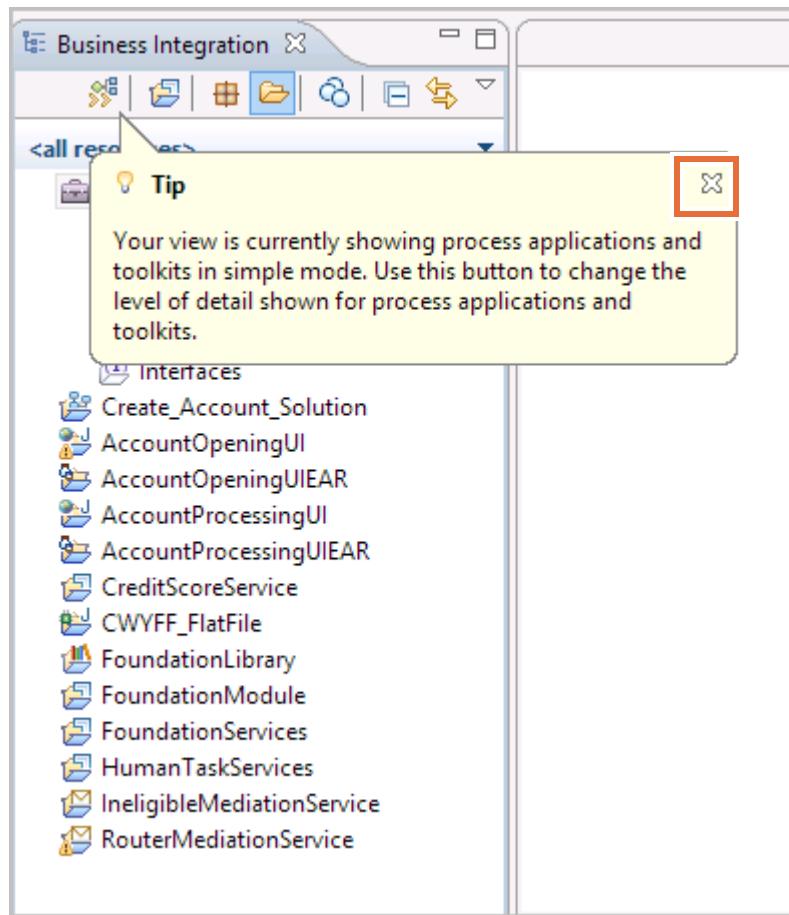
The screenshot shows a user interface for managing toolkits. At the top, there are tabs: Process Apps, Toolkits (which is the active tab, indicated by a blue background), Servers, and Admin. Below the tabs, there is a list of toolkits:

- Account Verification Services (AVS101)** (highlighted with a red border)  
Last updated on 5/23/16 by podeadmin
- Coaches (deprecated) (SYSC)**  
Last updated on 3/10/16 by podeadmin
- System Data (TWSYS)**  
Last updated on 3/10/16 by podeadmin

### **Part 3: Associate IBM Integration Designer artifacts with the toolkit**

- \_\_\_ 1. In the **Toolkits** tab, click **Open in workspace** next to the **Account Verification Services** toolkit.

2. The Business Integration perspective opens with the **Account Verification Services** project. If displayed, close the Tip window by clicking X. Wait for the workspace to build before going to the next step. It might take a while.

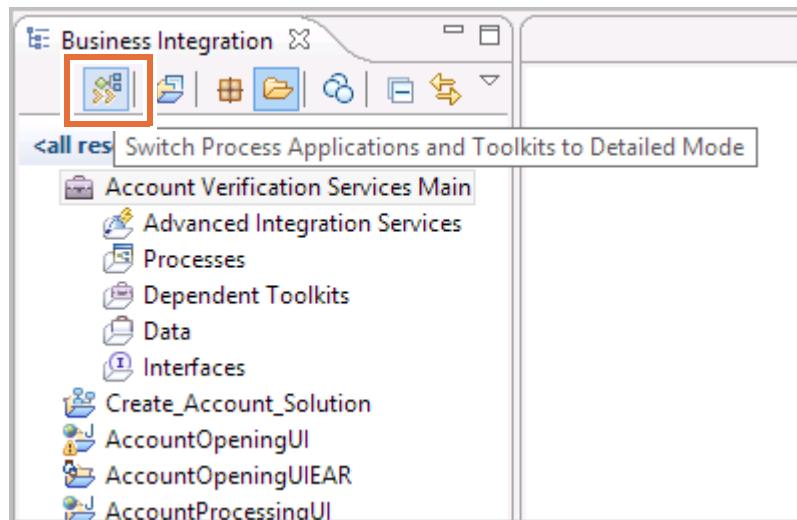


3. Select **Account Verification Services Main** and click the **Properties** tab. Examine the values.

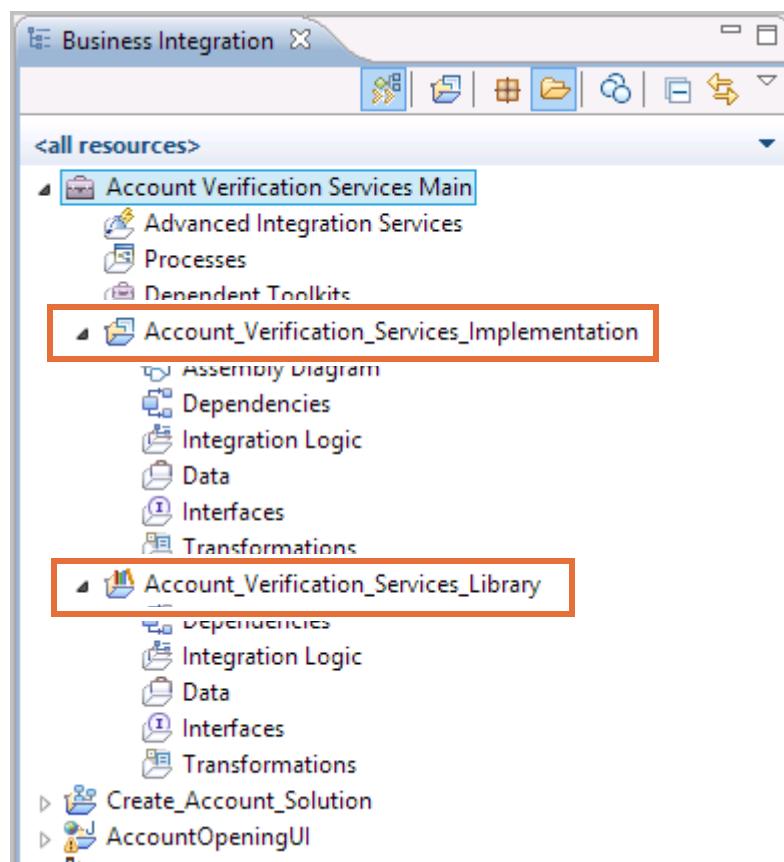
The screenshot shows the Properties tab of a selected project. The table displays the following properties:

| Property           | Value                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------|
| Toolkit Name       | Account Verification Services                                                           |
| Acronym            | AVS101                                                                                  |
| Track              | Main                                                                                    |
| Track Acronym      | Main                                                                                    |
| Process Center URL | <a href="https://localhost:9444/ProcessCenter">https://localhost:9444/ProcessCenter</a> |
| Connection State   | Connected                                                                               |

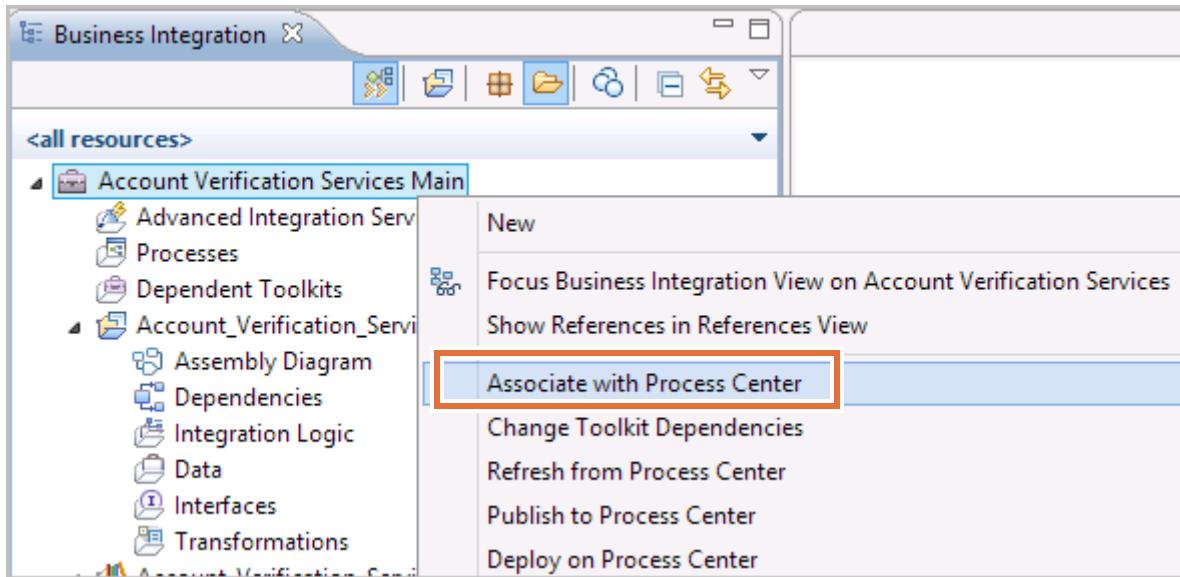
- 4. If you do not see the implementation and library projects, switch your view to Detailed mode. You can switch views by clicking the **Switch Process Applications and Toolkits to Detailed Mode** icon.



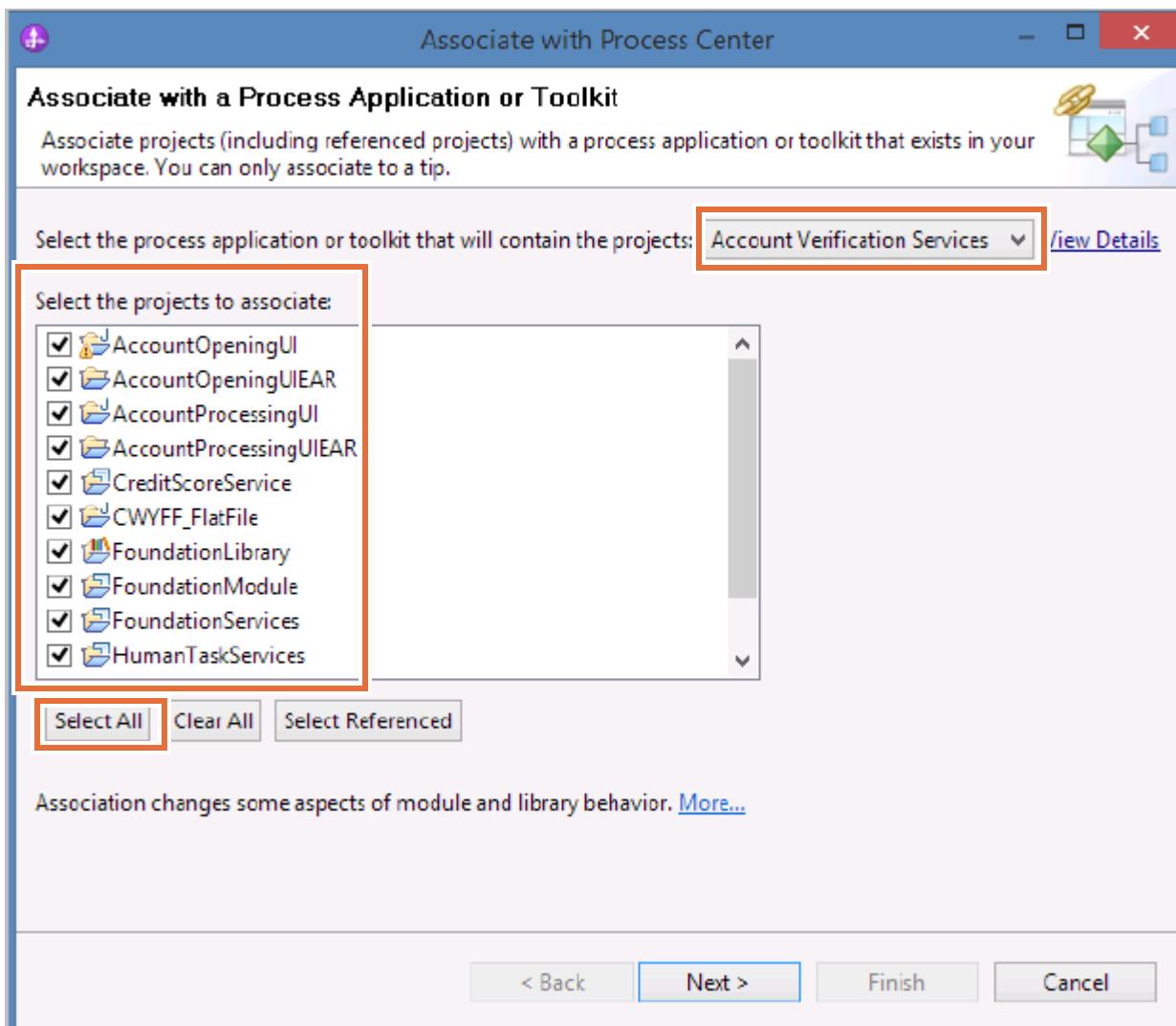
- 5. Expand **Account\_Verification\_Services\_Implementation** and **Account\_Verification\_Services\_Library**.



- \_\_\_ 6. Associate the imported project with the **Account Verification Services** toolkit you created earlier.
  - \_\_\_ a. Right-click **Account Verification Services Main**, and click **Associate with Process Center**.

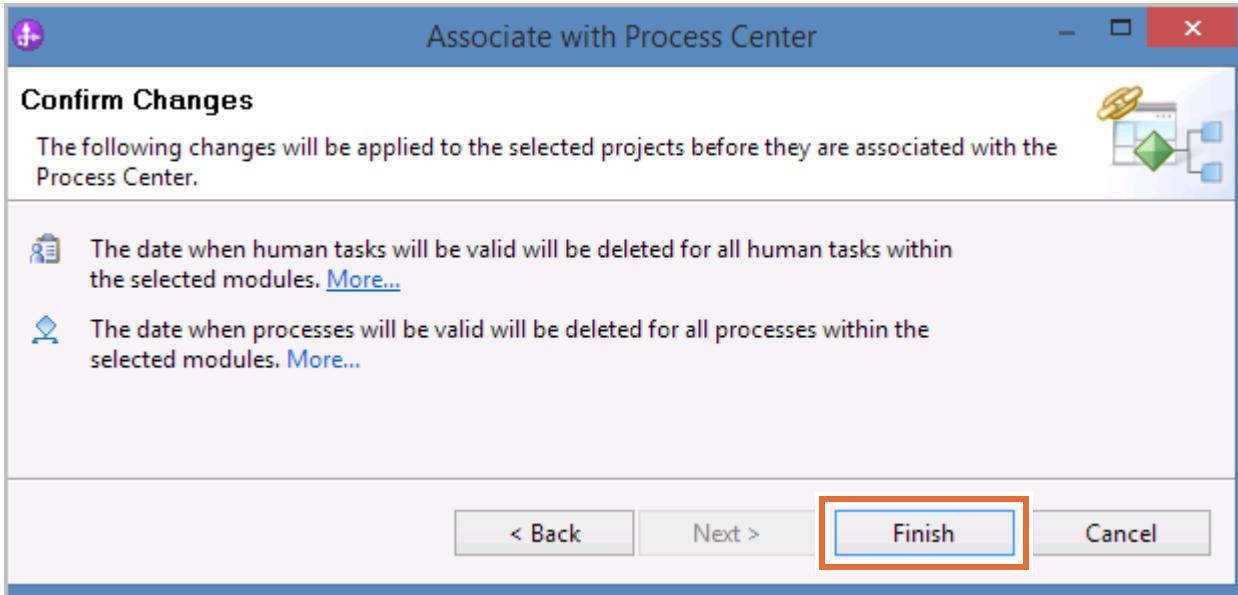


- \_\_\_ b. In the “Associate with Process Center” window, verify that **Account Verification Services** is selected as the toolkit. Click **Select All** to associate the toolkit with the other projects in IBM Integration Designer.



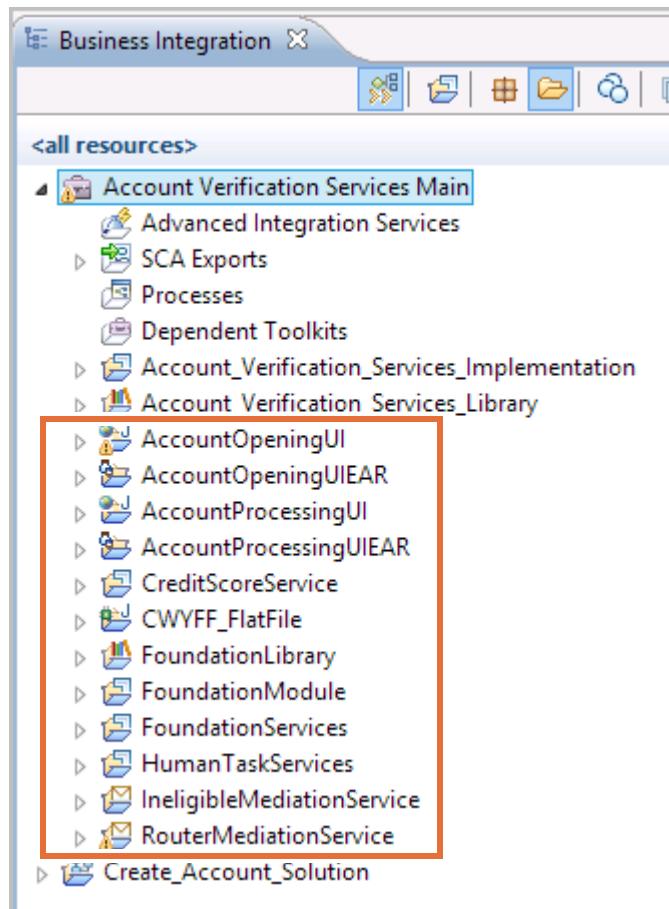
- \_\_\_ c. Click **Next**.

- \_\_ d. Click **Finish** in the “Confirm changes” pane to complete the association.

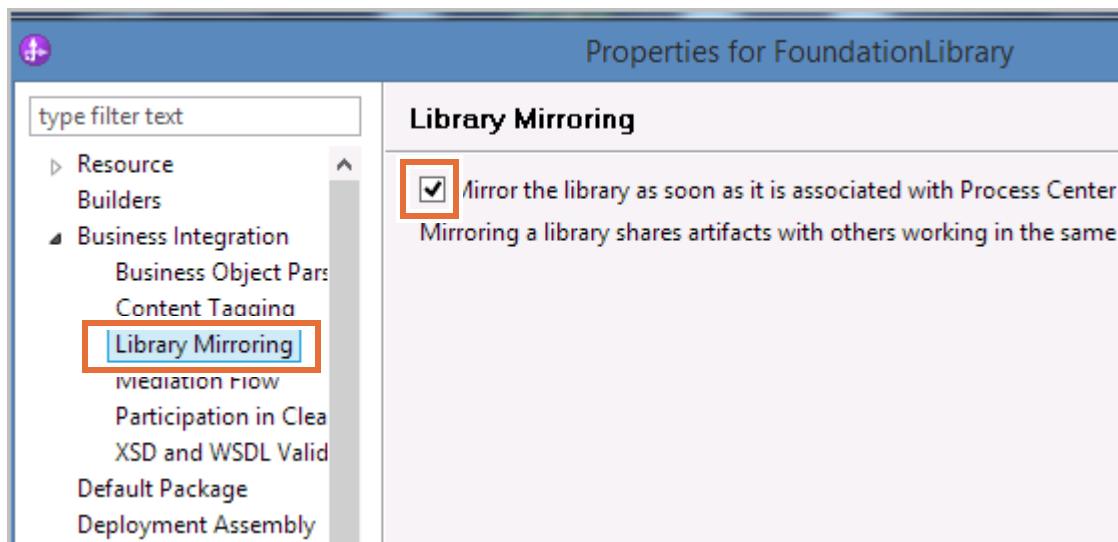


- \_\_ 7. It takes a while for the association with the Process Center to take place. Watch the status at the lower right. Wait until the status reaches 100 percent, which indicates that the update is complete.

8. Examine the **Account Verification Services Main** project on the left. All the modules are moved into it.



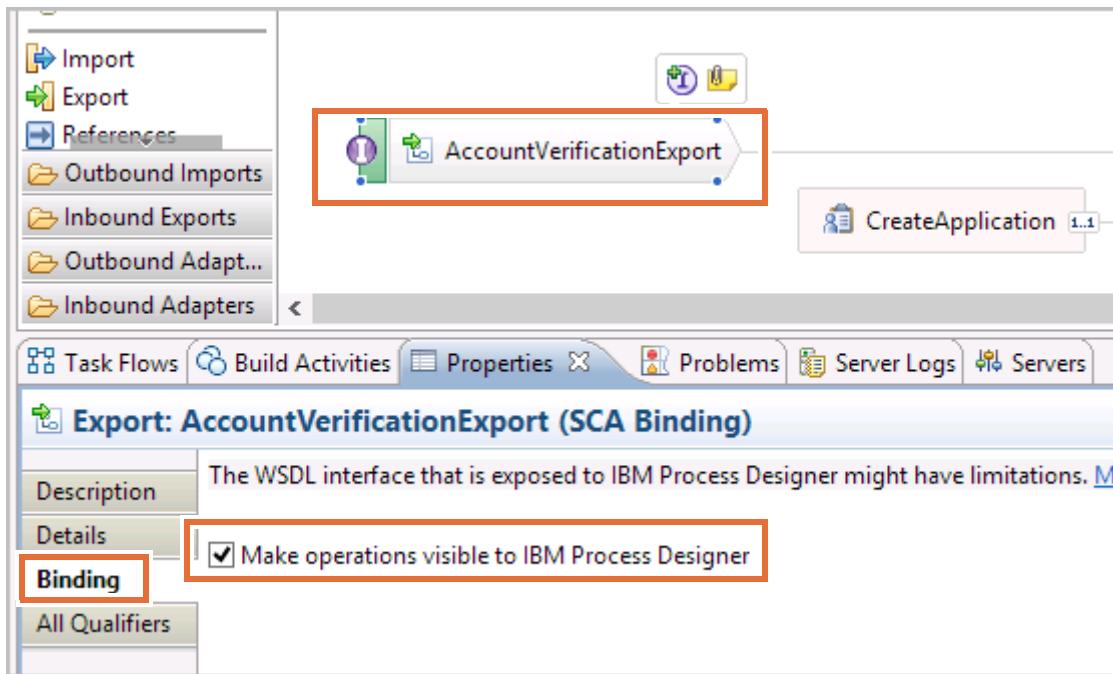
9. Right-click **FoundationLibrary** and click **Properties**.  
 10. In the “Properties for FoundationLibrary” pane, expand **Business Integration** and select **Library Mirroring**.  
 11. Select **Mirror the library as soon as it is associated with Process Center** and click **OK**.



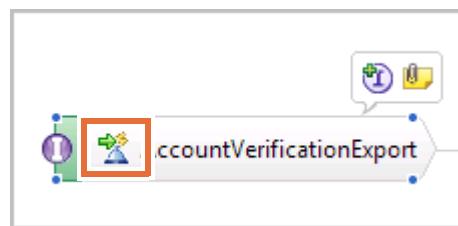
**Information**

In a collaborative development environment between IBM Integration Designer and IBM Process Designer, artifacts like business objects are shared in libraries. When you put an artifact in your library in Integration Designer, it is made available to others who work with the same library in Process Designer. This sharing of artifacts is called library mirroring.

- 12. Make the operations available in IBM Process Designer by setting the attributes in the IBM Integration Designer.
  - a. Expand **FoundationModule** and double-click **Assembly Diagram**.
  - b. In the assembly diagram, select **AccountVerificationExport** and click the **Properties** tab.
  - c. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.

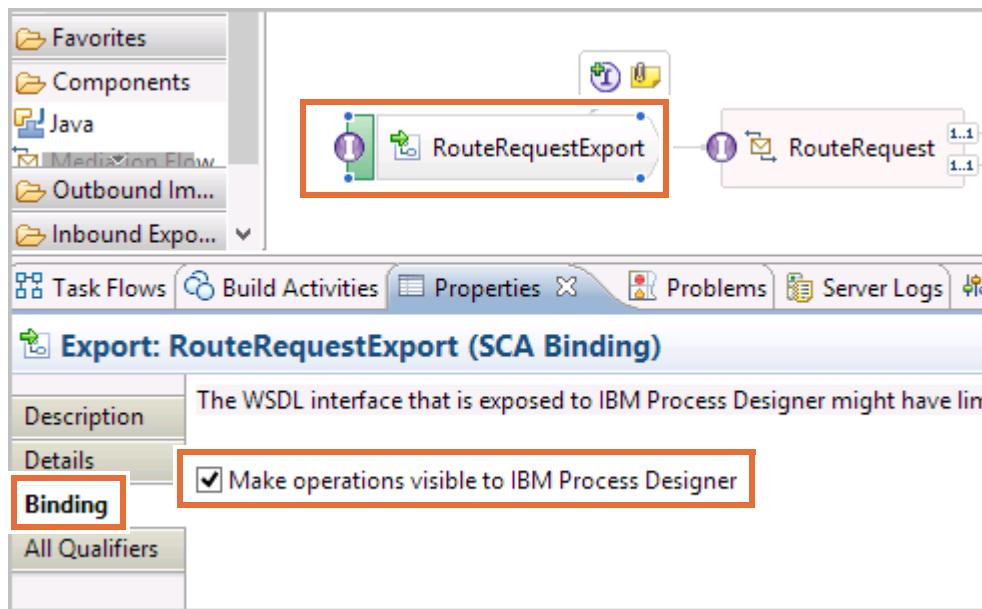


- d. Click **File > Save All** to save your changes.
- e. Notice the new icon for the export.

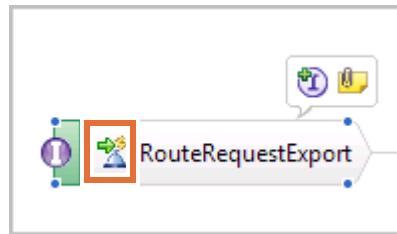


- f. Expand **RouterMediationService** and double-click **Assembly Diagram**.
- g. In the assembly diagram, select **RouteRequestExport** and click the **Properties** tab.

- \_\_ h. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.

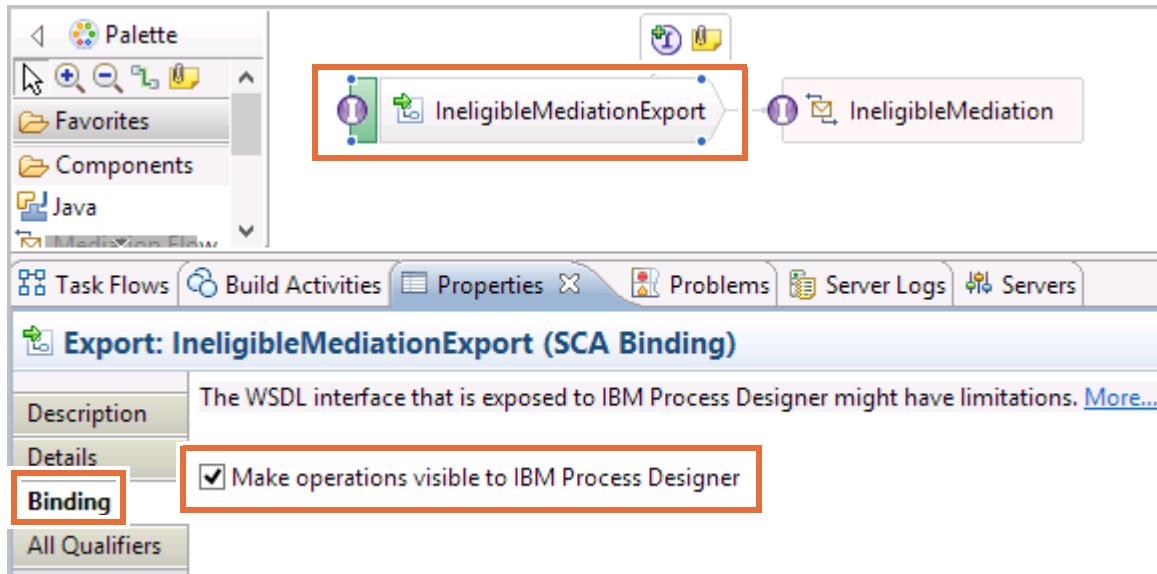


- \_\_ i. Click **File > Save All** to save your changes.  
\_\_ j. Notice the new icon for the export.



- \_\_ k. Expand **IneligibleMediationService** and double-click **Assembly Diagram**.  
\_\_ l. In the assembly diagram, select **IneligibleMediationExport** and click the **Properties** tab.

- \_\_ m. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.



- \_\_ n. Click **File > Save All** to save your changes.

- \_\_ o. Notice the new icon for the export.

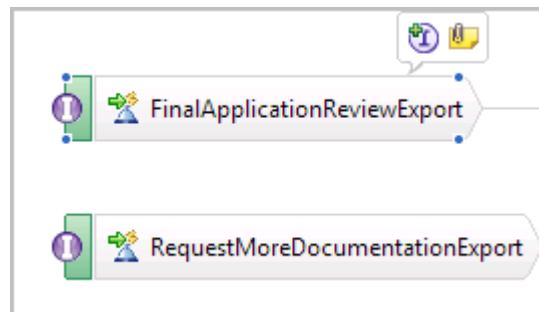


- \_\_ p. Expand **HumanTaskServices** and double-click **Assembly Diagram**.

- \_\_ q. Select the **Make operations visible to IBM Process Designer** for two export components in the assembly diagram: **RequestMoreDocumentationExport** and **FinalApplicationReviewExport**. The properties are in the **Binding** tab in the **Properties** view for the respective components.

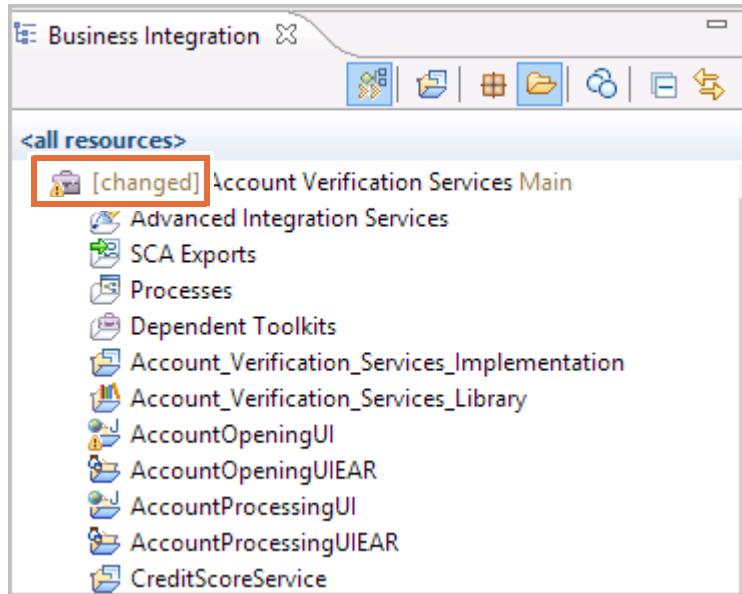
- \_\_ r. Click **File > Save All** to save your changes.

- \_\_ s. Notice the new icon for the export.

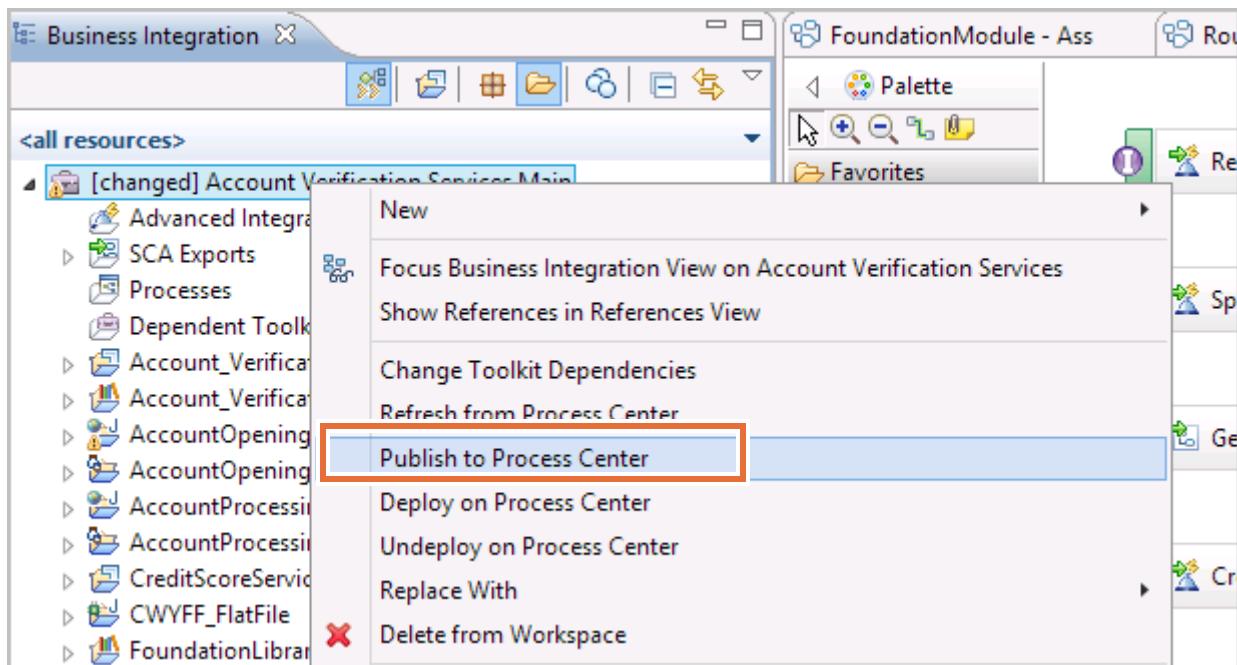


- \_\_ t. Expand **FoundationServices** and double-click **Assembly Diagram**.

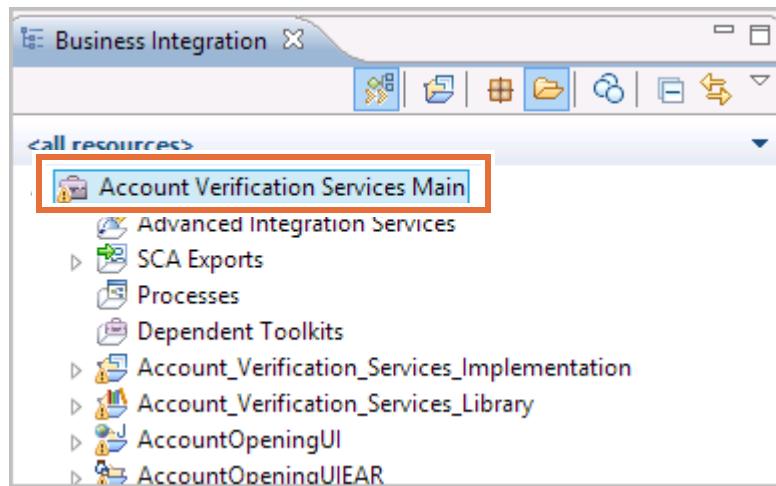
- \_\_ u. Select the **Make operations visible to IBM Process Designer** for three export components in the assembly diagram: **RecordIneligibleApplicationExport**, **SpecialDeclineExport**, and **CreditAssessmentExport**. The properties are in the **Binding** tab in the **Properties** view for the respective components.
  - \_\_ v. Click **File > Save All** to save your changes.
  - \_\_ w. Notice the new icons for the three exports.
- \_\_ 13. Examine the changed status of the **Account Verification Services Main** project. The newly added text [changed], which is to the left of the project, indicates the changed status.



- \_\_ 14. Publish **Account Verification Services Main** to the Process Center.
- \_\_ a. Right-click **Account Verification Services Main** and click **Publish to Process Center**.



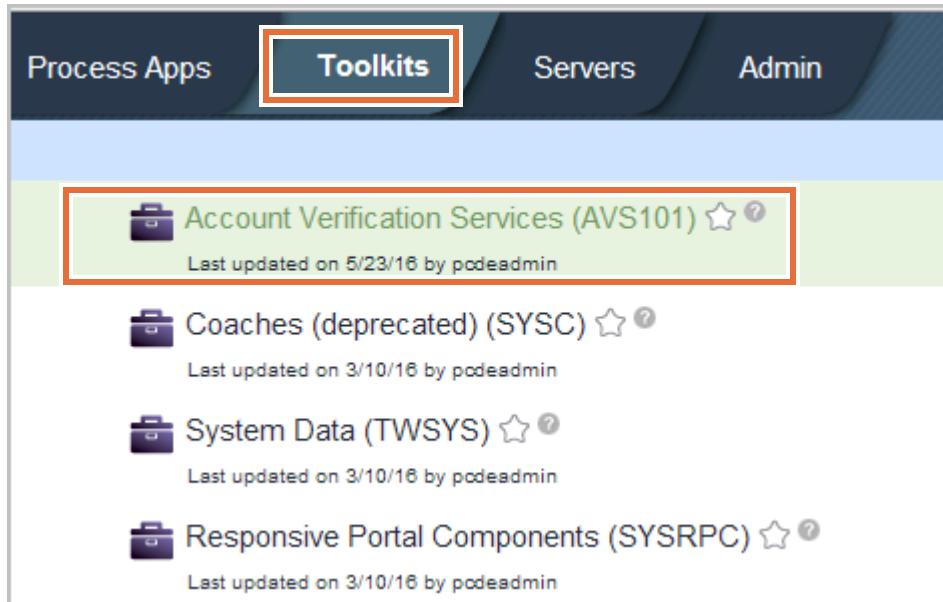
- \_\_\_ 15. Click **OK** in the **Library Mirroring** window.
- \_\_\_ 16. Wait for the update to complete. It takes a while for the status at the lower right to change. The update is complete when the status **changed** is no longer displayed next to **Account Verification Services Main** as it was before.



## Create a snapshot of the published toolkit

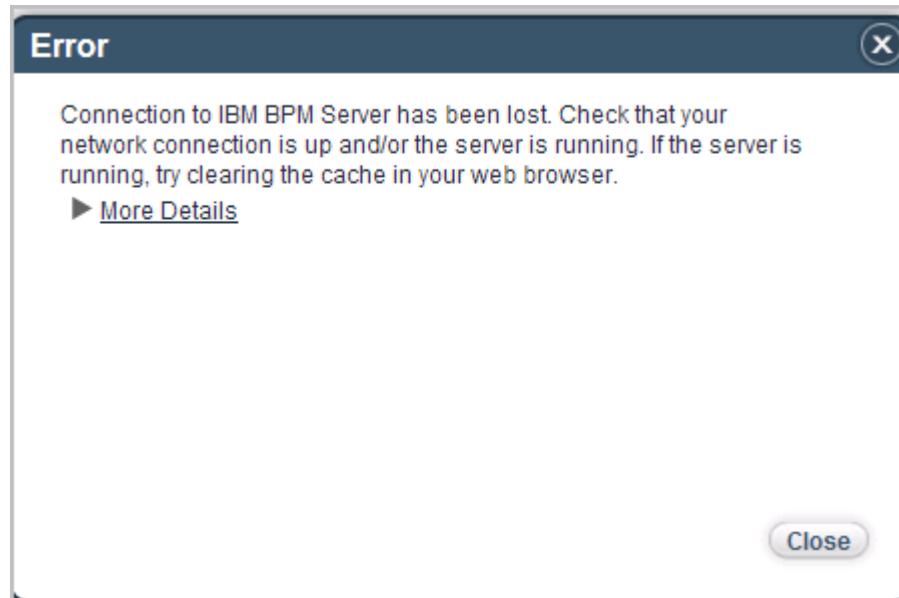
Snapshots record the state of library items within a process application or track at a specific point in time. You can create snapshots in the Process Center console or in the IBM Integration Designer view. Snapshot management, such as deploying, exporting, and archiving, is done in the Process Center console.

- \_\_\_ 1. Switch to the Process Center perspective (click **Window > Switch to Process Center**) and verify that you are in the **Toolkits** tab. Click **Account Verification Services (AVS101)**.

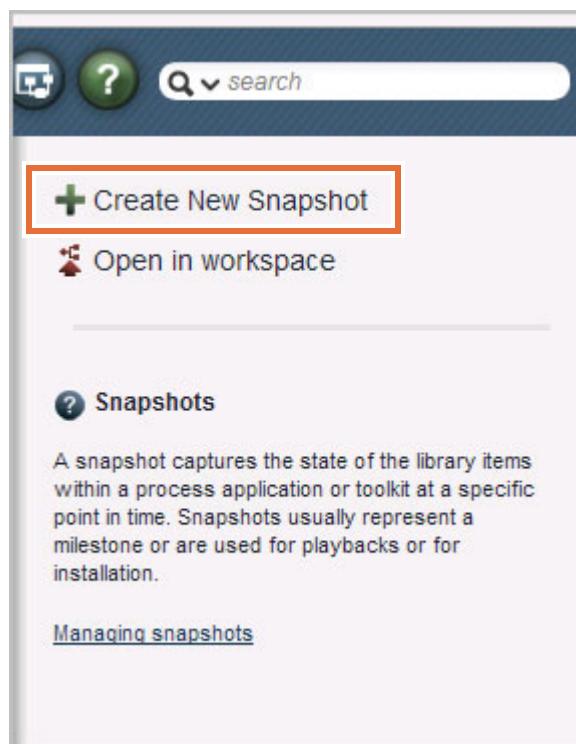


**Important**

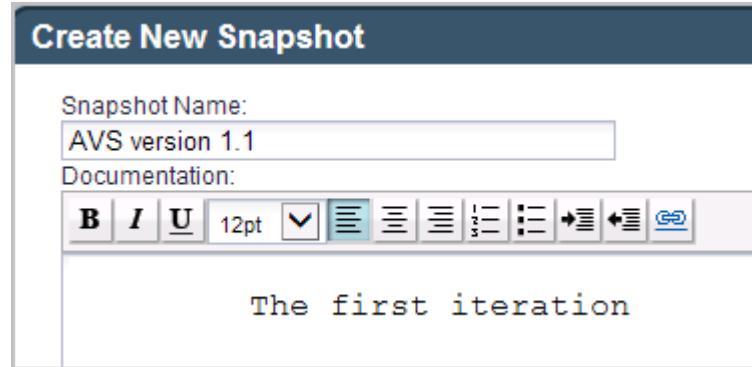
At any time, while working with the Process Designer or Process Center, you might receive an IBM Business Process Manager error with a message that the connection was lost. If that occurs, do not be concerned. Click **Close** and continue with your lab exercise.



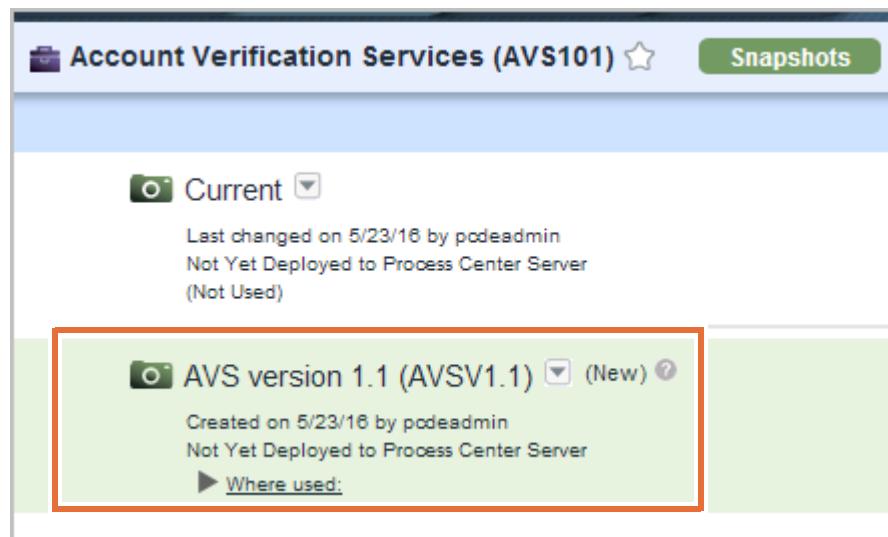
2. Click **Create New Snapshot** on the right.



- \_\_\_ 3. In the Create New Snapshot window, enter AVS version 1.1 in the **Snapshot Name** field and enter The first iteration as the **Description**.

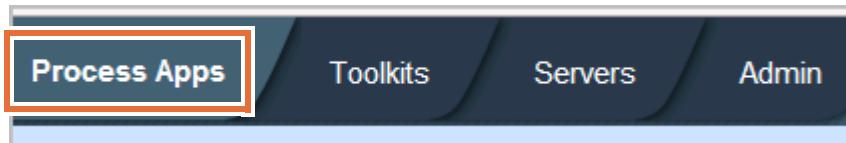


- \_\_\_ 4. Click **Create**. It is not necessary to click **Create** twice. When it is created, the **Create New Snapshot** window closes.
- \_\_\_ 5. The toolkit is now ready to be shared among process developers (refresh might be needed).

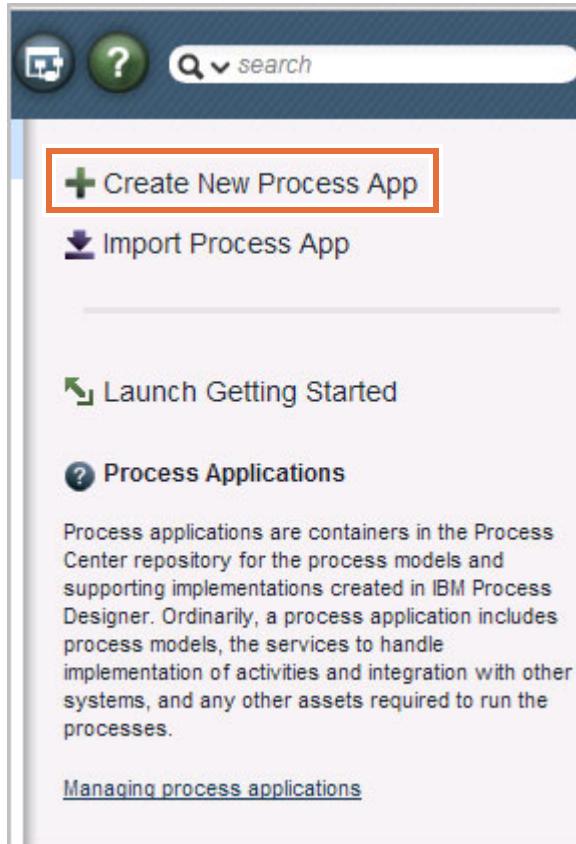


#### **Part 4: Generate access to process applications and toolkits**

- \_\_\_ 1. Click the **Process Apps** tab at the upper left of Process Center.



- 2. Click **Create New Process App** from the upper-right corner of the screen.



- 3. In the Create New Process App window, enter the following information:

- **Process App Name:** AccountServicesApp
- **Acronym:** ASA101
- **Description:** The Account Services process application uses the Account Verification toolkit

A screenshot of the 'Create New Process App' dialog box. The title bar says 'Create New Process App'. The form has three fields: 'Process App Name:' with 'AccountServicesApp' typed in, 'Acronym:' with 'ASA101' typed in, and 'Documentation:' with a rich text editor containing the text 'The Account Services process application uses the Account Verification toolkit'.

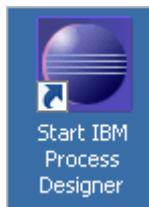
**Hint**

The acronym must be unique in the repository. You can change this value in case this acronym is in use in the repository.

- \_\_\_ 4. Click **Create**. The newly created process application is listed in the tab.

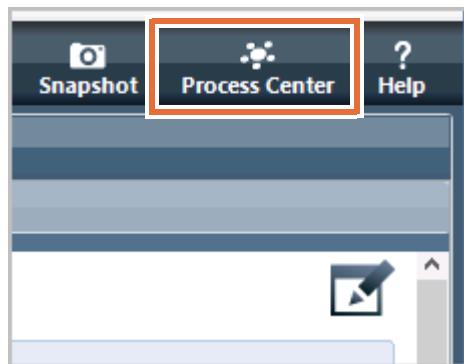
The screenshot shows a navigation bar with tabs: Process Apps, Toolkits, Servers, and Admin. Below the tabs, there is a list of process applications. The first application, "AccountServicesApp (ASA101)", is highlighted with a red box. It has a star icon and a question mark icon next to it. Below the application name, it says "Last updated on 5/23/16 by pdeadmin". The other two applications listed are "Hiring Sample (HSS)" and "Hiring Sample Advanced (HSBV1)", each with similar star and question mark icons and update information.

- \_\_\_ 5. Set the dependency between the **AccountServicesApp** and the **Account Verification Services** toolkit.  
\_\_\_ a. Start IBM Process Designer and log in to the IBM Process Center repository. On the Windows desktop, locate the icon that is labeled **Start IBM Process Designer**.

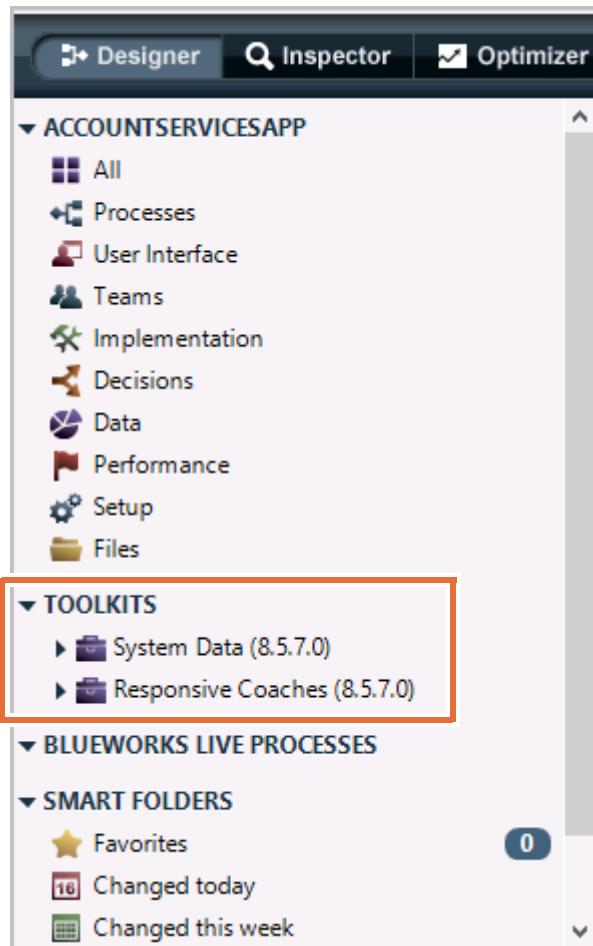


- \_\_\_ b. Double-click the icon or press Enter to start IBM Process Designer.  
\_\_\_ c. To log in and connect to the IBM Process Center repository, enter `pcdeadmin` and `web1sphere` for the user name and password fields.  
\_\_\_ d. Click **Login**.  
\_\_\_ e. In the **Security Alert** dialog box, click **Yes**.  
\_\_\_ f. Click **Yes** one more time to continue. After a few moments, IBM Process Designer starts.

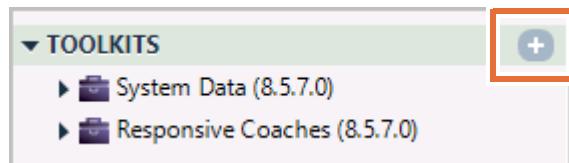
- \_\_\_ g. If you are not already on the Process Center page, switch to the **Process Center** tab by clicking **Process Center** at the upper right.



- \_\_\_ 6. Click **Open in Designer** next to **AccountServicesApp (ASA101)**.
- \_\_\_ 7. Expand **Toolkits**, unless it is already expanded. The default **System Data** and **Coaches** toolkits are listed.



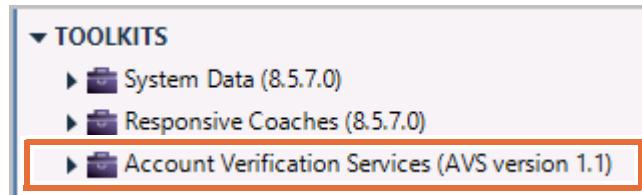
- \_\_\_ 8. Click the plus sign (+) to the right of **Toolkits**. The plus sign (+) is not visible by default. You must hover over **Toolkits** to see the icon.



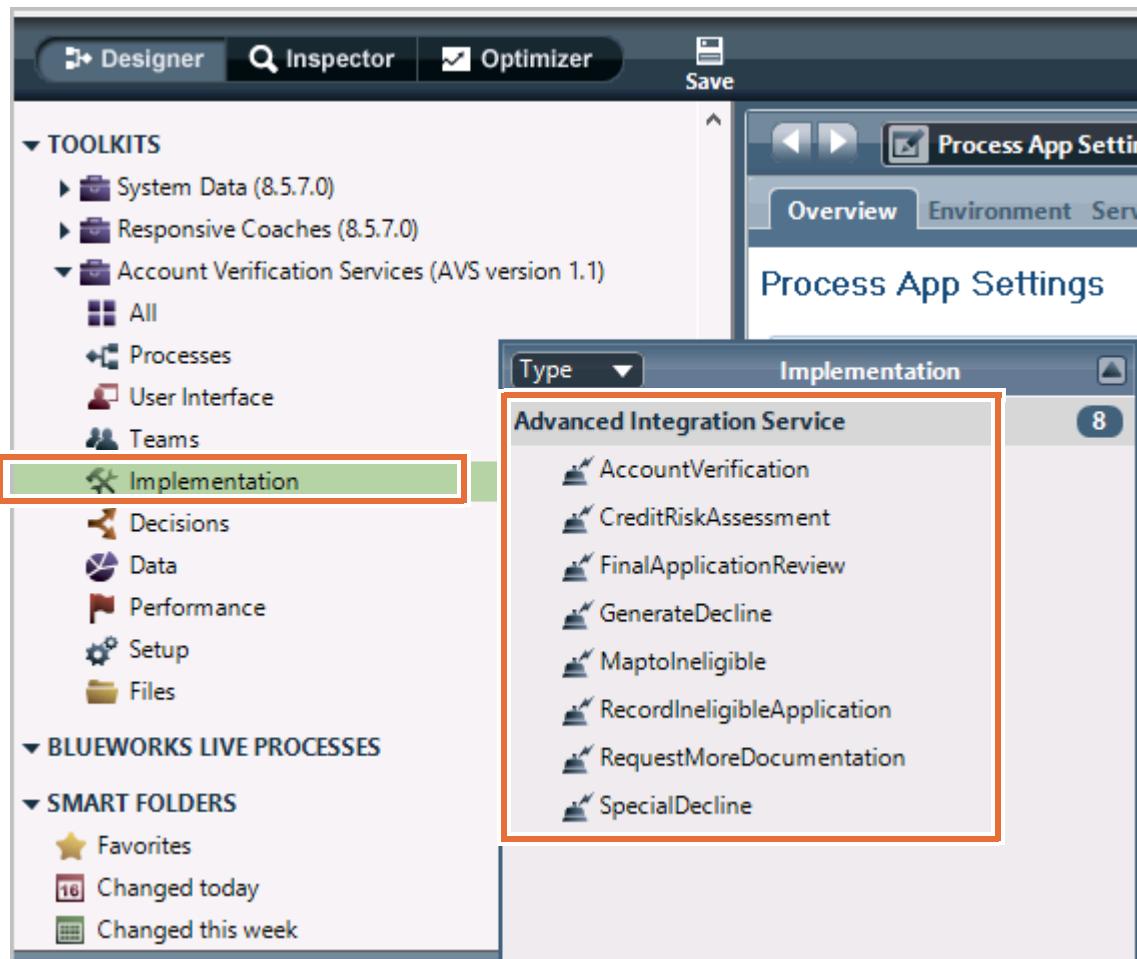
- \_\_\_ 9. Click **AVS Version 1.1** to add the dependency to the **AccountServicesApp** process application.



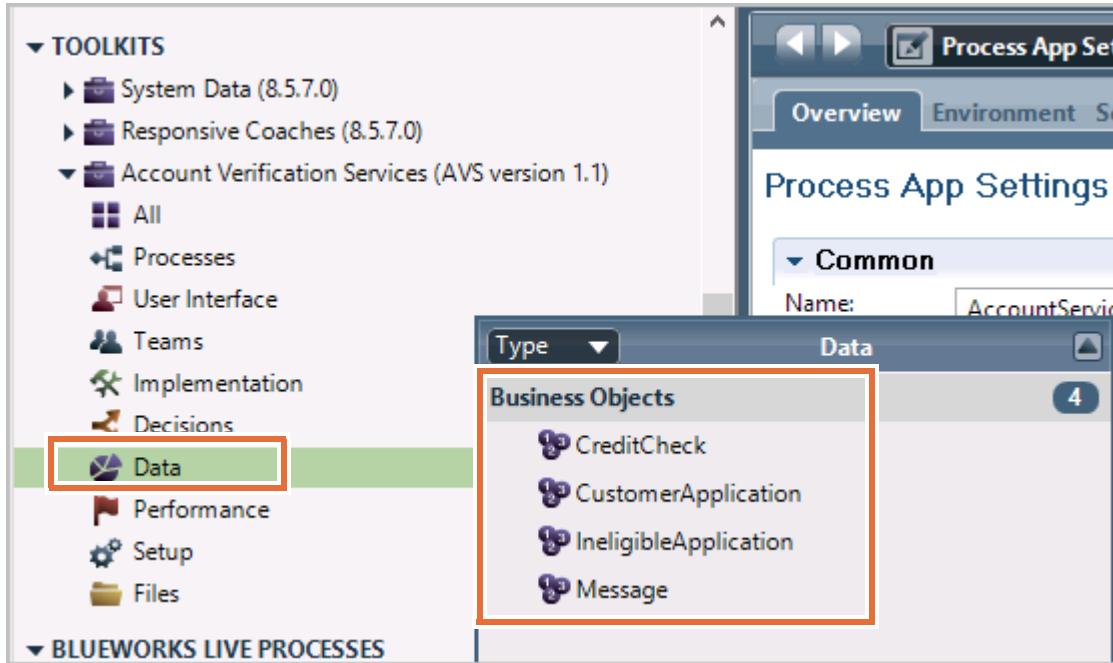
- \_\_\_ 10. Verify that the newly added dependency is listed under **Toolkits**.



11. Expand **Account Verification Services (AVS version 1.1)** and click **Implementation** to view its contents.

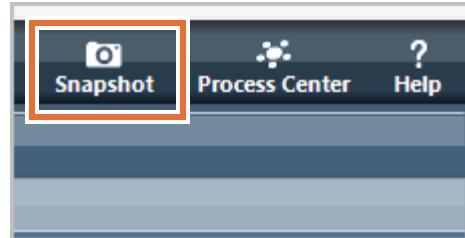


- 12. Click **Data** to see a list of business objects from the SCA module.



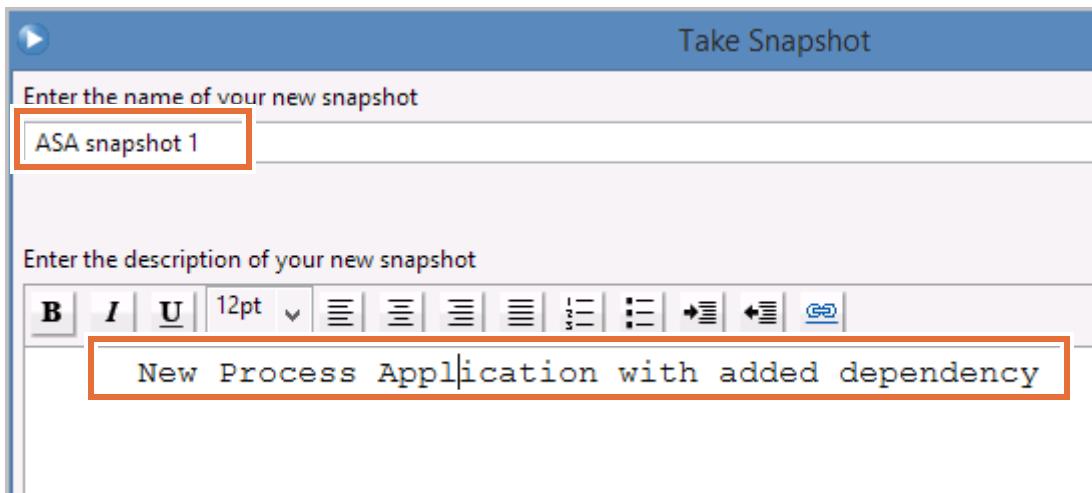
## Deploying the Account Services process application

- 1. Before you deploy the process application, you must take its snapshot. In the Process Designer, click **Snapshot**.

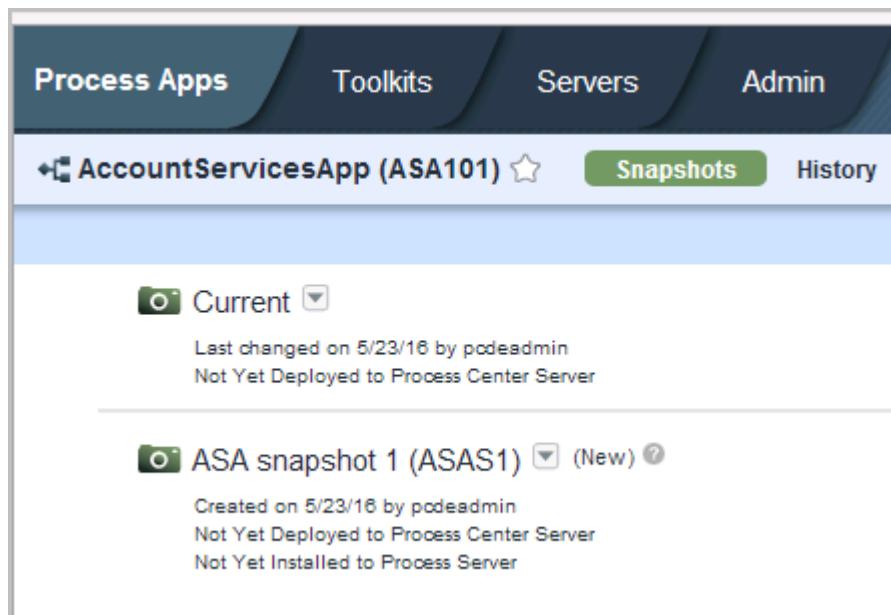


- 2. In the Take Snapshot window, enter ASA snapshot 1 in the **Enter the name of your new snapshot** field.

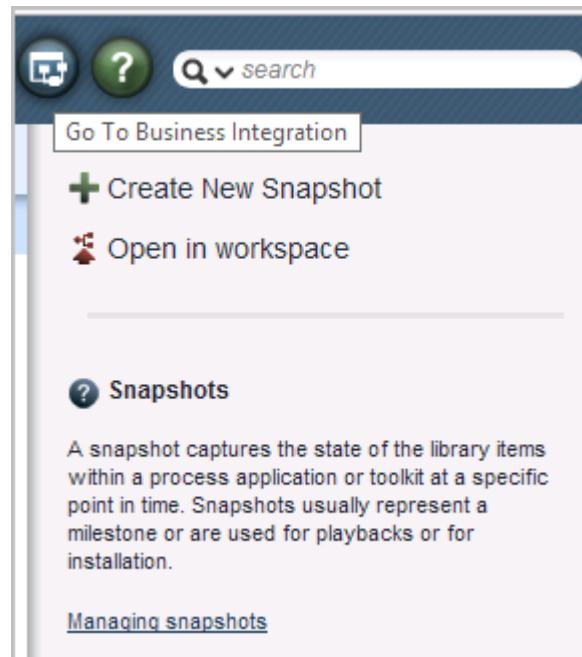
- 3. Enter New Process Application with added dependency in the **Enter the description of your new snapshot** field. Several rich text formatting options are available here.



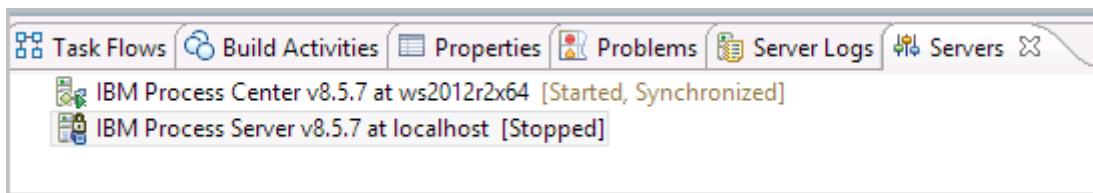
- 4. Click **OK**.
- 5. Verify that the new snapshot is listed following the **Revision History** section.
- 6. Minimize the Process Designer window as you are going to work with it again. If system resources are running low, you can also choose to close the IBM Process Designer.
- 7. Switch to the IBM Integration Designer window. Verify that you are in the Process Center perspective and click the **Process Apps** tab.
- 8. Click **AccountServicesApp (ASA101)**. The newly created snapshot is displayed under the **Snapshots** tab.



- \_\_\_ 9. The message that follows the snapshot gives the deployment status. Currently, the process application is not deployed. Before you deploy to IBM Process Server, start the server. Remember that only the Process Center server is running. The IBM Process Server must be started for a deployment.
- \_\_\_ a. Click the **Go to Business Integration** icon. You are returned to the Business Integration perspective.



- \_\_\_ b. Go to the **Servers** view. Two servers are present at the local host: IBM Process Center v8.5.7 and IBM Process Server v8.5.7. If you do not see that the Process Center server is listed, then verify that your Process Center server is running. When the Process Center server is running, it is automatically listed in the Servers view.



- \_\_\_ c. If IBM Process Server v8.5.7 is not started already, right-click **IBM Process Server v8.5.7 at localhost**, and click **Start**. It takes several minutes for the server to start. If the server is already started, then go to the next step.
- \_\_\_ 10. When the server starts, return to the Process Center perspective by clicking **Window > Switch to Process Center**.
- \_\_\_ 11. Return to the **Process Apps** tab and click the **AccountServicesApp (ASA101)** link.

- 12. To the right of **ASA snapshot 1**, click **Install**.

The screenshot shows the 'Schemas' tab selected in the top navigation bar. Below it, a table lists application snapshots. The first row is for 'Current'. The second row is for 'ASA snapshot 1 (ASAS1)', which is labeled '(New)' and has a status of 'Not Yet Deployed to Process Center Server'. To the right of this row are 'Export' and 'Install' buttons. The 'Install' button is highlighted with a red box.

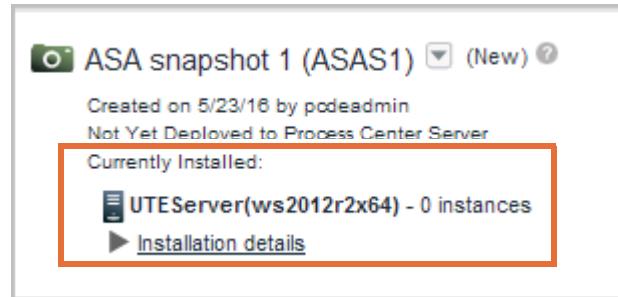
- 13. In the “Install Snapshot to Server” window, select **UTEServer**. A check mark is displayed to the right.

The screenshot shows the 'Install Snapshot to Server' dialog box. It asks to select a server to install snapshot ASA snapshot 1 to. A list box contains 'UTEServer (ws2012r2x64)' with a status of 'TEST - Status: Connected' and a green checkmark icon to its right.

- 14. Click **Install** at the lower right of the window to begin deployment. It takes several minutes for the process application to install on the Process Server test environment. In the meantime, the status changes to **Installation in progress**.

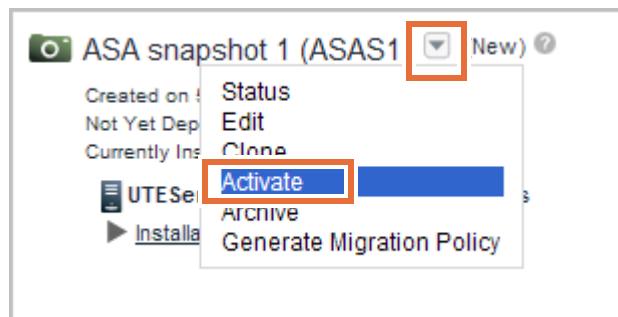
The screenshot shows the details page for 'ASA snapshot 1 (ASAS1)'. It displays the creation date ('Created on 5/23/16 by pcdeadmin') and a message box stating 'UTEServer(ws2012r2x64) - Installation in progress'. This message box is highlighted with a red border.

- 15. Do not click other options in the Process Center. When the snapshot is installed, the status that is displayed is updated to **Currently Installed**.

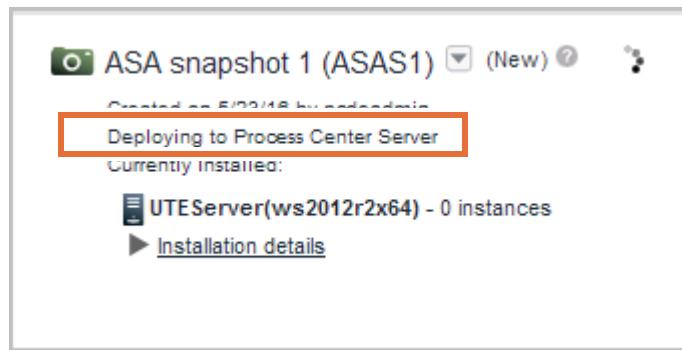


The process application is not yet deployed to the Process Center. Note the status: Not yet Deployed to Process Center Server

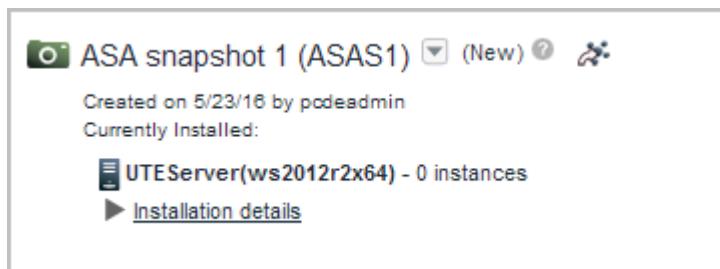
- 16. Click the down arrow next to the snapshot and select **Activate**.



- 17. It takes several minutes for process application to deploy. In the meantime, the status changes to “Deploying to Process Center Server”.



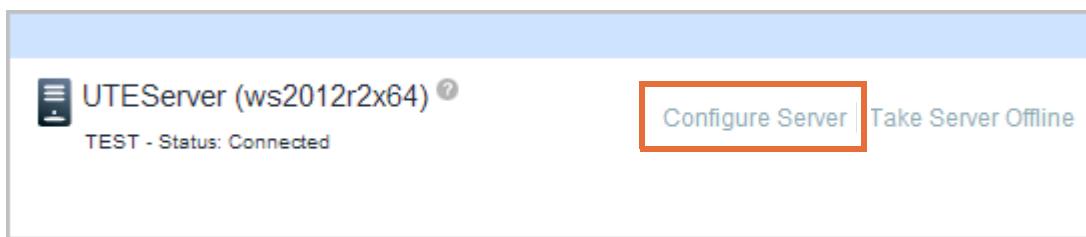
- 18. When the deployment is done, verify that the “Deploying to Process Center Server” message is no longer displayed.



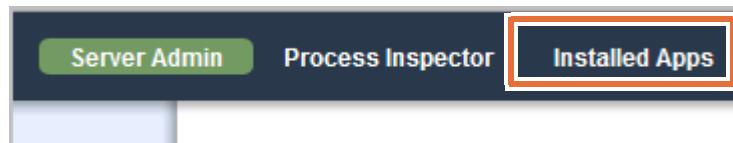
 Troubleshooting

In the interest of conserving memory, you can now choose to close IBM Integration Designer before you continue with IBM Process Designer.

- \_\_\_ 19. Switch to the IBM Process Designer window. If you closed it earlier, you can start it by double-clicking the shortcut icon on the desktop.
- \_\_\_ 20. If prompted, log in using `pcdeadmin` and `web1sphere` for the **User Name** and **Password** fields.
- \_\_\_ 21. Verify that you are in the Process Center perspective and click the **Servers** tab.
- \_\_\_ 22. Click the **Configure Server** link.



- \_\_\_ 23. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.
- \_\_\_ 24. On the next screen, accept the defaults and click **Confirm Security Exception**.
- \_\_\_ 25. Log in using `admin` and `web1sphere` for the **User Name** and **Password** fields.
- \_\_\_ 26. Click **Installed Apps** at the top.



- \_\_\_ 27. Verify that **AccountServicesApp** is listed among the deployed applications.

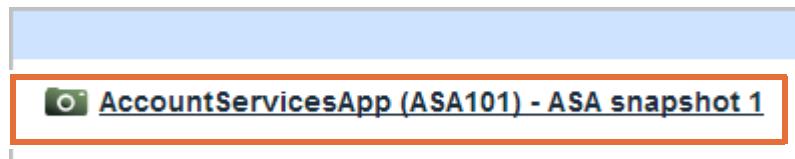
The screenshot shows the IBM Process Admin Console interface. At the top, there are tabs: Process Admin Console, Server Admin, Process Inspector, and a green-highlighted tab labeled 'Installed Apps'. Below the tabs, there is a search bar with the placeholder 'Sort Snapshots By: Application Name' and dropdown options 'All', 'Active', and 'Default'. The main area displays a list of applications:

| Application                                                  | Status          |
|--------------------------------------------------------------|-----------------|
| <a href="#">AccountServicesApp (ASA101) - ASA snapshot 1</a> | Active, Default |
| <a href="#">Heritage Process Portal (TWP) - 8.5.7.0</a>      | Active, Default |
| <a href="#">Process Portal (SYSRP) - 8.5.7.0</a>             | Active, Default |

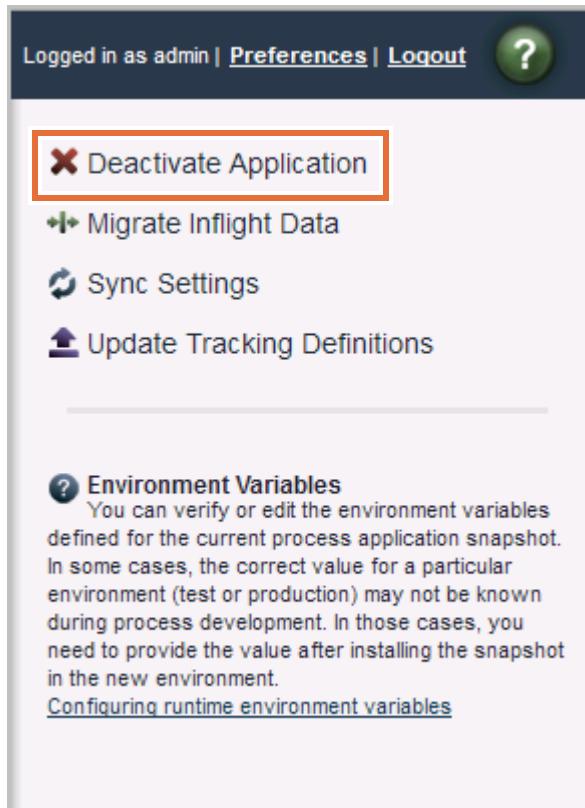
### ***Part 5: Removing the Account Services process application***

In this section, you learn how to stop and undeploy process applications so that you can control what applications are available to users.

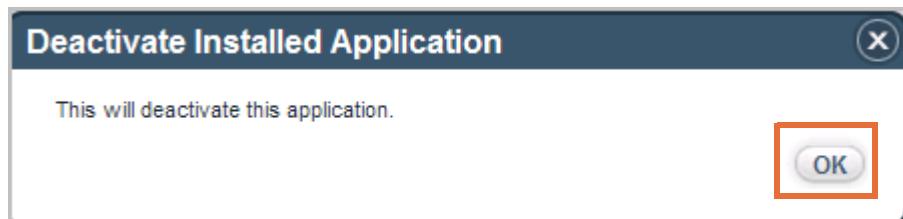
- \_\_\_ 1. In the **Installed Apps** pane of the console, click **AccountServicesApp (ASA101) - ASA snapshot 1**.



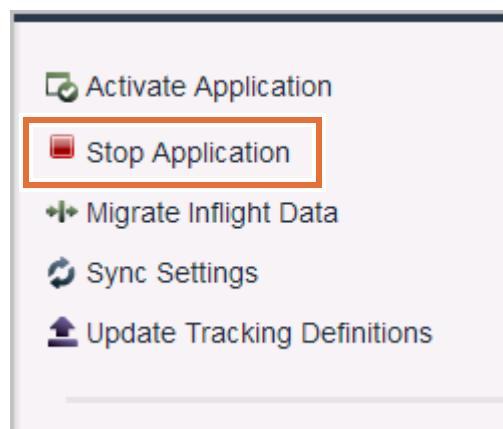
- 2. Click **Deactivate Application** in the upper-right corner.



- 3. In the Deactivate Installed Application dialog box, click **OK** to confirm the deactivation.



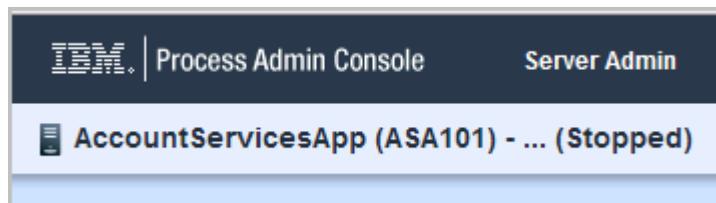
- 4. When the application is deactivated, a new **Stop Application** link is listed on the right. Click that link.



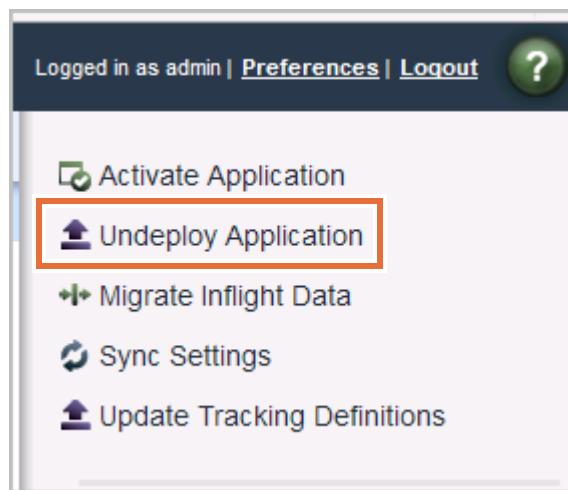
- 5. In the Stop Installed Application dialog box, click **OK**.



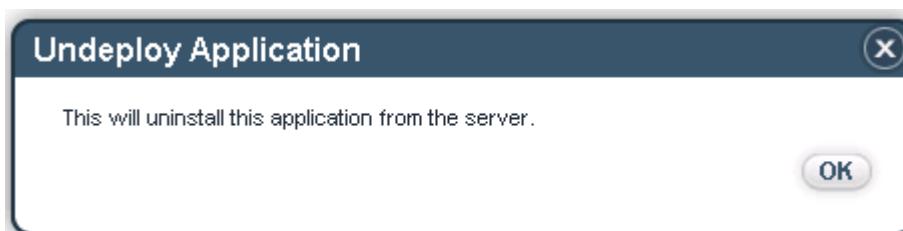
- 6. It takes a few minutes for the application to stop. When it stops, the dialog box closes and a stopped status is displayed at the top. If you receive an error that the connection is lost, close that message box and check the status of the application. If it is stopped, then you can continue, or else click the **Stop Application** link again.



- 7. When the application stops, a new **Undeploy Application** link is listed on the right. Click that link.



- 8. In the Undeploy Application dialog box, click **OK**. Do not click **OK** multiple times as it takes a few minutes for the application to uninstall, and the dialog box stays open. When the application is uninstalled, the dialog box closes.



- \_\_\_ 9. Click **All** to verify that the application has a status of **Undeployed**.

| Application                                  | Status              |
|----------------------------------------------|---------------------|
| AccountServicesApp (ASA101) - ASA snapshot 1 | Undeployed, Default |
| Heritage Process Portal (TWP) - 8.5.7.0      | Active, Default     |
| Process Portal (SYSRP) - 8.5.7.0             | Active, Default     |

- \_\_\_ 10. Log out and close the Process Admin Console.

### **Part 6: Deleting the Account Services process application snapshot**

- \_\_\_ 1. Switch to the Process Center perspective window.
- \_\_\_ 2. In the Process Apps tab, click **AccountServicesApp**.
- \_\_\_ 3. Click **Manage**.
- \_\_\_ 4. Click **Archive Process App**.
- \_\_\_ 5. Click **Archive** in the dialog box. It takes a couple of minutes until the application is archived. No further confirmation is displayed.
- \_\_\_ 6. Wait a few minutes and then click the **Process Apps** tab. If AccountServicesApp is listed, then it is still being archived. Eventually it goes away from the list. Click the **Archived** link.



- \_\_\_ 7. Only one application is listed in the Archived category. The application is now ready to be deleted. Click **Delete**.

- \_\_\_ 8. Click **Delete** again in the dialog box. The application is now permanently deleted.

- \_\_\_ 9. Close Process Designer.
- \_\_\_ 10. (Optionally) Stop the IBM Process Center server. If you are continuing to the next exercise, do not stop the server.
  - \_\_\_ a. Double-click the **Stop Process Center cluster** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
  - \_\_\_ b. Double-click the **Stop Process Center node agent** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
  - \_\_\_ c. Click the **Stop Process Center deployment manager** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
- \_\_\_ 11. (Optionally) Stop the IBM Process Server unless it is already stopped. If you are continuing to the next exercise, do not stop the server.
  - \_\_\_ a. Click **Start > All Programs > IBM > Business Process Manager Advanced Process Server 8.5 > Profiles > qbpmaps > Stop the server**.
  - \_\_\_ b. Enter `admin` and `web1sphere` for **User Identity** and **User Password** when prompted, and click **OK**. Wait until it stops successfully.
- \_\_\_ 12. Close IBM Integration Designer.
- \_\_\_ 13. Close any other open windows.

## **End of exercise**

## Exercise review and wrap-up

In this exercise, you used the IBM Process Center console to explore IBM Process Center. You deployed a process application to the Process Server test environment and then archived and deleted that application.



# Exercise 17. Implementing Advanced Integration services

## What this exercise is about

In this exercise, you create an Advanced Integration service that is used by IBM Process Designer to invoke IBM Integration Designer implementations.

## What you should be able to do

After completing this exercise, you should be able to:

- Create an AIS in IBM Process Designer
- Implement an AIS in IBM Integration Designer
- Unit test an AIS in IBM Integration Designer
- Invoke a BPEL process from a coach and BPD

## Introduction

Most IBM BPM implementations require a combination of human tasks and integrations with back-end systems. Certain integrations might be complex in nature and require the orchestration of multiple service invocations, and a higher level of transactional integrity and compensation. Such integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL). IBM Process Designer has Advanced Integration services (AIS) which can invoke those IBM Integration Designer implementations.

This exercise shows you how Process Designer and Integration Designer can work in concert to create a fully integrated end-to-end IBM Business Process Manager solution.

A business process can take advantage of integration services in two fundamental ways:

- The business process designer, by using Process Designer, can create an AIS that has the high-level details of the interface (inputs, outputs, and exceptions). The more technical integration developer then completes the implementation in IBM Integration Designer. This approach is a top-down approach.
- Conversely, integration developers can create a number of integration services ahead of time. Then, the process developer can discover them and reuse them later on, for example, by packaging them in a Toolkit. This approach is called the bottom-up approach.

This exercise captures the steps to create a simple Advanced Integration service in IBM Business Process Manager by using the top-down approach.

## Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Portal, IBM Process Center, and the IBM Process Server test environment. Instructions for creating the lab environment are included in the exercise appendixes.

## Exercise instructions

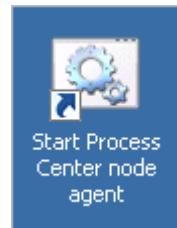
### Part 1: Start the IBM Process Center environment

If the Process Center environment is running from the previous exercise, then go to Part 2: Create an AIS by using IBM Process Designer. Otherwise, complete the following steps to start the environment:

- 1. On your Windows desktop in your lab environment, select the **Start Process Center deployment manager** shortcut. Double-click the shortcut or press Enter to start the server.



- a. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.
- 2. On your Windows desktop, select the **Start Process Center node agent** shortcut. Double-click the shortcut or press Enter to start the server.

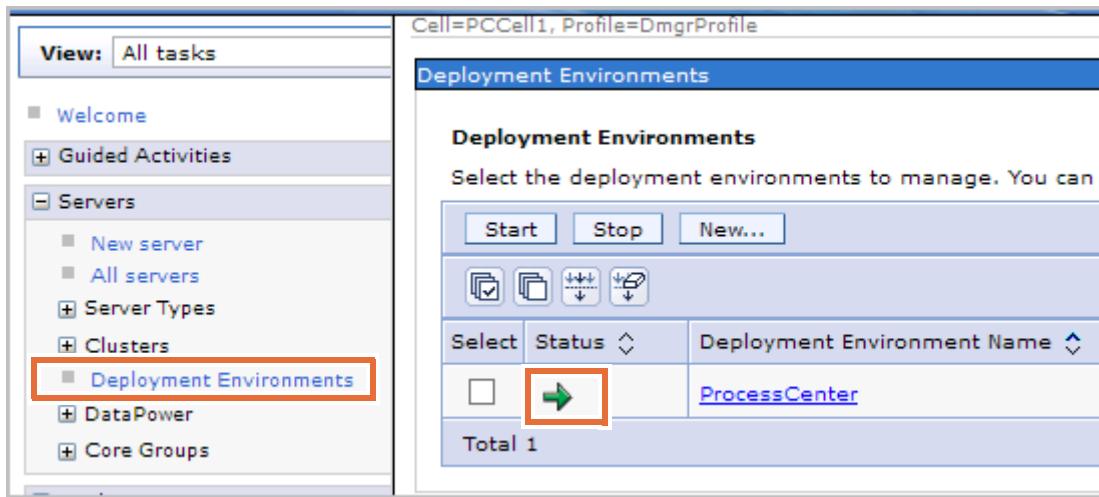


- a. A DOS command window is displayed; press any key to continue when prompted.
- 3. Start the process center single cluster.
- a. On your Windows desktop, select the **Start Process Center Cluster** shortcut. Double-click the shortcut or press Enter to start the cluster member.



- b. A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts. Press any key to continue when prompted.
- 4. Verify that the Process Center Single Cluster is started.
- a. On your Windows desktop, select the **Deployment Manager Administrative Console** shortcut. Double-click the shortcut or press Enter to go to the console.

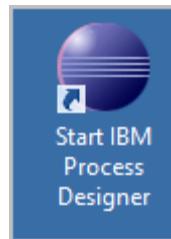
- \_\_ b. Log in by entering `bpmadmin` in the **User ID** field and `websphere` in the **Password** field.
- \_\_ c. Select **Servers > Deployment Environments**.



- \_\_ d. Verify that the deployment environment is started. When it starts, the status turns green. It might be necessary to refresh the browser or click the **Deployment Environments** link to view the status change.
- \_\_ e. Click **Servers > Clusters > WebSphere application server clusters** to verify that the single cluster is running.
- \_\_ f. Click **Logout** to exit.
- \_\_ g. Close the browser.

## **Part 2: Create an Advanced Integration service (AIS) by using IBM Process Designer**

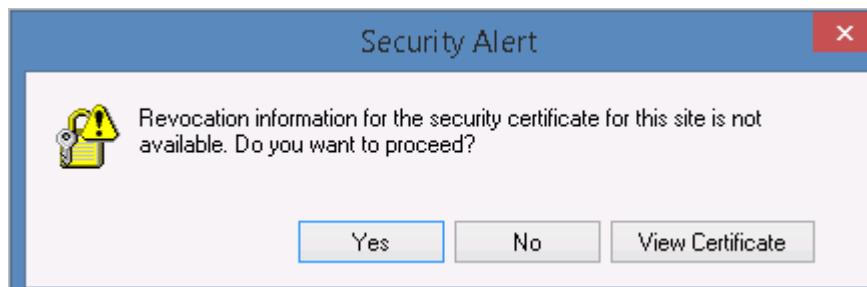
- \_\_ 1. Start the desktop IBM Process Designer.
  - \_\_ a. Double-click the **Start IBM Process Designer** icon on the desktop.



- \_\_\_ b. Enter `pcdeadmin` in the **User name** field and `web1sphere` in the **Password** field, and click **Log In**.



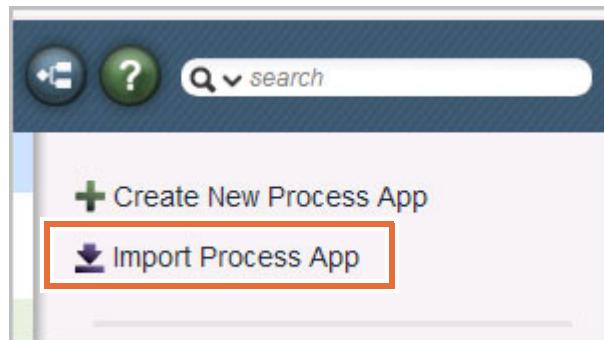
- \_\_\_ c. When the **Security Alert** dialog box is displayed, click **Yes** to proceed.



- \_\_ d. After a few moments, click **Yes** one more time.

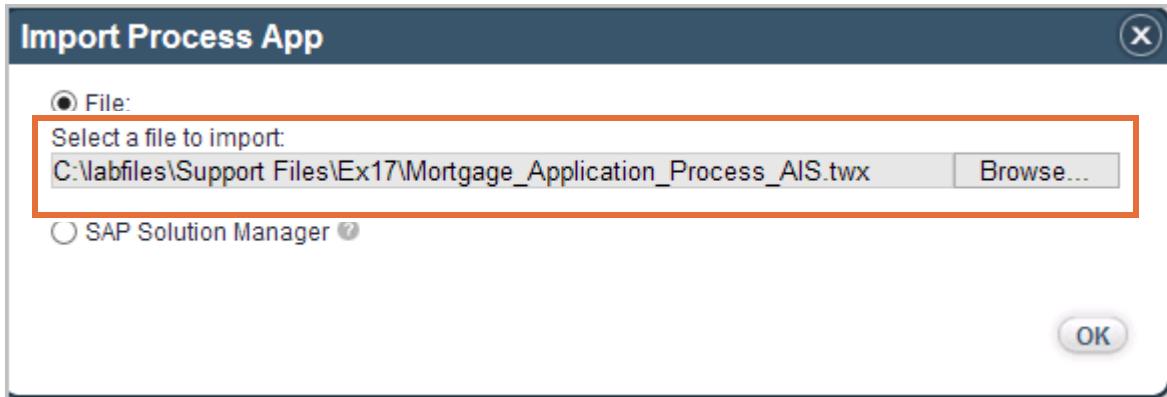


- \_\_ e. Make sure that you are in the Process Center view. If you are in the Designer view, then click the Process Center icon at the upper right of the Designer view.
- \_\_ 2. Import a Mortgage Application Process application in the Process Center. After importing this process app, you then augment it with a new Advanced Integration service.
- \_\_ a. In the Process Apps tab, click **Import Process App** on the right.

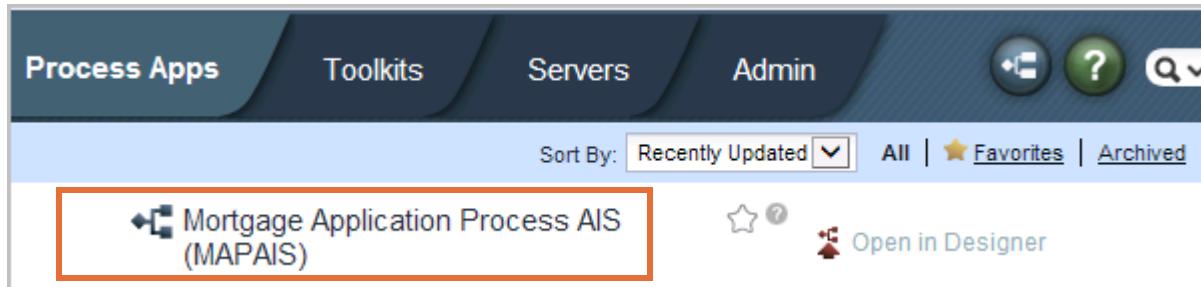


- \_\_ b. In the Import Process App window, click **Browse**.
- \_\_ c. Go to C:\labfiles\Support Files\Ex17 and select **Mortgage\_Application\_Process\_AIS.twx**.
- \_\_ d. Click **Open**.

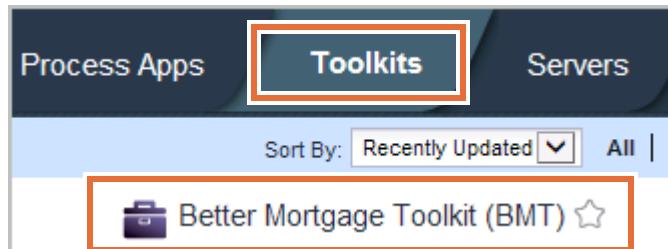
- \_\_ e. Verify that the selected application is listed in the **Select the file to import** field, and click **OK**.



- \_\_ f. In the Import Process App window, click **Import**. It takes few minutes for the process app to import.
- \_\_ g. Verify that the process app was imported successfully by making sure that Mortgage Application Process AIS is listed in the Process Apps tab.



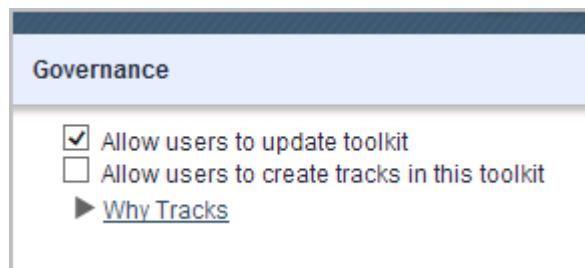
- \_\_ h. Switch to the **Toolkits** tab and click **Better Mortgage Toolkit**.



- \_\_ i. For the Integration Designer to access the toolkit that is used by the process app you are working with in this exercise, you need to make sure that it can be updated. Click **Manage** and then select the **Allow users to update toolkit** check box.

The screenshot shows the 'Toolkits' tab selected in the top navigation bar. Below it, the 'Better Mortgage Toolkit (BMT)' is listed. The 'Manage' tab is active. On the right side, there is a section labeled 'Allow users to update toolkit' with an unchecked checkbox. Other fields shown include 'Toolkit Name: Better Mortgage Toolkit', 'Acronym: BMT', and 'Documentation'.

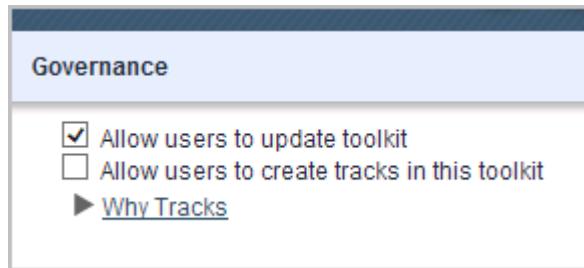
- \_\_ j. After selection, verify that the view looks similar to the following image:



- \_\_ k. Click the Toolkits tab, and then click **Color TK**. This toolkit is another that the process app uses.
- \_\_ l. For the Integration Designer to access the Toolkit that is used by the process app you are working with in this exercise, you need to make sure that it can be updated. Click **Manage** and then select the **Allow users to update toolkit** check box.

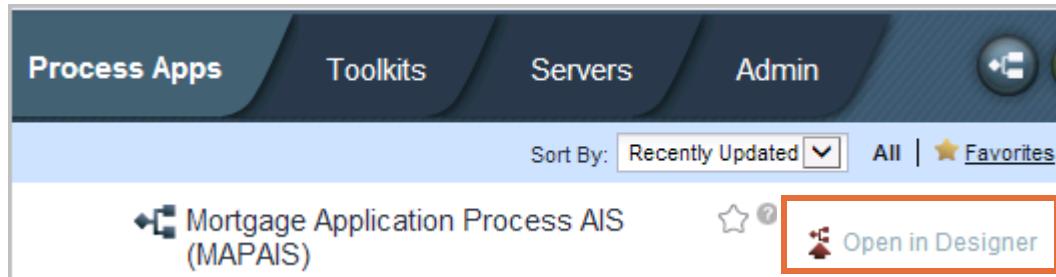
The screenshot shows the 'Toolkits' tab selected in the top navigation bar. Below it, the 'Color.TK (CTK)' toolkit is listed. The 'Manage' tab is active. On the right side, there is a section labeled 'Allow users to update toolkit' with an unchecked checkbox. Other fields shown include 'Toolkit Name: Color.TK', 'Acronym: CTK', and 'Documentation'.

- \_\_\_ m. After selection, verify that the view looks similar to the following image:

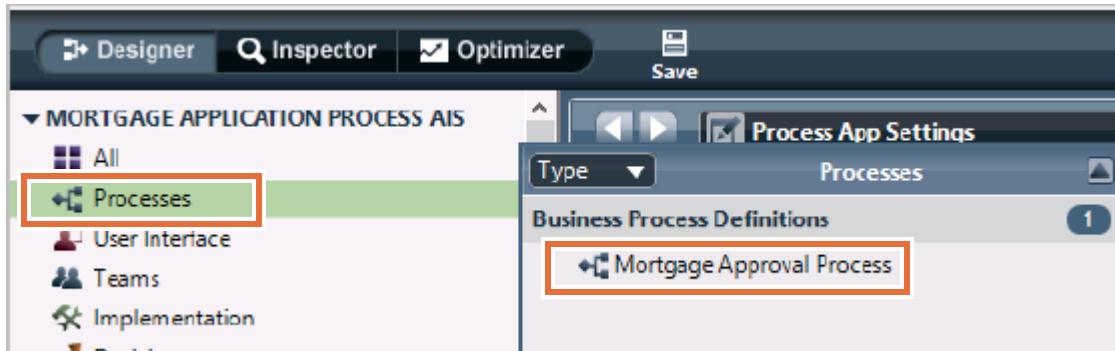


- \_\_\_ 3. Create an AIS in the business process.

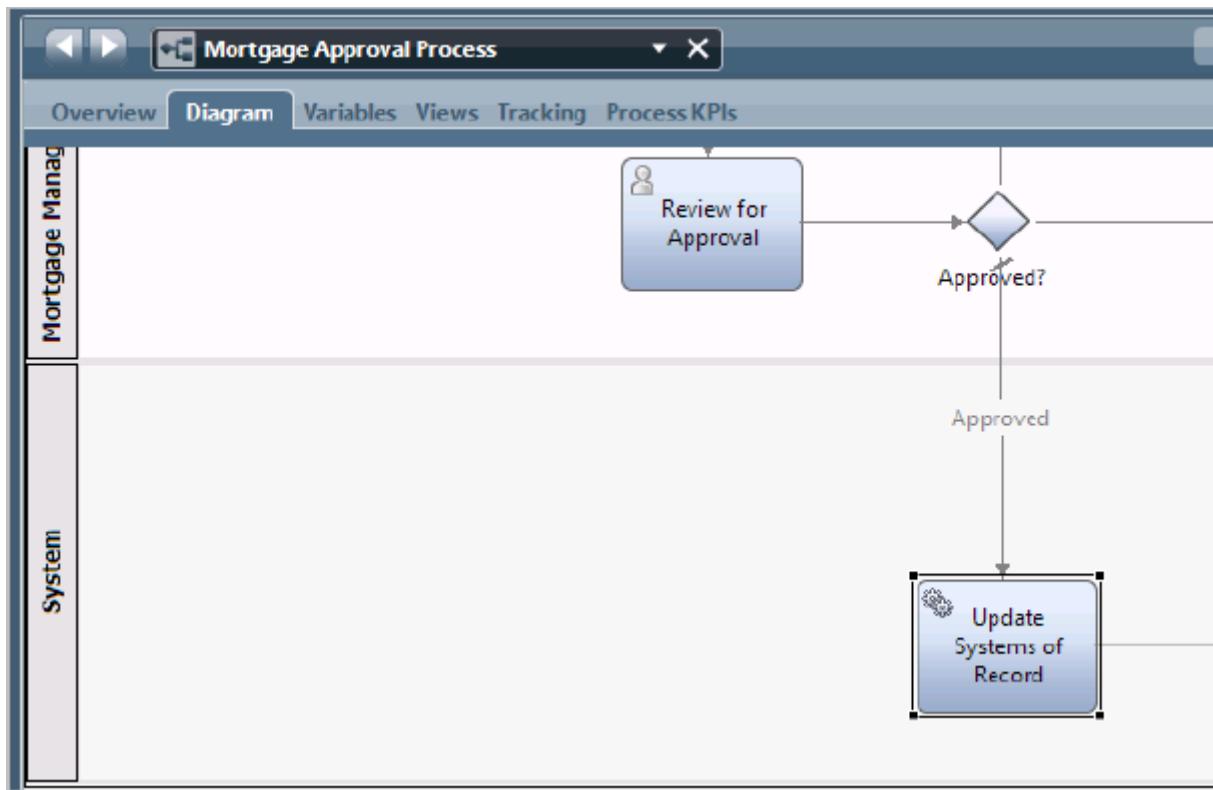
- \_\_\_ a. Switch to the Process Apps tab and then click **Open in Designer** to the right of Mortgage Application Process AIS.



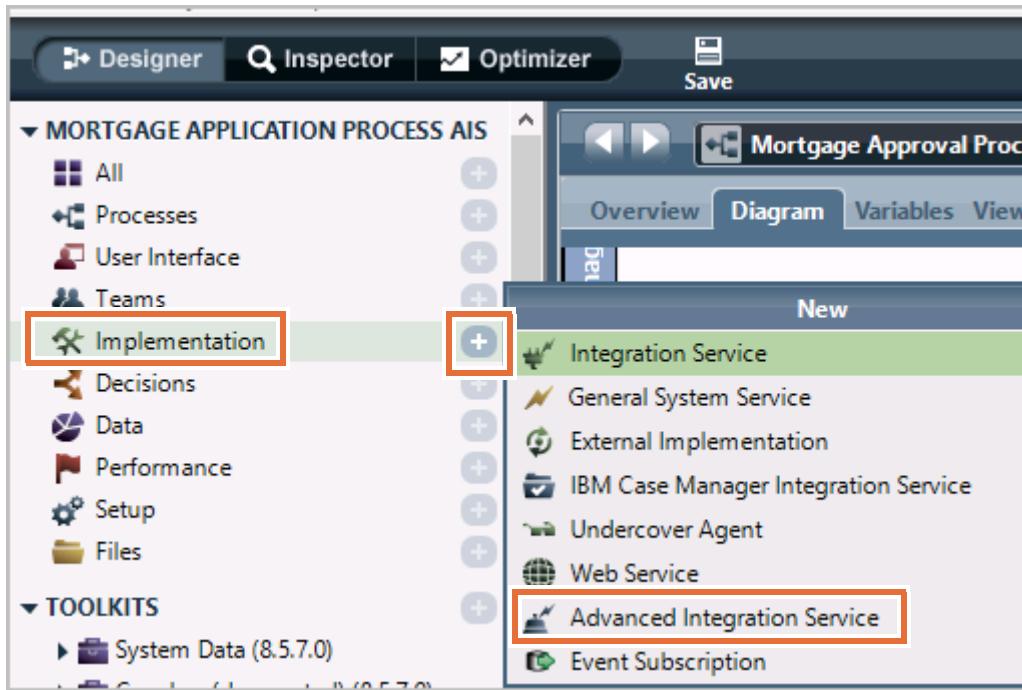
- \_\_\_ b. In the process library, click **Processes** and then double-click the **Mortgage Approval Process** under Business Process Definitions.



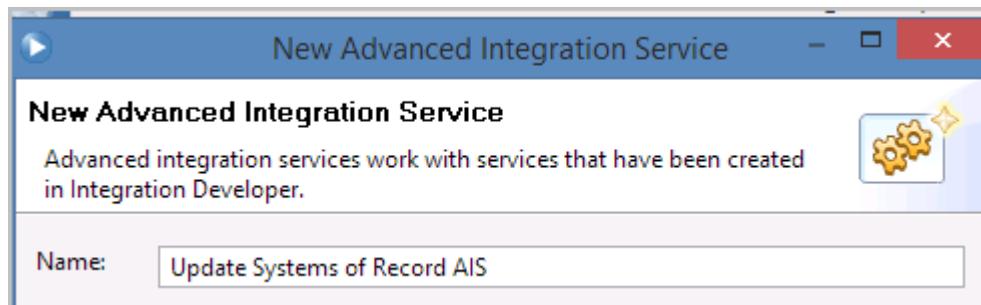
- \_\_\_ c. The process diagram opens in the main editor pane. Scroll down to the System lane. In the next step, you add an Advanced Integration service to replace the current **Update Systems of Record** Activity with an Advanced Integration service.



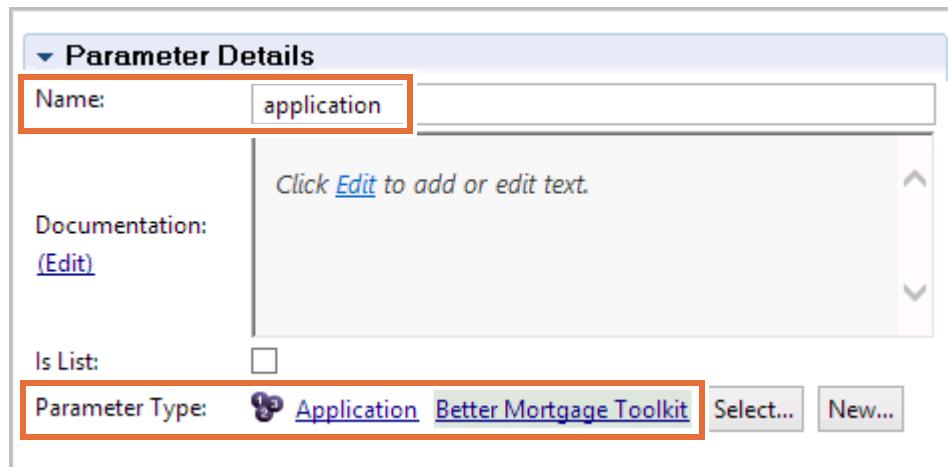
- \_\_\_ d. In the process library, next to Implementation click the + icon. Then, in the New window, click **Advanced Integration Service**.



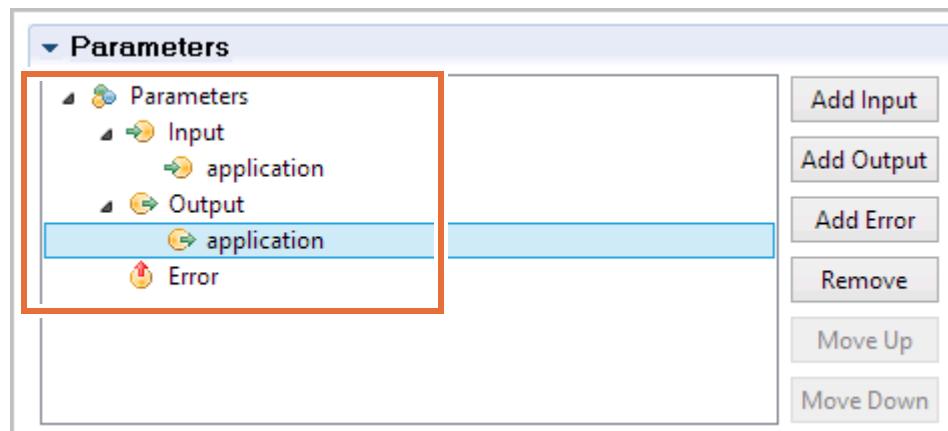
- \_\_\_ e. In the **New Advanced Integration Service** window, enter `Update Systems of Record AIS` for the Name and then click **Finish**.



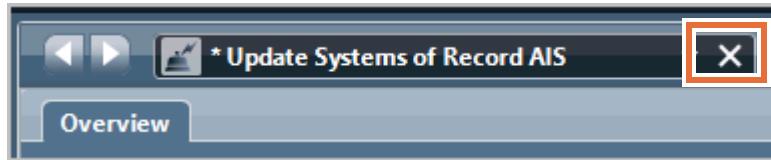
- \_\_\_ f. The Advanced Integration service editor opens. You define the input and output of this new service. In the **Parameters** section, click **Add Input**.
- \_\_\_ g. In the **Parameter Details** section, enter `application` for the **Name** and then set the **Parameter Type** to `Application`.



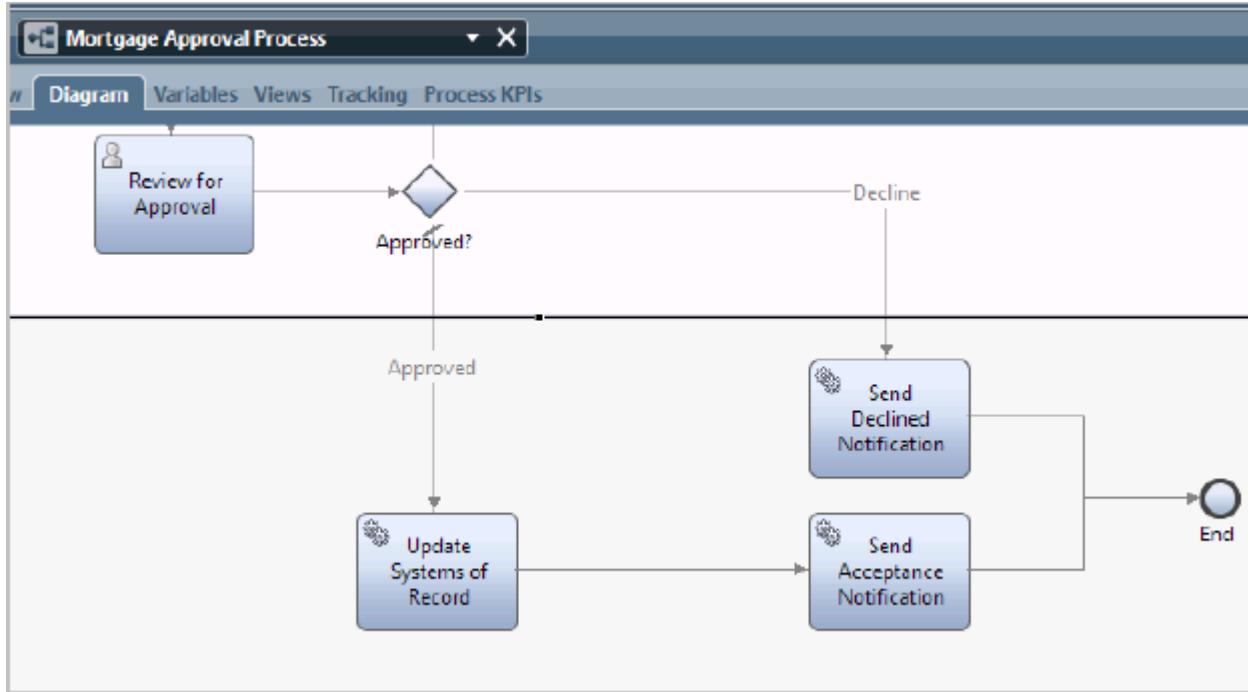
- \_\_\_ h. Add an output parameter by clicking **Add Output** in the **Parameters** section. In the **Parameter Details** section, enter `application` for the **Name** and then set the **Parameter Type** to `Application`.
- \_\_\_ i. The Parameters section looks similar to the following image:



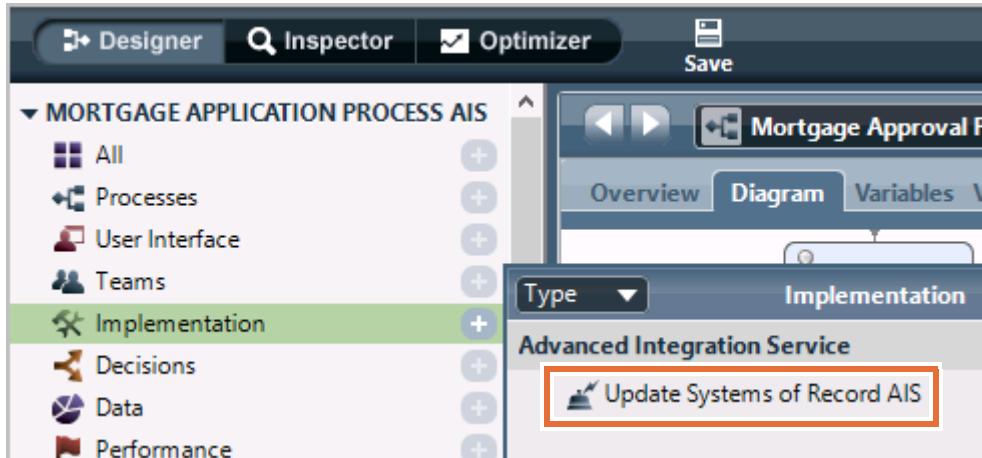
- \_\_ j. Save (Ctrl-S) your work and close the Advanced Integration service editor by clicking X.



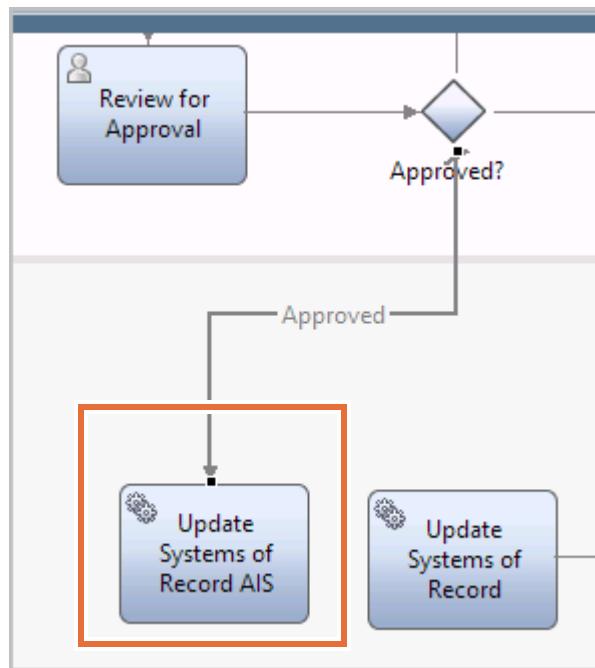
- \_\_ k. You now replace the **Activity Update Systems of Record** in the process with the recently created **Update Systems of Record AIS** Advanced Integration service. Make sure that the Mortgage Approval Process is displayed in the Diagram tab.



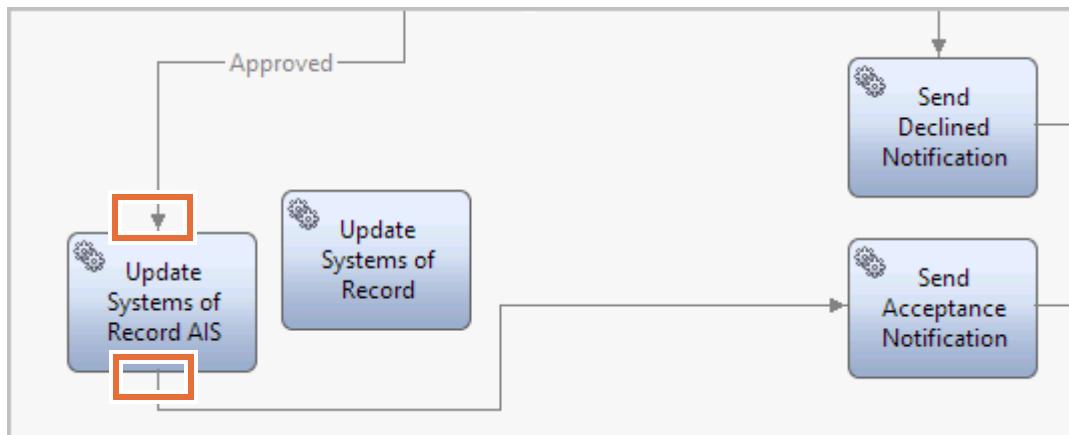
- \_\_ l. In the process library, click Implementation, and under **Advanced Integration Service** drag **Update Systems of Record AIS** to the System swimlane next to the existing **Update Systems of Record** activity.



- \_\_\_ m. Click the tip of the sequence flow arrow that goes into **Update Systems of Record**, and drag it to the input of **Update Systems of Record AIS**.

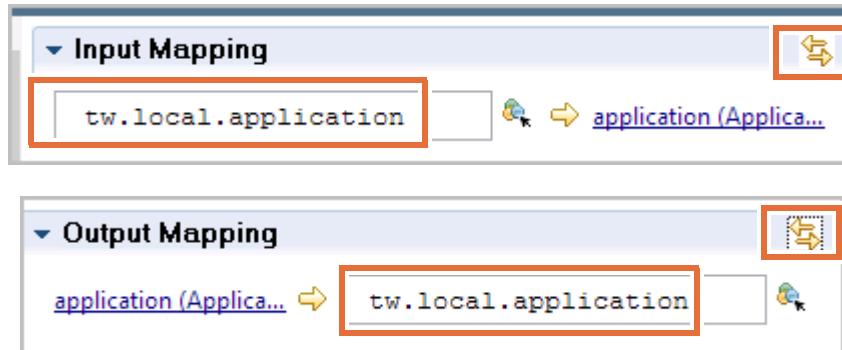


- \_\_\_ n. Drag the starting end of the arrow that flows out of **Update Systems of Record** to the corresponding spot in **Update Systems of Record AIS**. The diagram looks similar to the following image:

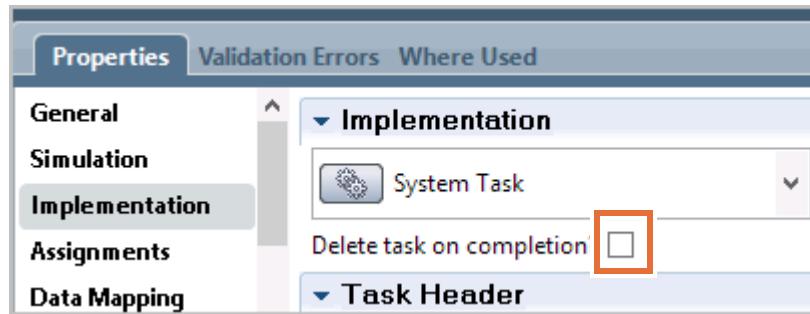


- \_\_\_ o. Right-click **Update Systems of Record** then click **Edit > Delete**.  
 \_\_\_ p. Select **Update Systems of Record AIS** and click the **Data Mapping** tab.

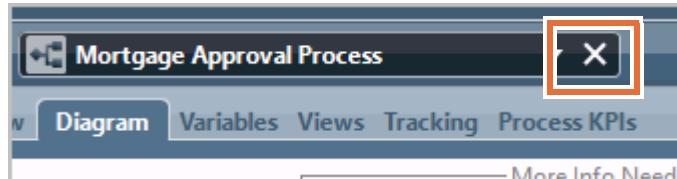
- \_\_\_ q. Click the auto mapping icons for both input and output mapping. The two fields are automatically populated with tw.local.application:



- \_\_\_ r. Click Implementation and clear Delete task on completion check box.



- \_\_\_ s. Save your work (Ctrl-S).  
\_\_\_ t. Close Mortgage Approval Process by clicking X.



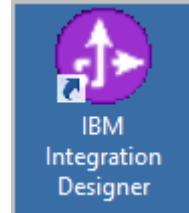
- \_\_\_ u. Click Process Center at the upper right corner to switch to the Process Center view.  
\_\_\_ v. Minimize IBM Process Designer.

Your work as a process author is done for now. You created the definition of an Advanced Integration service, which a more technical user implements by using the IBM Integration Designer.

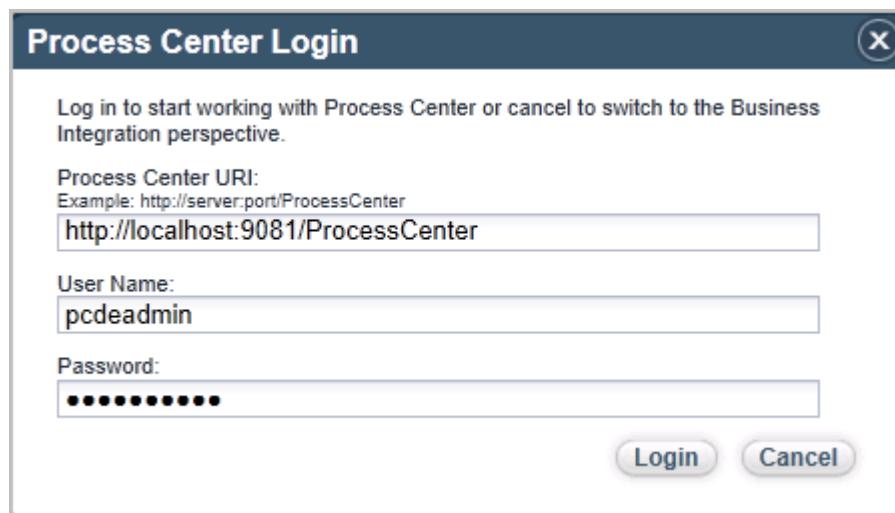
### Part 3: Implement the AIS in IBM Integration Designer

In this section, you take the unimplemented Advanced Integration service and complete its implementation in IBM Integration Designer.

- 1. Create the AIS workspace.
  - a. On the Desktop, double-click IBM Integration Designer.

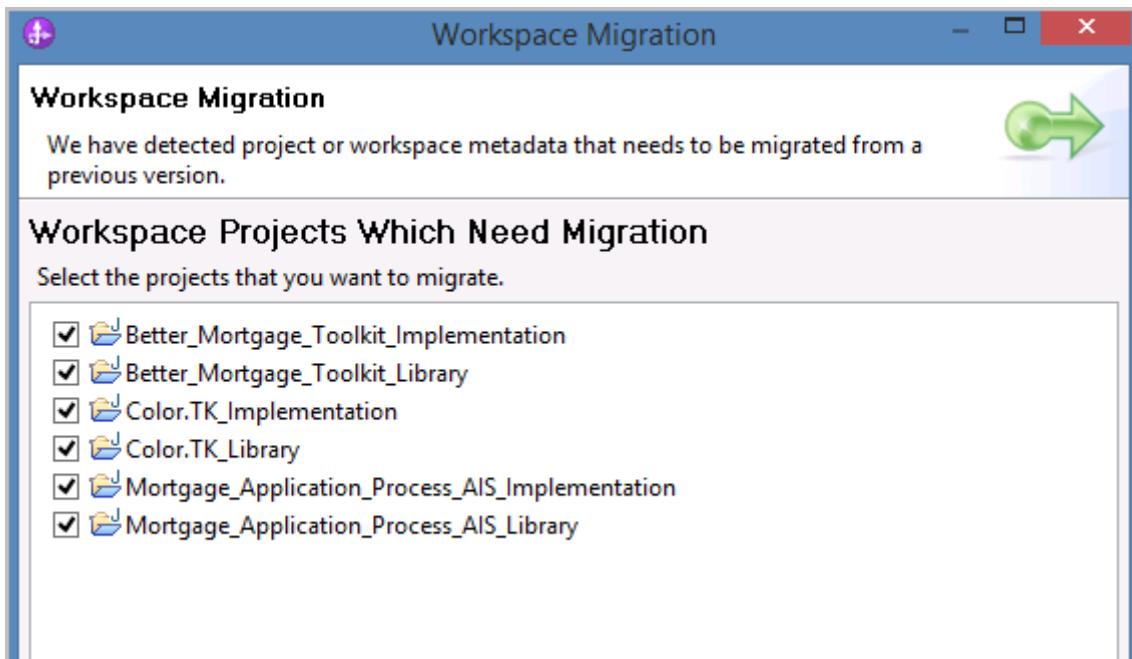


- b. In the Workspace Launcher window, enter C:\Workspaces\AIS in the **Workspace** field and click **OK**.
- c. In the Process Center Login window, enter the following credentials:
  - **Process Center URI:** `http://localhost:9081/ProcessCenter`
  - **User Name:** `pcdeadmin`
  - **Password:** `web1sphere`

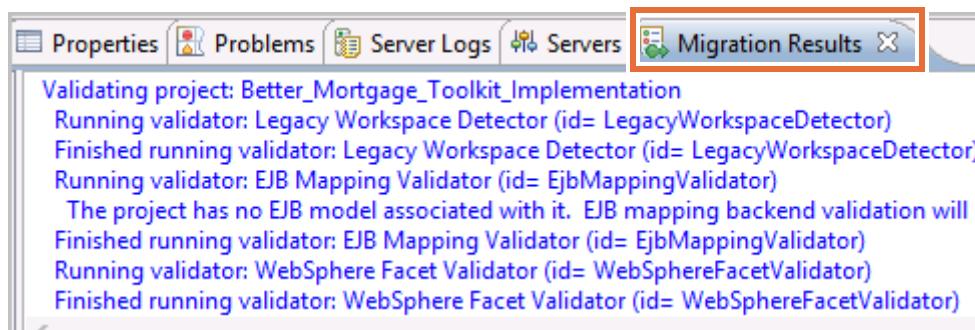


- d. Click **Login**.
- e. When the Secure Storage window opens, click **Cancel**.
- f. If a Security Alert window is displayed asking to proceed, then click **Yes** each time.
- g. If the Secure Storage window opens again, click **Cancel**.
- h. When the Secure Storage Warning window opens, click **OK**.
- i. Close the "Getting Started with IBM Process Center 8.5.7.0" welcome screen, by clicking the **X** at the upper right corner of the window.
- j. Next to **Mortgage Application Process AIS (MAPAIS)**, click **Open in workspace**.

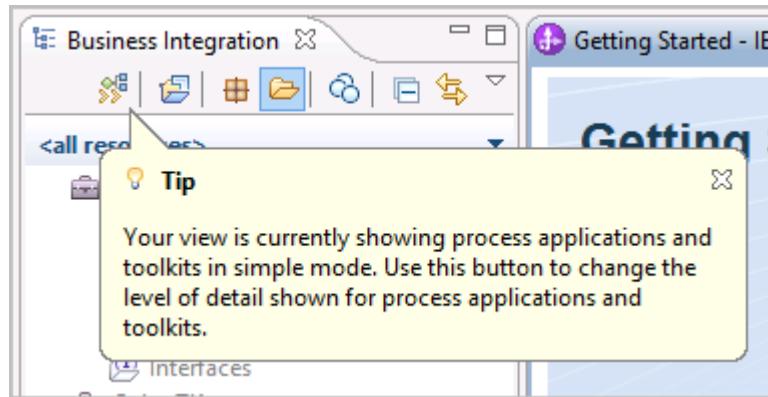
- \_\_ k. In the **Workspace Migration** window make sure that all the projects are selected and click **Next**.



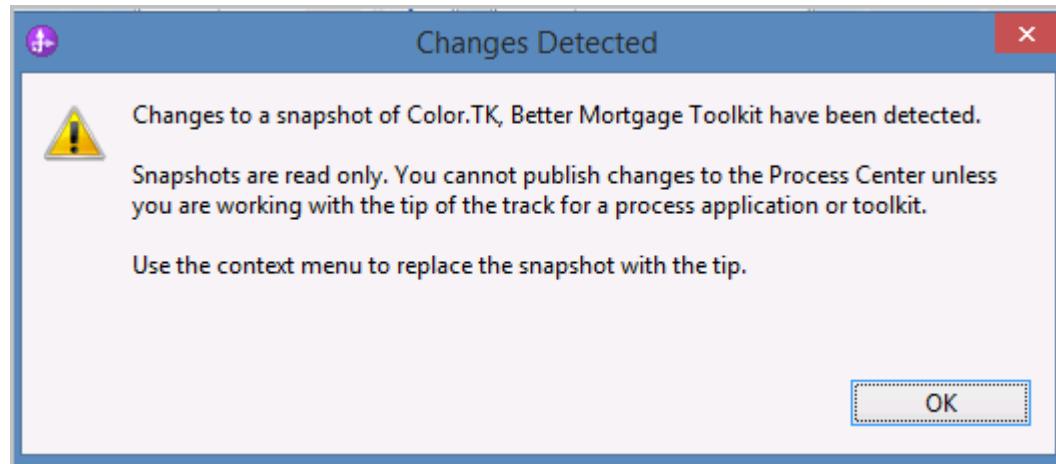
- \_\_ l. Click **Next** one more time.
- \_\_ m. Click **Finish** to complete the migration.
- \_\_ n. When the **Migration Validation** window is displayed confirming that the migration validation completed successfully, click **OK**.
- \_\_ o. Close the Migration Results view.



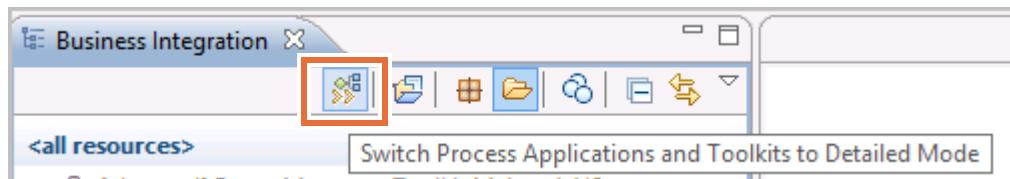
- \_\_\_ p. If the Tip window is displayed, close it by clicking X.



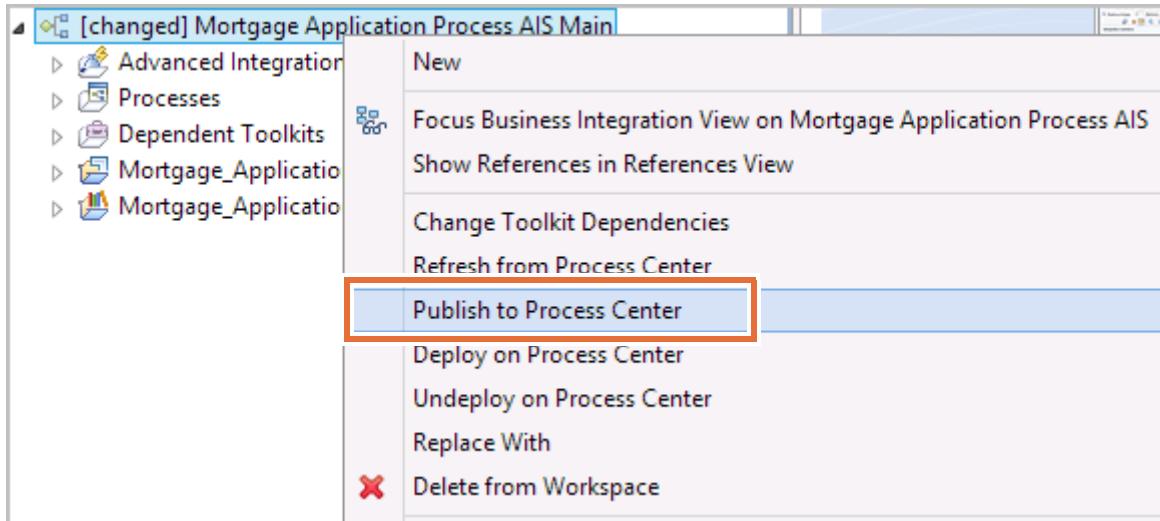
- \_\_\_ q. Close the **Getting Started** tab.  
\_\_\_ r. Wait until the workspace is built. It takes few minutes for the project to build. A progress bar is on the lower right corner of the Integration Designer.  
\_\_\_ s. If a window is displayed confirming that changes are detected, then click OK to close that window.



- \_\_\_ t. Switch to the detail mode by clicking the **Switch process Applications and Toolkits to Detailed Mode** icon.

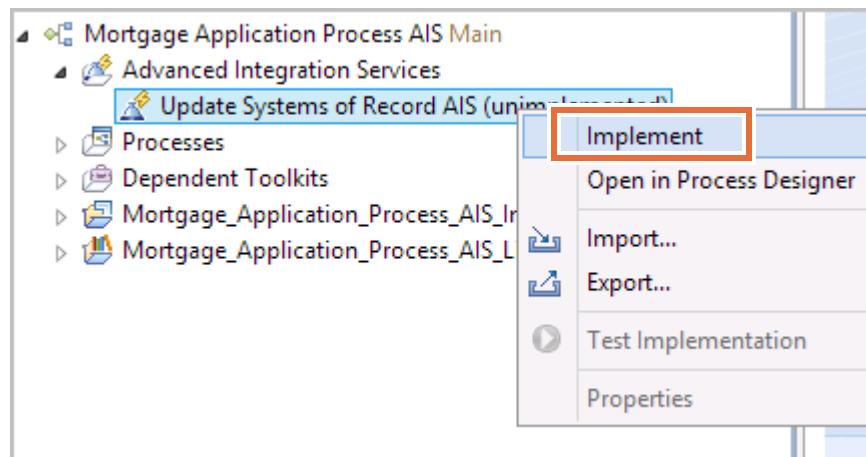


- \_\_ u. Notice that Mortgage Approval Process has [changed] next to it. Right-click [changed] Mortgage Approval Process Main, and click **Publish to Process Center**.

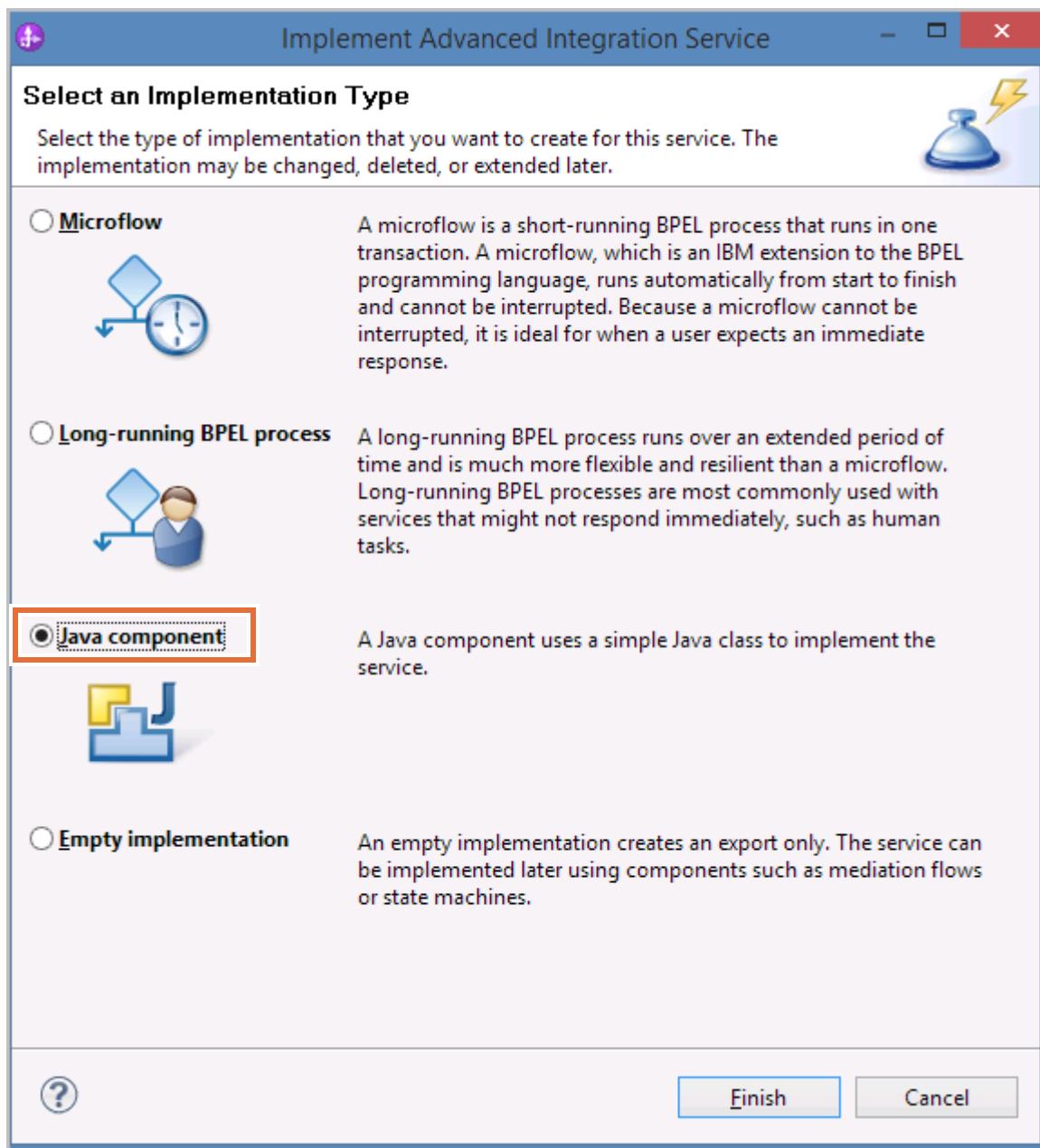


- \_\_ 2. Implement the **Update Systems of Record AIS** service.

- \_\_ a. In the project tree of the Business Integration perspective, expand Advanced Integration Services under Mortgage Application Process. AIS Main. Notice that **Update Systems of Record AIS** is labeled as unimplemented. Right-click it and click **Implement**.



- \_\_ b. In the Implement Advanced Integration Service window, select **Java component**, and click **Finish**. This action opens a Java editor with predefined functions and methods.



- \_\_\_ c. In the Java editor, scroll down to the bottom, and locate the invoke method.

```
public DataObject invoke(DataObject application) {
 // To create a DataObject, use the creation methods
 // com.ibm.websphere.bo.BOFactory boFactory = (com.
 //
 // To get or set attributes for a DataObject such
 // To set a string attribute in application, use
 // To get a string attribute in application, use
 // To set a dataObject attribute in application,
 // To get a dataObject attribute in application,
 return null;
}
```

- \_\_\_ d. You write some logic that sets the mortgage application status to “RECORDED FROM AIS” and print some information to the log files, for initial debugging purposes. Delete return null at the end of the invoke method.
- \_\_\_ e. Open Windows Explorer and browse to: C:\labfiles\Support Files\Ex17
- \_\_\_ f. Open AIS\_Recorded.txt in a text editor such as Notepad.
- \_\_\_ g. Copy the text in AIS\_Recorded.txt and paste the content in the Invoke method.

```
//-----
// Invoke
//-----
application.setString("status", "RECORDED FROM AIS");
System.out.println("*****<<Advanced Integration Service Java
Implementation successfully executed>>*****");
return application;
```

- \_\_\_ h. Alternatively, you can manually replace the code with the following text:

```
application.setString("status", "RECORDED FROM AIS");
System.out.println("*****<<Advanced Integration Service Java
Implementation successfully executed>>*****");
return application;
```

- i. Remember to delete **return null** from the invoke method.

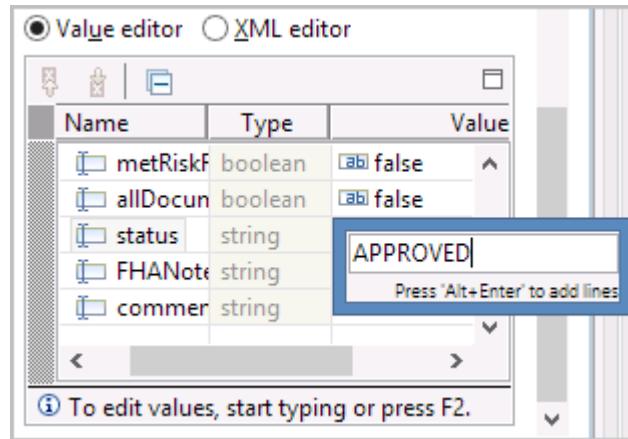
```
public DataObject invoke(DataObject application) {
 // To create a DataObject, use the creation methods on the
 // com.ibm.websphere.bo.BOFactory boFactory = (com.ibm.web
 //
 // To get or set attributes for a DataObject such as appli
 // To set a string attribute in application, use applicati
 // To get a string attribute in application, use applicati
 // To set a dataObject attribute in application, use appli
 //
 // application.setString("status","RECORDED FROM AIS");
 System.out.println("*****<<Advanced Integration Service
 return application;
```

- Save your changes and close the Java editor. You should have no errors.

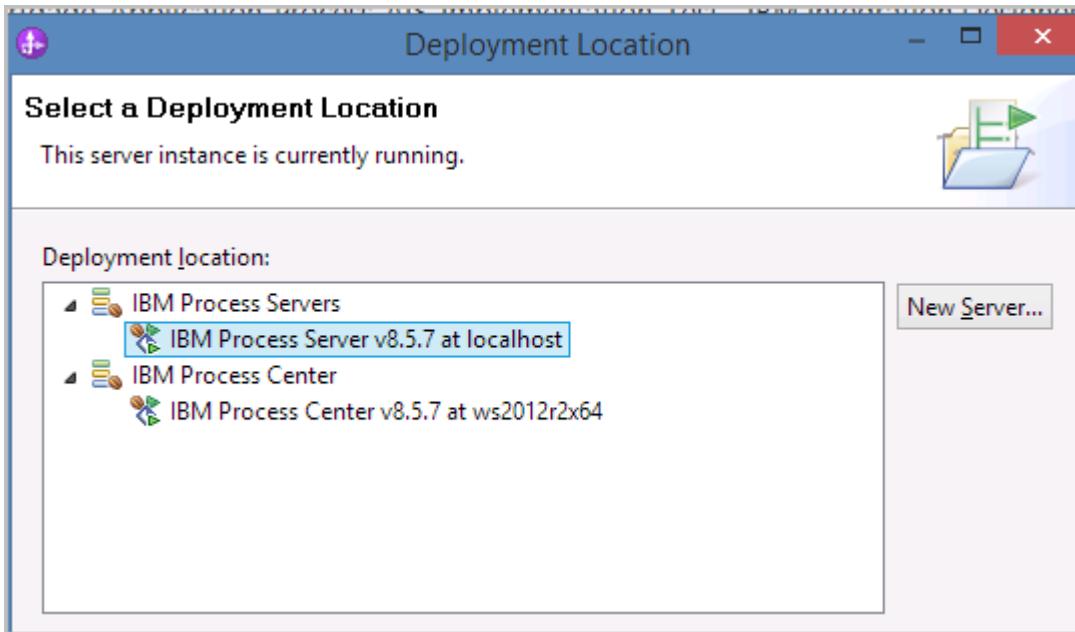
## **Part 4: Unit Test the Update Systems of Record AIS Advanced Integration service in IBM Integration Designer**

- 1. Publish your changes to Process Center.
    - a. Verify that Mortgage Approval Process has [changed] next to it.
    - b. Right-click [changed] Mortgage Approval Process Main, and click **Publish to Process Center**.
  - 2. Start the UTE server (if it is not already running).
    - a. If necessary, in the Servers view, right-click IBM Process Server v8.5.7 at localhost, and click Start. Wait for the startup process to complete before continuing.
    - b. The server is started when the message Server server1 open for e-business is displayed in the Server Logs view. The server status also changes to Started in the Servers view.
    - c. If the UTE server fails to start, then click **Start the server** icon again.
  - 3. Run the AIS test.
    - a. In Business Integration, expand **Mortgage Application Process AIS Main > Advanced Integration Services**, then right-click **Update Systems of Record AIS**, and select **Test Implementation**.
    - b. The Integration Test Client starts. In the Initial request parameters to the right, scroll through the value editor and locate the **status** field under the **Name** column.

- \_\_\_ c. Enter APPROVED in the **Value** column for status.

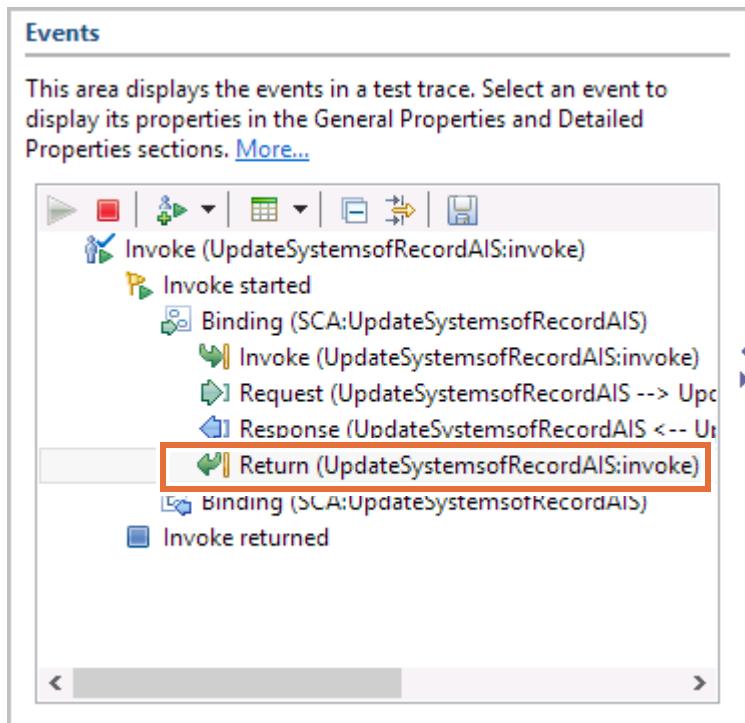


- \_\_\_ d. Under Events, click the Continue icon to start the Test.  
\_\_\_ e. In the Deployment Location window, expand IBM Process Servers and select **IBM Process Server v8.5.7 at localhost**. Click **Finish**.

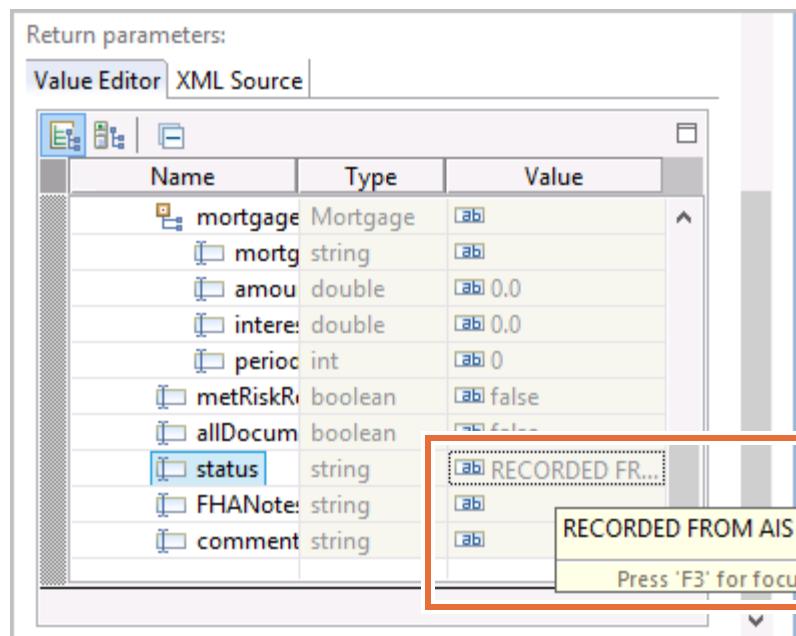


- \_\_\_ f. In the **User Login - Default Module Test** window, accept the default values and click **OK**.

- g. It takes few minutes after which the project is deployed to the test server and runs the AIS implementation. You see a trace of all the components that are being run, and as a result. In the Events section, click **Return**.



- h. In the return parameters section to the right, notice the value of **status**. It was earlier set to APPROVED, and it is now updated to RECORDED FROM AIS.

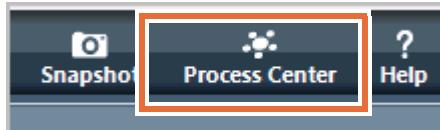


- i. You did AIS unit testing successfully. Next, you use it in the BPD in IBM Process Designer.

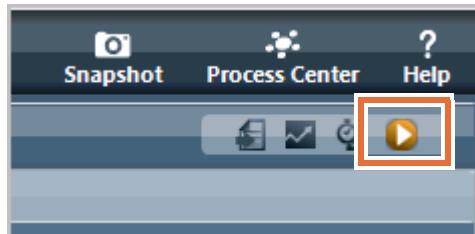
## Part 5: Run the BPD in IBM Process Designer and execute the AIS

In the previous section, you tested and confirmed that the Advanced Integration service is working as expected. In this section, you take the completed AIS and execute from the BPD. You use the Process Inspector to walk through the business process and execute the Advanced Integration service.

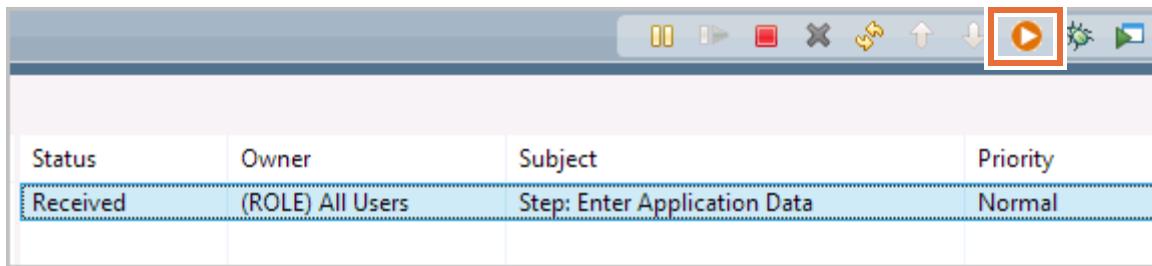
- 1. Test the end-to-end business process (including the AIS) in Process Designer.
  - a. Minimize Integration Designer, and switch back to IBM Process Designer.
  - b. If needed, go to the Process Center view by clicking the Process Center icon at the upper right.



- c. In the **Process Apps** tab, click **Open in Designer** next to Mortgage Application Process AIS.
- d. Click Processes and then double-click Mortgage Approval Process.
- e. Click the Run Process icon in the upper right corner to run the Mortgage Approval Process:

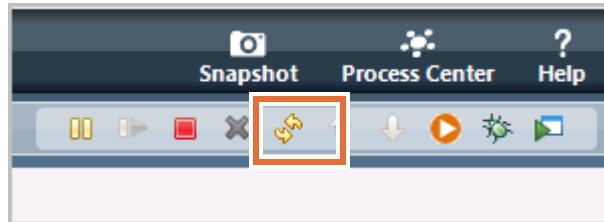


- f. If prompted, click **Yes** to switch to the Inspector view to see the progress of the process instance
- g. The Inspector opens and the focus is on the running process instance. Select the task on the upper right pane (Enter Application Data), and click **Runs the selected task**.



- h. In the **Select User** window, select **pcdeadmin** and click **OK**.
- i. If prompted with an untrusted connection, click **I Understand the Risks** and then click **Add Exception**. Click **Confirm security exception**.

- \_\_ j. After few minutes, the browser opens with the first coach of the mortgage approval process. All fields are optional for this exercise and you can leave them empty. Click **Submit** to complete the task.
- \_\_ k. Close the browser window.
- \_\_ l. Go back to Process Designer and click the refresh icon.



- \_\_ m. Find the second step on the stack (Assess Risk). Select it and click **Runs the selected task**. If you do not see it listed after some time, then click the instance on the left.

| Status   | Owner            | Subject                      | Priority |
|----------|------------------|------------------------------|----------|
| Closed   | pcdeadmin        | Step: Enter Application Data | Normal   |
| Received | (ROLE) All Users | Step: Assess Risk            | Normal   |

- \_\_ n. In the **Select User** window, select **pcdeadmin** and click **OK**.
- \_\_ o. When the coach opens, click **Submit** and then close the browser window.
- \_\_ p. Refresh the process state again as you did before. A third task is **Review for Approval**.
- \_\_ q. Select it and click **Runs the selected task**.

| Status   | Owner            | Subject                      | Priority |
|----------|------------------|------------------------------|----------|
| Closed   | pcdeadmin        | Step: Enter Application Data | Normal   |
| Closed   | pcdeadmin        | Step: Assess Risk            | Normal   |
| Received | (ROLE) All Users | Step: Review for Approval    | Normal   |

- \_\_ r. In the **Select User** window, select **pcdeadmin** and click **OK**.

- \_\_ s. In the coach, this time, make sure that you switch to the Final Decision tab and enter ACCEPT.

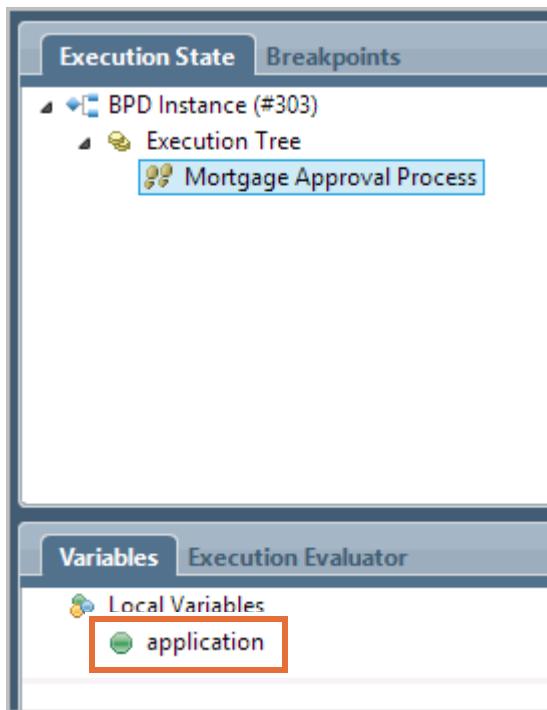
The screenshot shows a web-based application titled "Manager - Enter final application decision below". At the top, there are five tabs: Customer Information, Qualifying Information, Mortgage Information, Underwriting, and Final Decision. The "Final Decision" tab is currently active. Below the tabs, a section titled "Enter final application decision (ACCEPT, DENY, MORE INFO)" contains two buttons: "ACCEPT" and "MORE INFO". The "ACCEPT" button is highlighted with a red rectangular border.

- \_\_ t. Click **Submit** and close the browser.
- \_\_ u. In the Inspector, click the refresh icon a few times until the process completes. The stack should look similar to the one that follows (if you clicked refresh several times and the status of all the tasks is closed, it is OK if your screen does not match exactly):

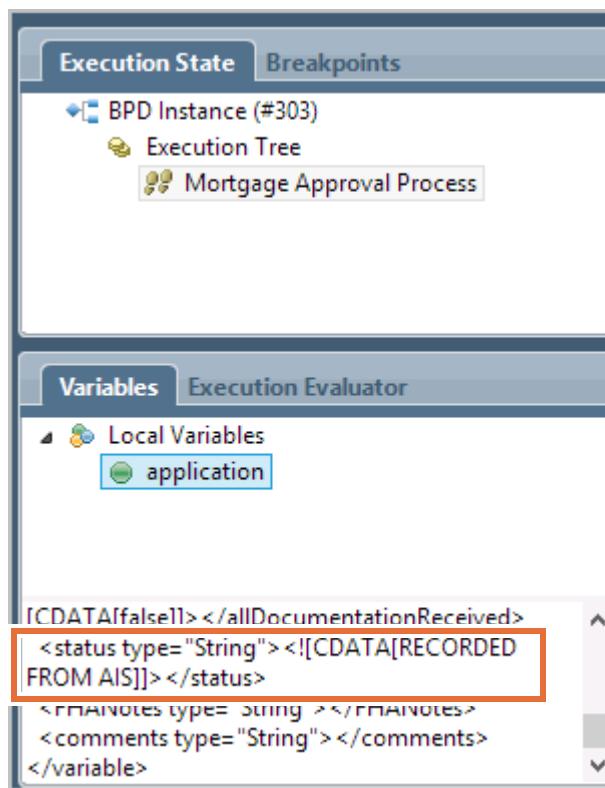
The screenshot shows a task list in an Inspector window. The window has a toolbar at the top with various icons. The main area displays a table with four rows of task information:

| Status | Owner     | Subject                            | Priority |
|--------|-----------|------------------------------------|----------|
| Closed | pcdeadmin | Step: Enter Application Data       | Normal   |
| Closed | pcdeadmin | Step: Assess Risk                  | Normal   |
| Closed | pcdeadmin | Step: Review for Approval          | Normal   |
| Closed | pcdeadmin | Step: Send Acceptance Notification | Normal   |

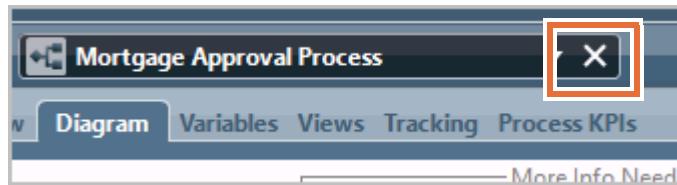
- \_\_\_ v. Under Execution State, select Mortgage Approval Process and under the Variables tab, double-click application.



- \_\_\_ w. The inspector displays the XML representation of the final application business object: Scroll down in the XML pane. You should notice that although the status was set to "ACCEPT" in the Review for Approval, the Advanced Integration service changed it to "RECORDED FROM AIS":



- \_\_ x. Close Mortgage Approval Process by clicking X.

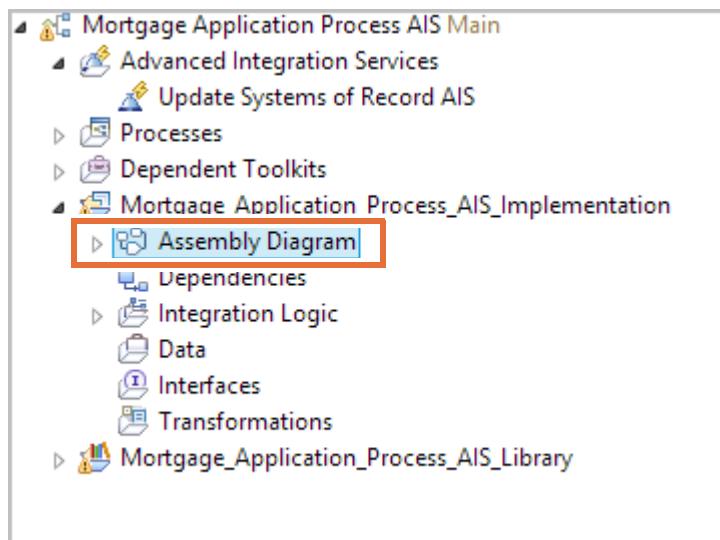


- \_\_ y. Click Process Center at the upper right corner to switch to the Process Center view.  
\_\_ z. This portion of the test is successfully concluded, as it shows the invocation of AIS from a running BPD and a coach in IBM Process Designer.

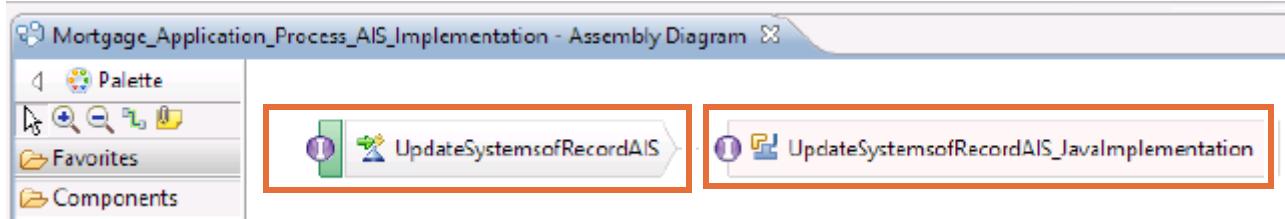
## Part 6: Implement the revised AIS in IBM Integration Designer

In the previous section, you created a dummy implementation of an AIS to prove the concept of being able to invoke integration artifacts from BPMN processes. The integration artifacts can be complex service orchestrations, with mappings and adapters. In this section, you take the implemented Advanced Integration service and reimplement it as a BPEL business process. As soon as you understand how this solution, including coach, BPD, AIS, and BPEL is implemented, you can use this scenario to customize and create several different complex integrations.

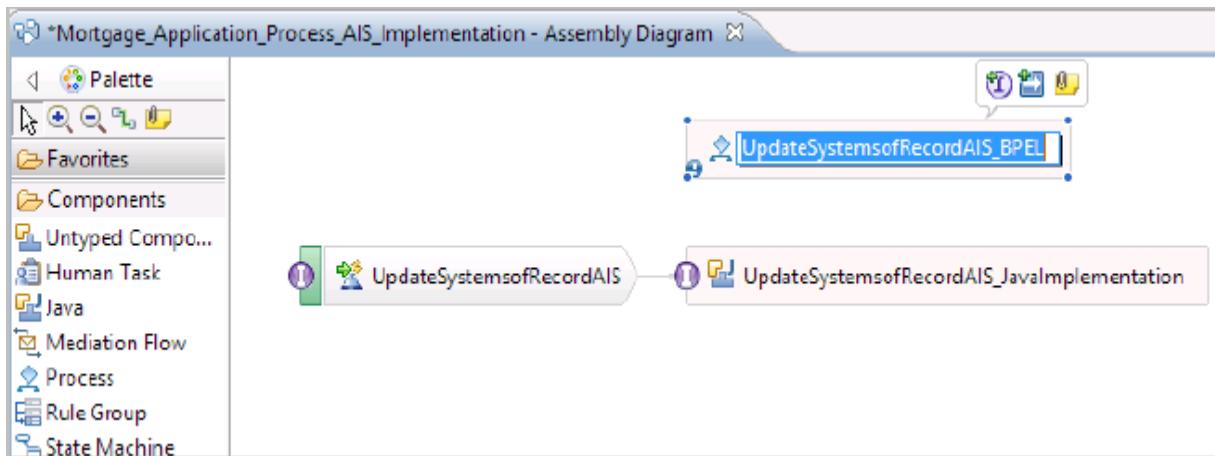
- \_\_ 1. Use IBM Integration Designer to reimplement the AIS.
  - \_\_ a. Minimize the IBM Process Designer.
  - \_\_ b. Maximize the IBM Integration Designer workspace that you worked on in the previous section.
  - \_\_ c. In Business Integration, expand Mortgage\_Application\_Process\_AIS\_Implementation and double-click Assembly Diagram.



- \_\_\_ d. The Assembly Diagram shows two components: An export (that the outside world uses to call the integration) – **UpdateSystemsofRecordAIS**, and the Java Implementation that you created in earlier portion of this exercise - **UpdateSystemsofRecordAIS\_JavalImplementation**.

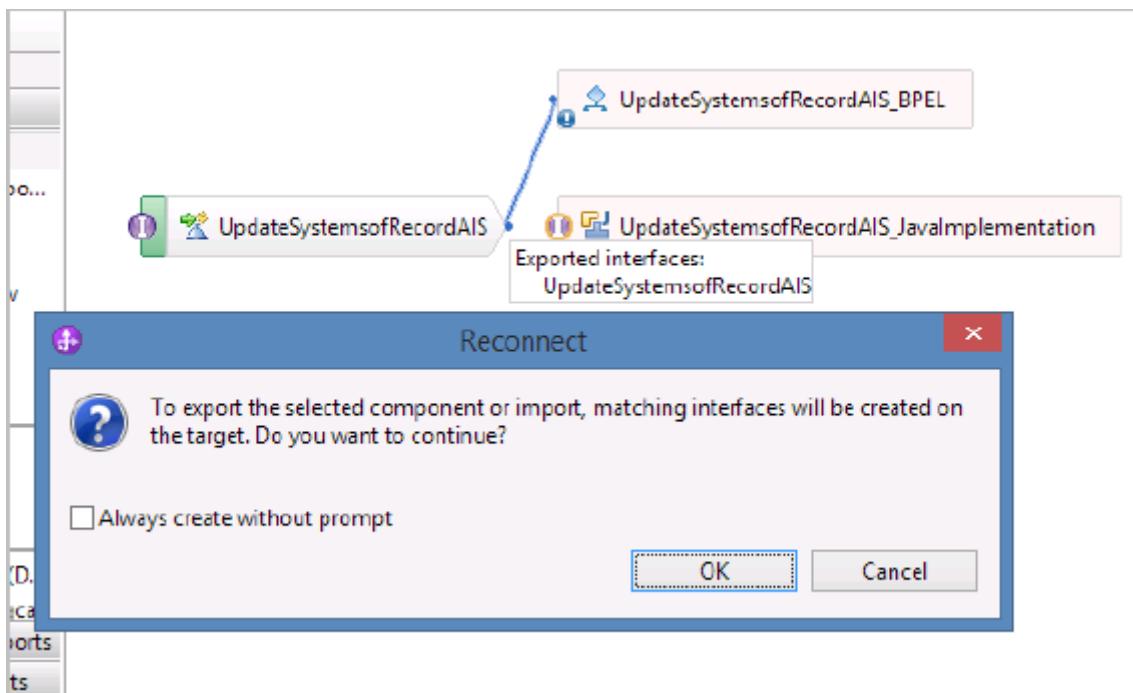


- \_\_\_ e. You replace the Java implementation with a BPEL process. To create a BPEL Process skeleton and implement it, drag the Process component from the palette to the canvas. Rename the Component to **UpdateSystemsofRecordAIS\_BPEL**.

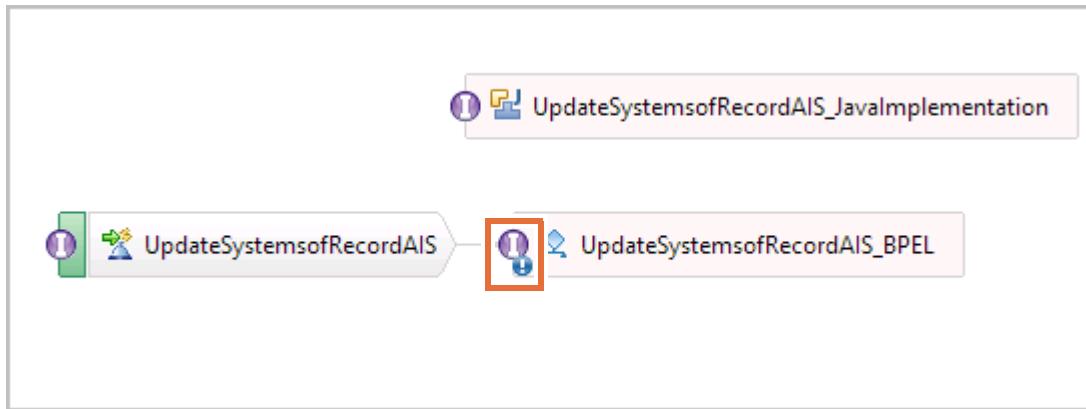


- \_\_\_ f. Disconnect the export from the Java Implementation and connect it to the **UpdateSystemsofRecordAIS\_BPEL** process.

- \_\_ g. Click OK when presented with the option to create a matching interface on the target BPEL process. It associates the same interface (business object application and operation invoke) that is used by the Java implementation.

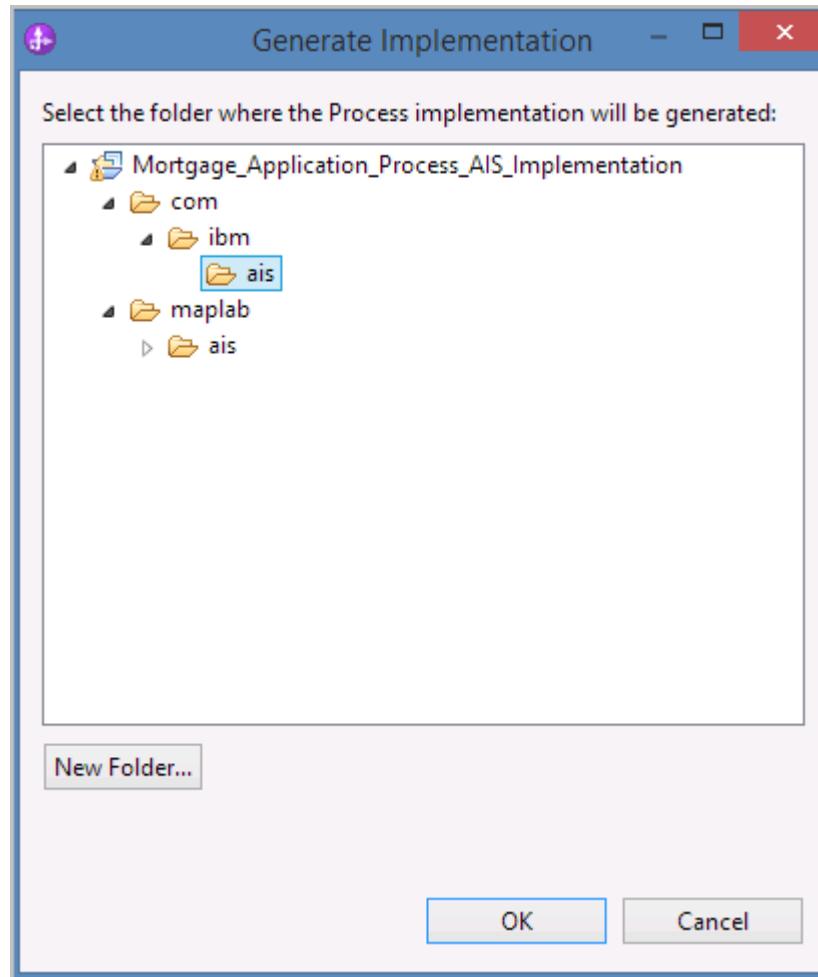


- \_\_ h. Notice an exclamation point next to the BPEL process, which means that it is not implemented yet.

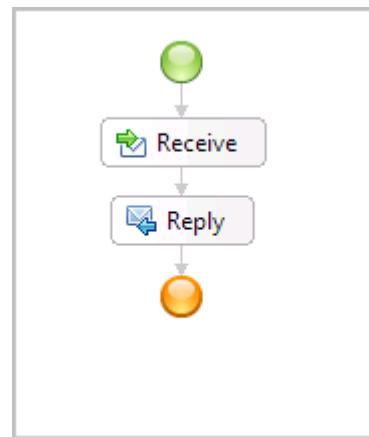


- \_\_ i. Double-click the BPEL process to open it in the BPEL editor. Click **Yes** at the Open window.  
\_\_ j. In the Generate Implementation window, click **New Folder** and specify com/ibm/ais for the target folder.

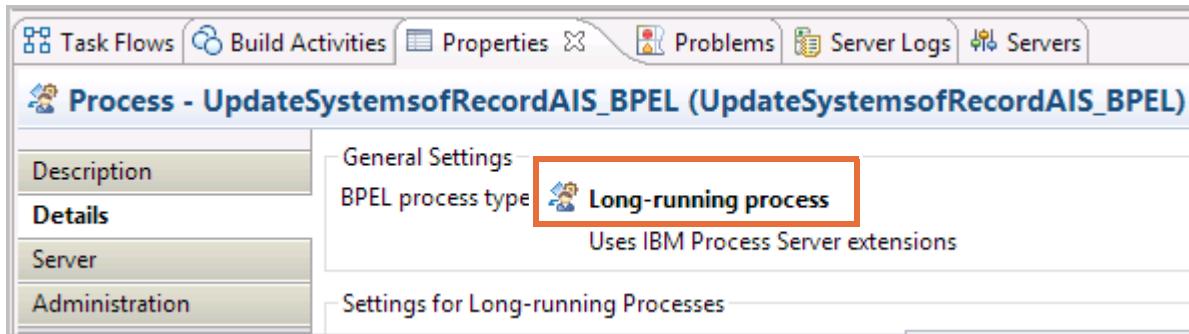
- \_\_\_ k. Click **OK** to confirm the new folder, and then **OK** to complete the operation.



- \_\_\_ 2. Continue with the implementation of the BPEL Process.
- \_\_\_ a. Wait until the workspace is fully built. The BPEL Process diagram is displayed when this workspace build is completed. The BPEL Process diagram is displayed top to down (while the BPMN in the Process Designer is displayed left to right), and starts off with a Receive and Reply activities.

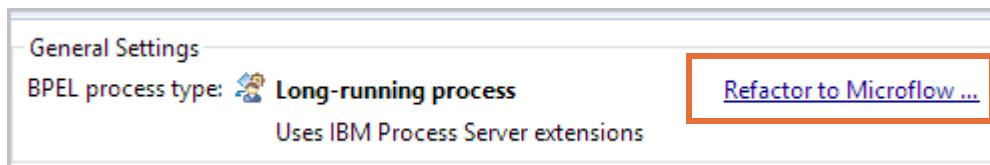


- \_\_\_ b. Right-click the green circle at the top of the process diagram, and then click **Show in > Properties View**.
- \_\_\_ c. Click **Details** to open the general properties for this process at the bottom of the screen. Notice that, by default, this process is configured to be a long-running process.

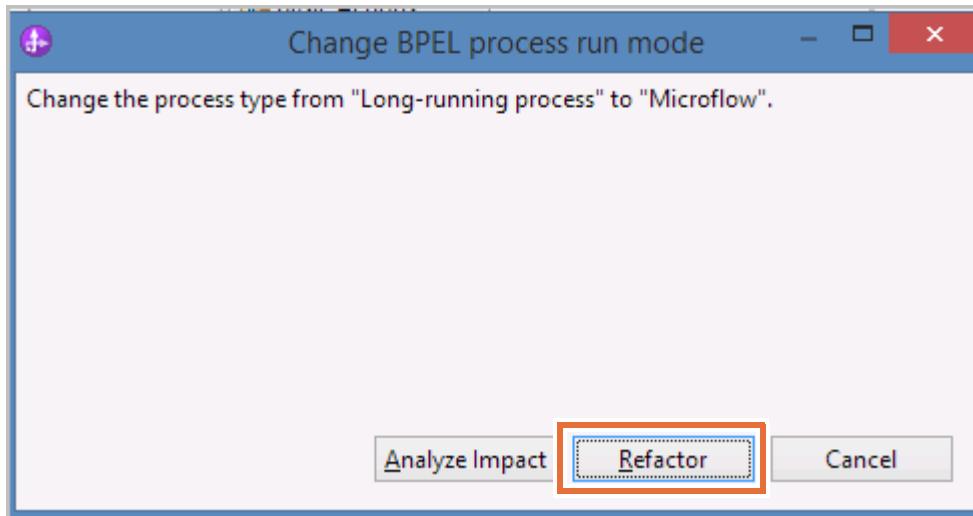


Long-running processes are useful when process state needs to be preserved for a longer time than the span of a single transaction. In this exercise, because the process needs to record the transaction to one or more systems of record, this state does not need to be preserved. You can make this process a microflow, which has better performance characteristics than a long running process.

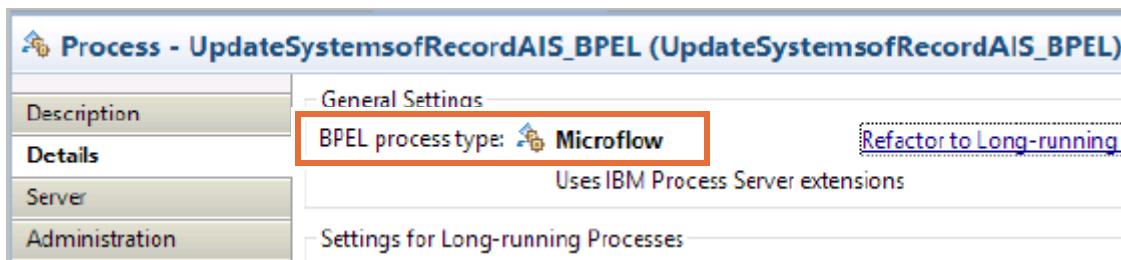
- \_\_\_ d. Click Refactor to microflow.



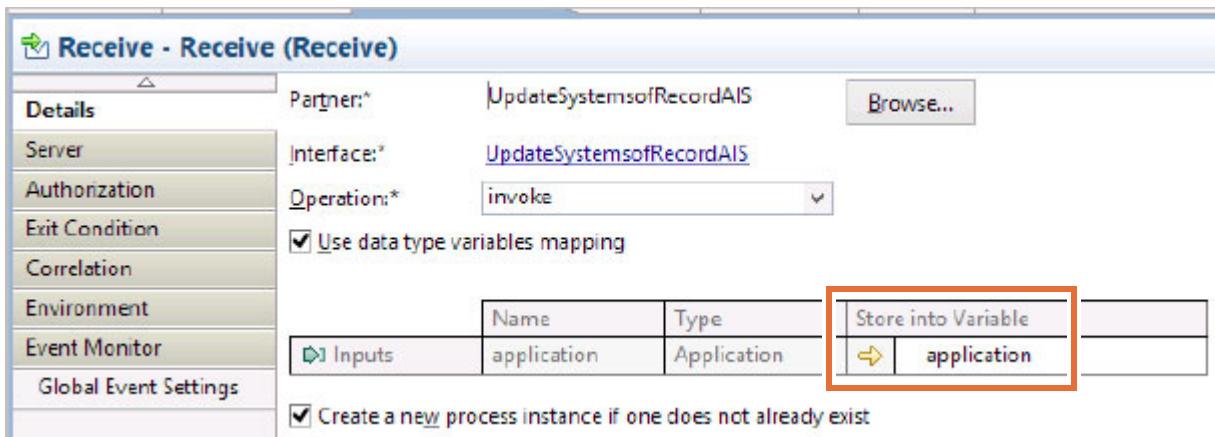
- \_\_\_ e. If you are prompted to save the outstanding changes, click **OK**.
- \_\_\_ f. In the **Change BPEL process run mode** window, click **Refactor**.



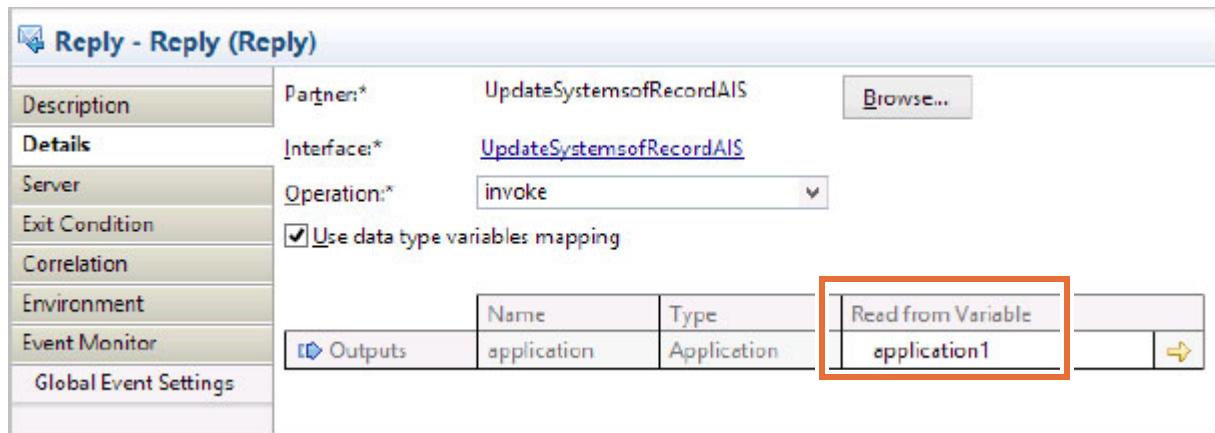
- \_\_\_ g. Notice that the process type is now **Microflow**.



- \_\_\_ h. In the process editor, select the **Receive** activity.
- \_\_\_ i. In the **Properties** view that follows the diagram, switch to the **Details** tab.
- \_\_\_ j. Notice the invoke operation and input parameter (application) that was specified in the earlier part of this exercise.
- \_\_\_ k. Notice the name of the BPEL variable, which maps to the input business object (application).

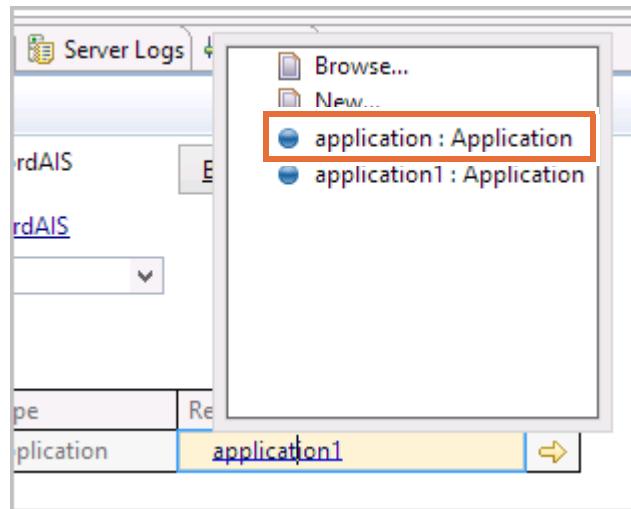


- \_\_\_ l. Click the **Reply** activity.
- \_\_\_ m. In the Properties view that follows the diagram, click the **Details** tab.
- \_\_\_ n. Notice at the operation and output parameter (application) that was specified in the earlier part of this exercise. The Output that is named application receives the data from the variable application1, which is also a local variable.



The reason why the input and output variables are different is because in many cases – you might want to preserve the original state of the input while you are working on the output. But in this exercise it is not needed. You can change the input variable directly, and return it as output.

- \_\_\_ o. Click **application1** and then select **application** from the list.

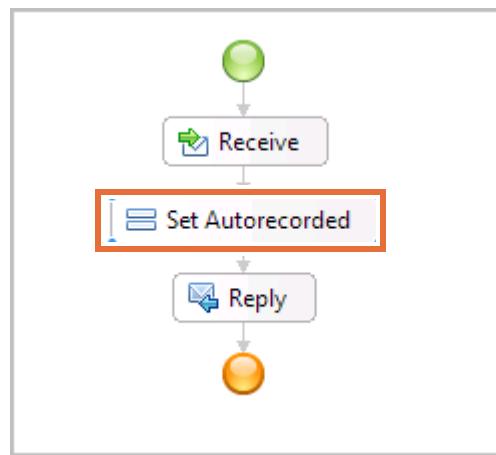


- \_\_\_ p. Verify that the **Read from Variable** value changes to **application**.

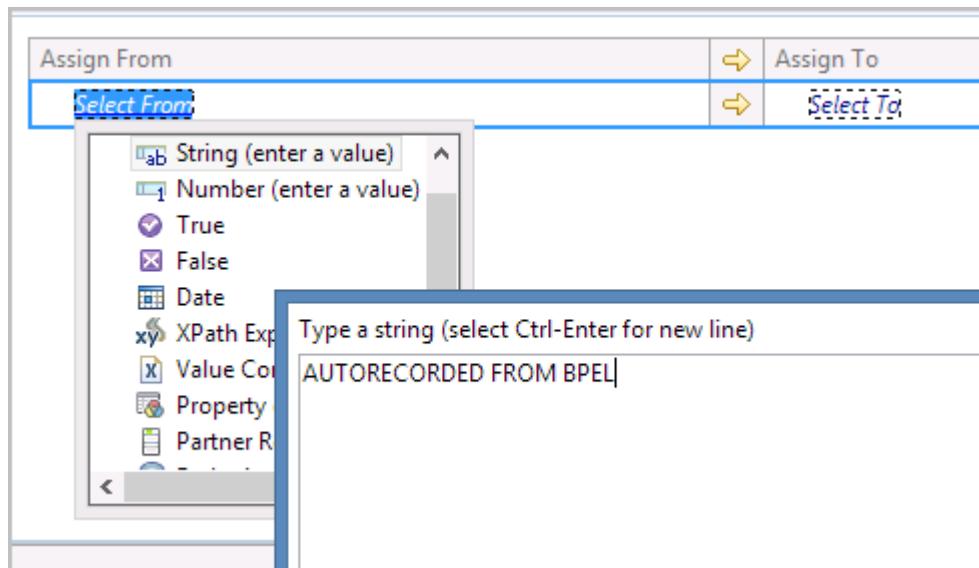
|         | Name        | Type        | Read from Variable |
|---------|-------------|-------------|--------------------|
| Outputs | application | Application | application        |

- \_\_\_ q. You can eliminate the extra BPEL variable **application1**. On the right side of the main editor pane, locate the **Variables** drawer, select **application1**, and then click the red X to delete it:
  - \_\_\_ r. Save your work (Ctrl-S) and wait for the build to finish.
- \_\_\_ 3. Add an assign activity that sets the mortgage application status.
- \_\_\_ a. Place an Assign activity between Receive and Reply:

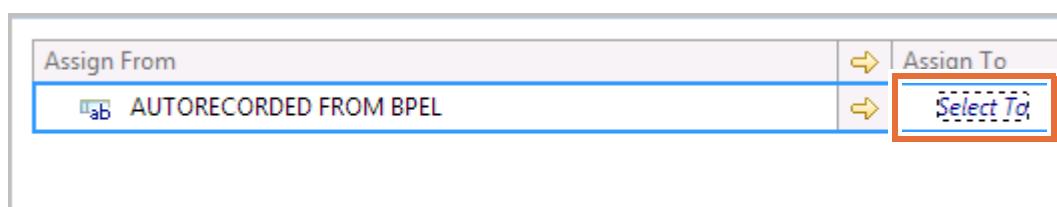
- \_\_ b. Change the display name of the activity to: Set Autorecorded



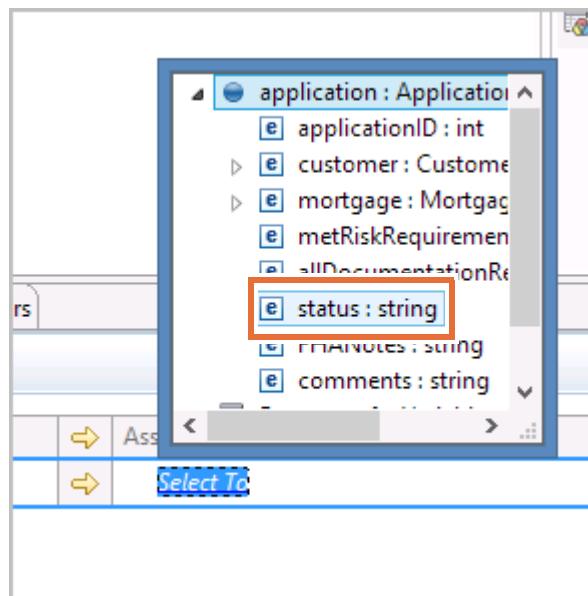
- \_\_ c. Select **Set Autorecorded** and then in the **Properties** view that follows, switch to the **Details** tab.
- \_\_ d. Click **Select From** and then select **String (enter a value)**. Type **AUTORECORDED FROM BPEL** in the **Type a string** field.



- \_\_ e. Press the Enter key to complete and then click **Select To**.



- \_\_ f. Expand the application and then select status.



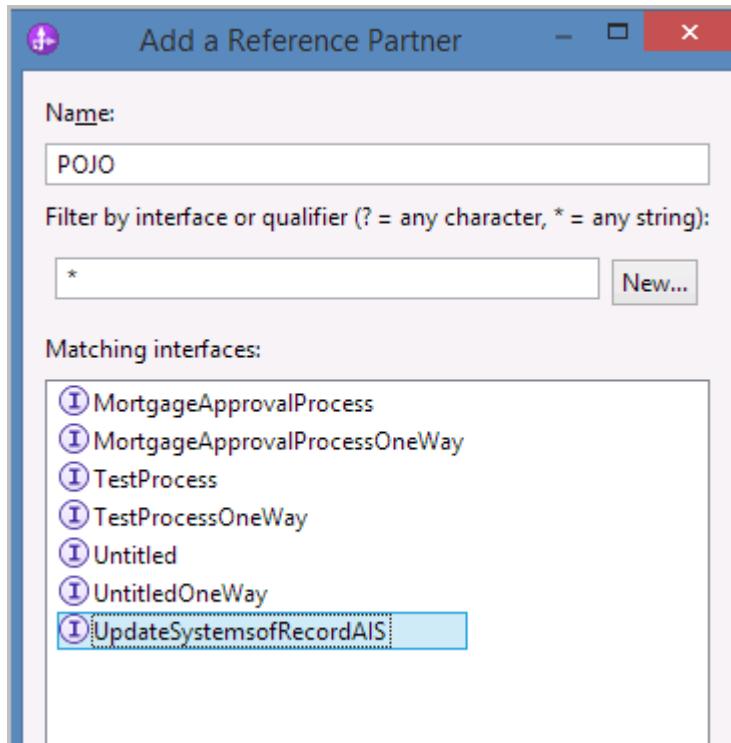
- \_\_ g. Verify that the assigned values are correct.

| Assign From            | Assign To          |
|------------------------|--------------------|
| AUTORECORDED FROM BPEL | application status |

- \_\_ h. Save your work.

- \_\_ 4. Add an invoke activity that does the recording. Normally, BPEL processes invoke a number of external services, by using web services, JMS, adapters, and other protocols. In this exercise, you are going to invoke a single service. For the service, you implement the POJO component.
- \_\_ a. In the process editor, add an invoke activity just above the Reply activity.
- \_\_ b. Change the display name of the new invoke activity to: Invoke POJO

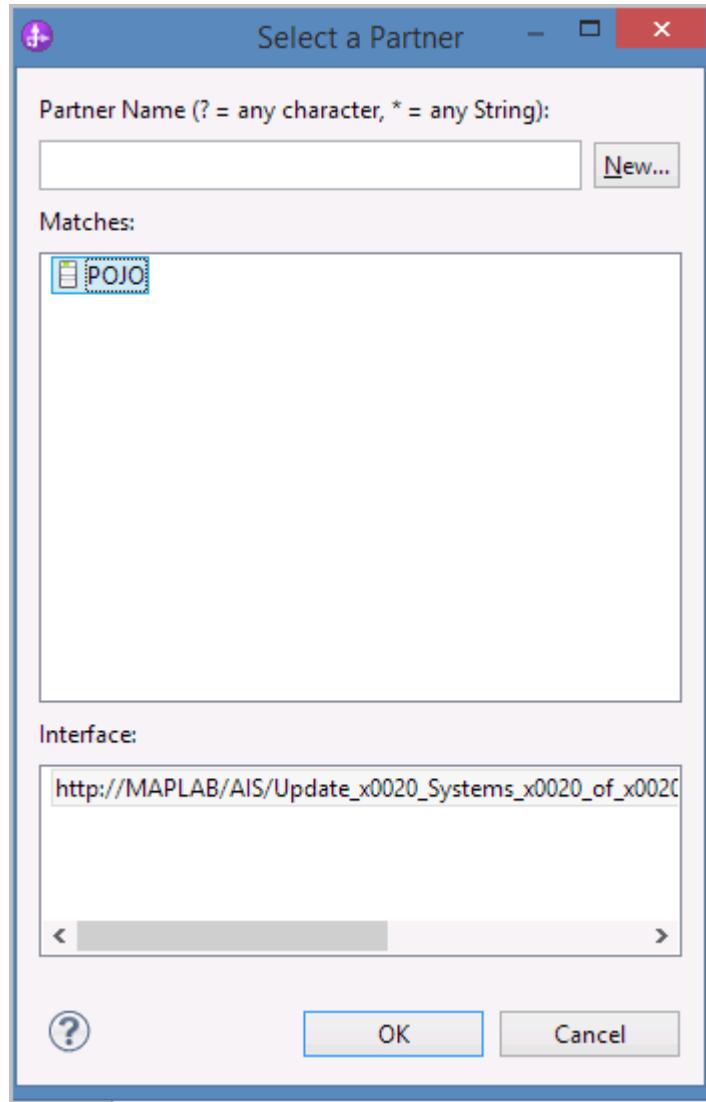
- c. Invoke operations need a “Reference Partner” to invoke. The reference partner represents the external service that is being invoked. On the right side, click the plus sign next to **Reference Partners**. For Name, enter **POJO** and then select the **UpdateSystemsofRecordAIS** interface under Matching Interfaces and click **OK**.



The partner gets created.

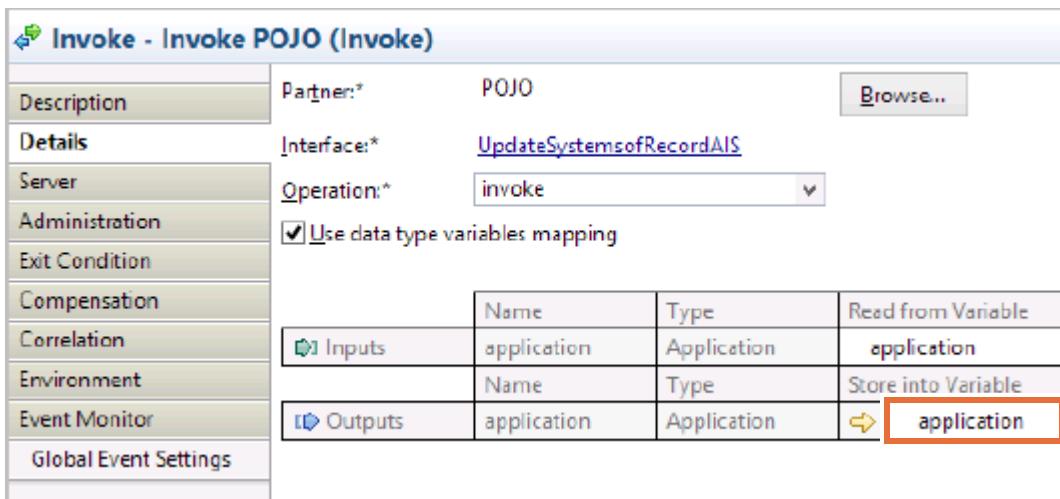
- d. Click the **Invoke POJO** operation that you just added.

- \_\_\_ e. In the Properties view, click the Details tab. Then, click **Browse** next to Partner. Select **POJO** and click **OK**.

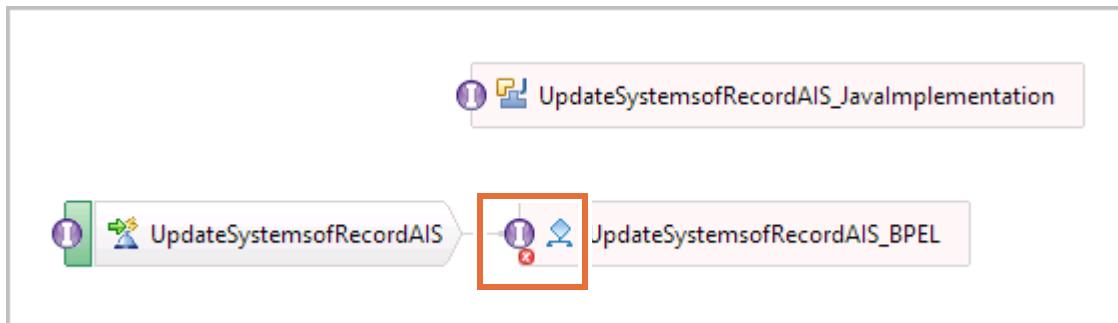


- \_\_\_ f. For the Inputs, click none and then select **application**.

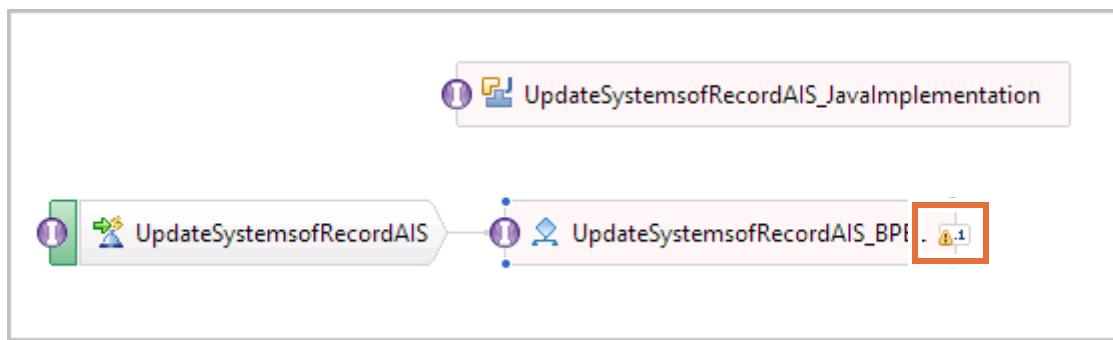
- \_\_ g. For the Outputs, click none and then select **application**.



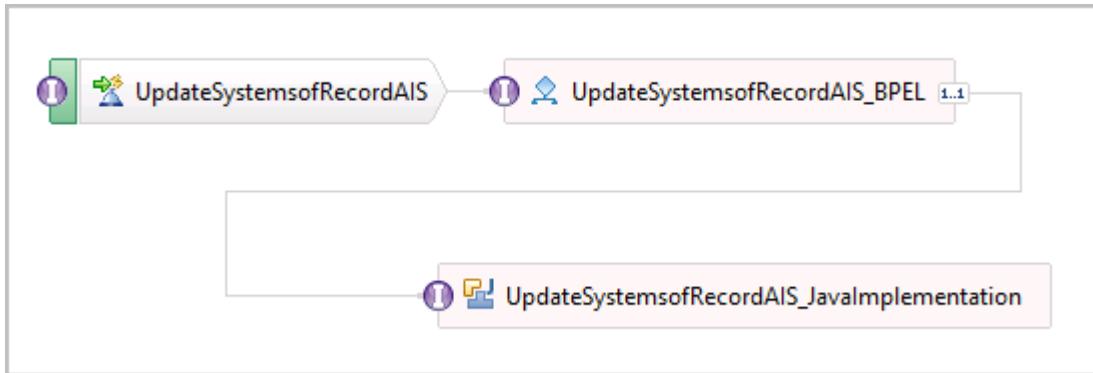
- \_\_ h. The BPEL process is now complete. This process sets the mortgage status to AUTORECORDED FROM BPEL and then calls a service that does the actual recording.
- \_\_ i. Save your work. (Ctrl+S), and close the BPEL editor.
5. Wire the BPEL process to the POJO service.
- \_\_ a. Go back to the Assembly Diagram. Notice the error indication by the BPEL process component.



- \_\_ b. The error exists because you manually added a partner reference, which is not yet reflected on the diagram. Right-click the BPEL process component and click **Synchronize Interfaces and References > from Implementation**.
- \_\_ c. Click **Yes** in the confirmation dialog. Notice that now the BPEL process has a reference icon and the error goes away.



- \_\_\_ d. You wire the reference to the target service, hover over the reference icon so that a connector appears. Then, click and drag it to the input interface of the POJO component. The connection is made, and the result looks similar to the following image:

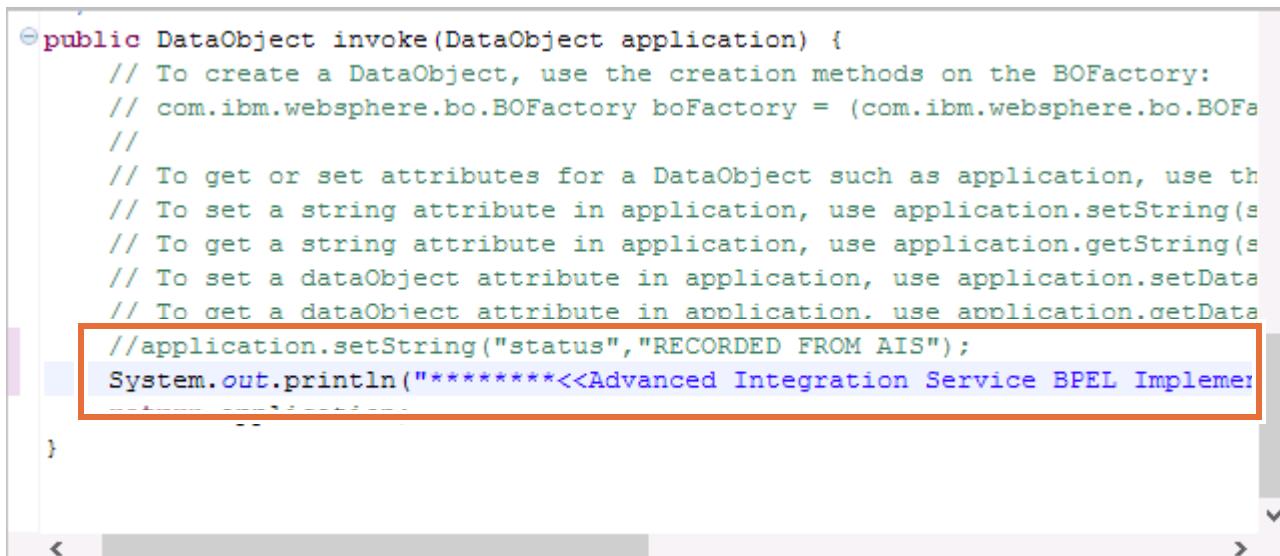


- \_\_\_ e. Save your work.  
\_\_\_ f. Double-click the POJO component to open up the implementation. The Java editor opens up.  
\_\_\_ g. Scroll down to the bottom and locate the invoke object. You implemented Java code earlier in the exercise.

```
public DataObject invoke(DataObject application) {
 // To create a DataObject, use the creation methods on the BOFactory
 // com.ibm.websphere.bo.BOFactory boFactory = (com.ibm.websphere.bo.
 //
 // To get or set attributes for a DataObject such as application, us
 // To set a string attribute in application, use application.setString()
 // To get a string attribute in application, use application.getString()
 // To set a dataObject attribute in application, use application.set
 // To get a dataObject attribute in application, use application.get
 application.setString("status", "RECORDED FROM AIS");
 System.out.println("*****<<Advanced Integration Service Java Impl
 return application;
}

}
```

- \_\_\_ h. Comment out: `application.setString("status", "RECORDED FROM AIS");`  
 Change the text of the print statement by replacing Java with: BPEL



```

public DataObject invoke(DataObject application) {
 // To create a DataObject, use the creation methods on the BOFactory:
 // com.ibm.websphere.bo.BOFactory boFactory = (com.ibm.websphere.bo.BOFa
 //
 // To get or set attributes for a DataObject such as application, use th
 // To set a string attribute in application, use application.setString(s
 // To get a string attribute in application, use application.getString(s
 // To set a dataObject attribute in application, use application.setDa
 // To get a dataObject attribute in application. use application.getDa
 //application.setString("status", "RECORDED FROM AIS");
 System.out.println("*****<<Advanced Integration Service BPEL Impleme
}

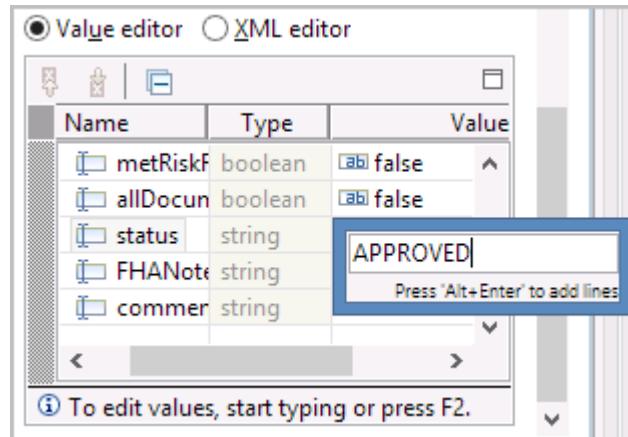
```

- \_\_\_ i. Now the Java POJO does not do anything other than writing to the log file. Save your work.  
 \_\_\_ j. Save your changes and close the Java editor.

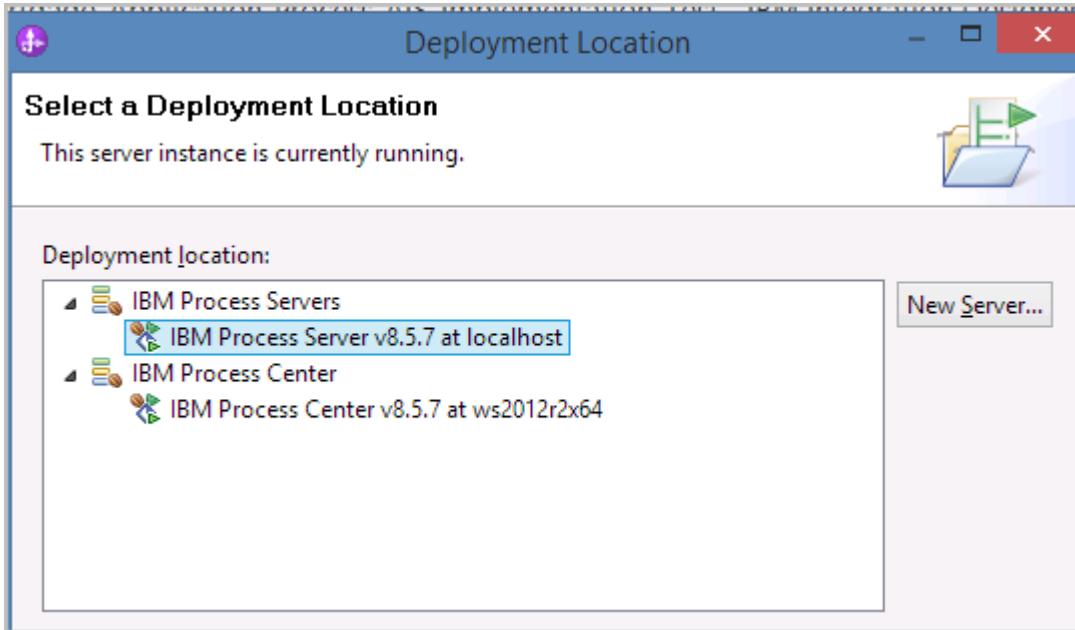
## Part 7: Unit Test the updated AIS and the new BPEL component in IBM Integration Designer

- \_\_\_ 1. Publish your changes to Process Center.
  - \_\_\_ a. Verify that Mortgage Approval Process has [changed] next to it.
  - \_\_\_ b. Right-click [changed] Mortgage Approval Process Main, and click **Publish to Process Center**.
- \_\_\_ 2. Start the UTE server (if it is not already running).
  - \_\_\_ a. If necessary, in the Servers view, right-click IBM Process Server v8.5.7 at localhost, and click Start. Wait for the startup process to complete before continuing.
  - \_\_\_ b. The server is started when the message Server server1 open for e-business is displayed in the Server Logs view. The server status also changes to Started in the Servers view.
- \_\_\_ 3. Run the AIS test.
  - \_\_\_ a. In Business Integration, expand **Mortgage Application Process AIS Main > Advanced Integration Services**, then right-click **Update Systems of Record AIS** and then select **Test Implementation**.
  - \_\_\_ b. The Integration Test Client starts. In the Initial request parameters to the right, scroll through the value editor and locate the **status** field under the **Name** column.

- \_\_\_ c. Enter APPROVED in the value column for status.

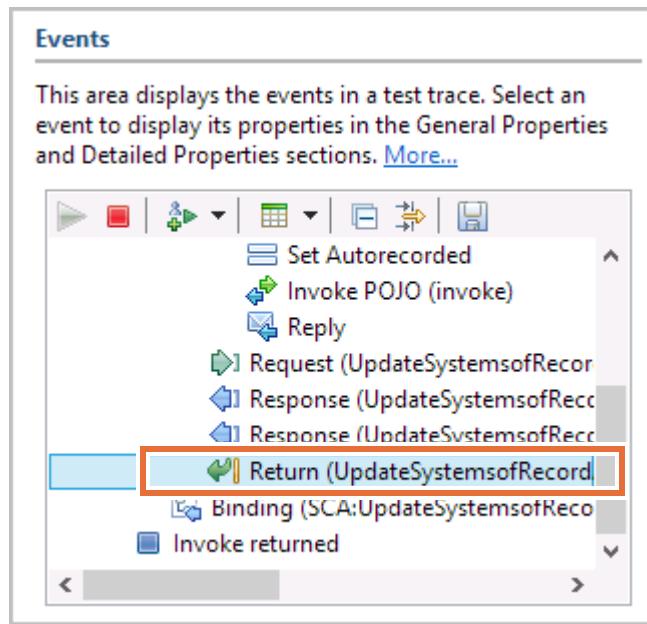


- \_\_\_ d. Under Events, click the Continue icon to start the Test.  
\_\_\_ e. In the Deployment Location window, expand IBM Process Servers and select IBM Process Server v8.5.7 at localhost. Click **Finish**.



- \_\_\_ f. In the User Login - Default Module Test window, accept the default values and click **OK**.

- g. It takes a few minutes, after which the project is deployed to the test server and runs the AIS implementation. You see a trace of all the components that are being executed, and as a result, In the Events section, click **Return**.



- h. In the return parameters section to the right, notice the value of status. It was earlier set to APPROVED, and it is now updated to AUTORECORDED FROM BPEL.

The screenshot shows the 'Value Editor' window with the following table:

| Name                     | Type     | Value                         |
|--------------------------|----------|-------------------------------|
| liabilities              | double   | 0.0                           |
| downPayment              | double   | 0.0                           |
| mortgage                 | Mortgage |                               |
| mortgageType             | string   |                               |
| amount                   | double   | 0.0                           |
| interestRate             | double   | 0.0                           |
| period                   | int      | 0                             |
| metRiskRequirement       | boolean  | false                         |
| allDocumentationRequired | boolean  | false                         |
| <b>status</b>            | string   | <b>AUTORECORDED FROM BPEL</b> |
| FHANotes                 | string   |                               |
| comments                 | string   |                               |

- \_\_ i. Review the events stack trace. It shows that the BPEL process is being executed too.

The screenshot shows the 'Events' tab in the Process Inspector. A message box at the top says: 'This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)'.

The main pane lists various events in a tree structure:

- Invoke (UpdateSystemsofRecordAIS:invoke)**
- Invoke started**
  - Binding (SCA:UpdateSystemsofRecordAIS)**
  - Invoke (UpdateSystemsofRecordAIS:invoke)**
  - Request (UpdateSystemsofRecordAIS --> UpdateSystemsofRecordAIS\_BPEL:invoke)**
  - Fine-Grained Trace (UpdateSystemsofRecordAIS\_BPEL:UpdateSystemsofRecordAIS\_BPEL:invoke)**
    - Receive (invoke)**
    - Set Autorecorded**
    - Invoke POJO (invoke)**
    - Reply**
  - Request (UpdateSystemsofRecordAIS\_BPEL --> UpdateSystemsofRecordAIS\_JavaImpl:invoke)**
  - Response (UpdateSystemsofRecordAIS\_BPEL <-- UpdateSystemsofRecordAIS\_JavaImpl:invoke)**
  - Response (UpdateSystemsofRecordAIS <-- UpdateSystemsofRecordAIS\_BPEL:invoke)**
  - Return (UpdateSystemsofRecordAIS:invoke)**
- Binding (SCA:UpdateSystemsofRecordAIS)**
- Invoke returned**

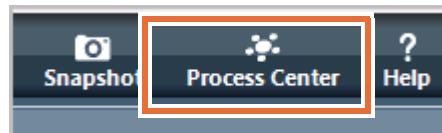
A red box highlights the 'Fine-Grained Trace' section under the initial invoke, which includes the receive, set autorecorded, invoke POJO, and reply events.

- \_\_ j. You tested the revised AIS successfully. You also successfully tested the BPEL process.

### Part 8: Run the BPD in IBM Process Designer and execute the revised AIS and invoke the new BPEL component

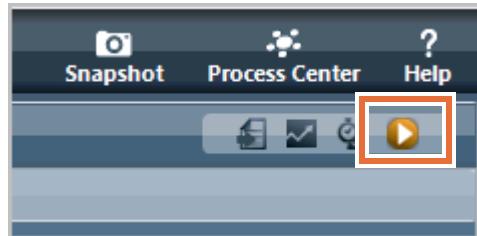
In the previous section, you tested and confirmed that the Advanced Integration service is working as expected. In this section, you take the completed AIS and execute from the BPD that used the AIS to invoke a BPEL process. You use the Process Inspector to walk through the business process and execute the Advanced Integration service.

- \_\_ 1. Test the end-to-end business process (including the AIS) in Process Designer.
  - \_\_ a. Minimize Integration Designer, and switch back to IBM Process Designer.
  - \_\_ b. If you are in the Process Designer view, then go to the Process Center view by clicking the Process Center icon at the upper right.



- \_\_ c. In the **Process Apps** tab, click **Open in Designer** next to Mortgage Application Process AIS.

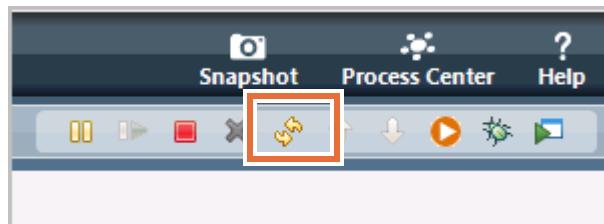
- \_\_\_ d. Click Processes and then double-click Mortgage Approval Process.
- \_\_\_ e. Click the Run Process icon in the upper right corner to run the Mortgage Approval Process:



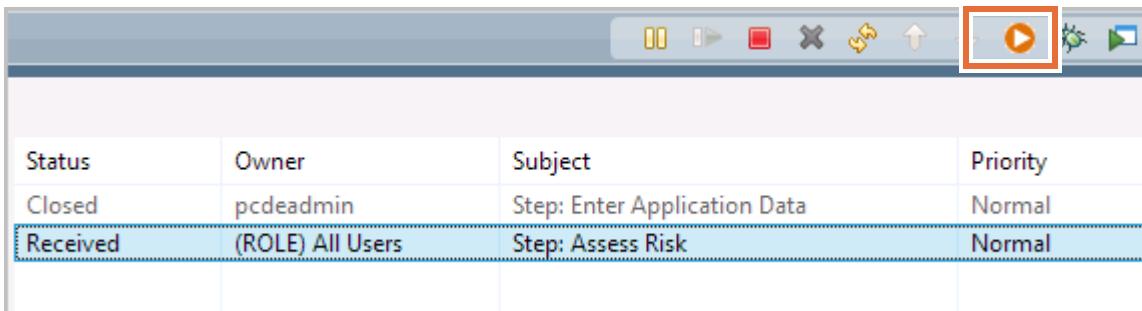
- \_\_\_ f. If prompted, click Yes to switch to the Inspector view to see the progress of the process instance.
- \_\_\_ g. The Inspector opens and the focus is on the running process instance. Select the task on the upper right pane (Enter Application Data), and click Runs the selected task.

| Status   | Owner            | Subject                      | Priority |
|----------|------------------|------------------------------|----------|
| Received | (ROLE) All Users | Step: Enter Application Data | Normal   |

- \_\_\_ h. In the Select User window, select pcdeadmin and click OK.
- \_\_\_ i. After few minutes, the browser opens with the first coach of the mortgage approval process. All fields are optional for this exercise and you can leave them empty. Click Submit to complete the task.
- \_\_\_ j. Close the browser window.
- \_\_\_ k. Go back to Process Designer and click the refresh icon.

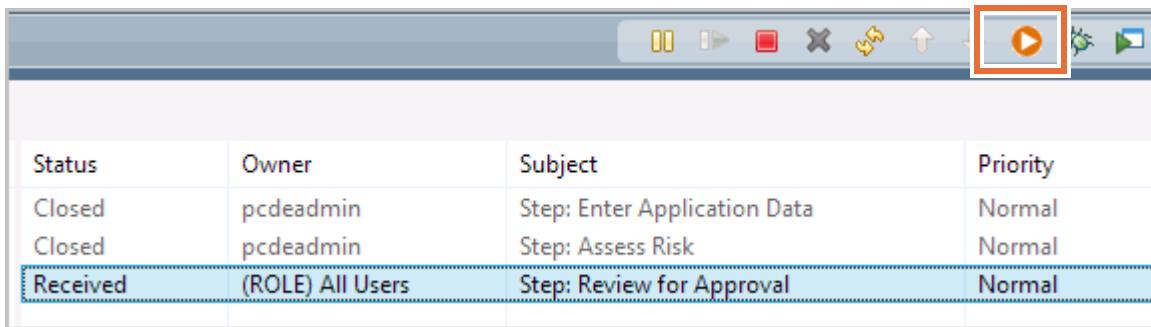


- \_\_ I. The second step on the stack is Assess Risk. Select it and click **Runs the selected task**.



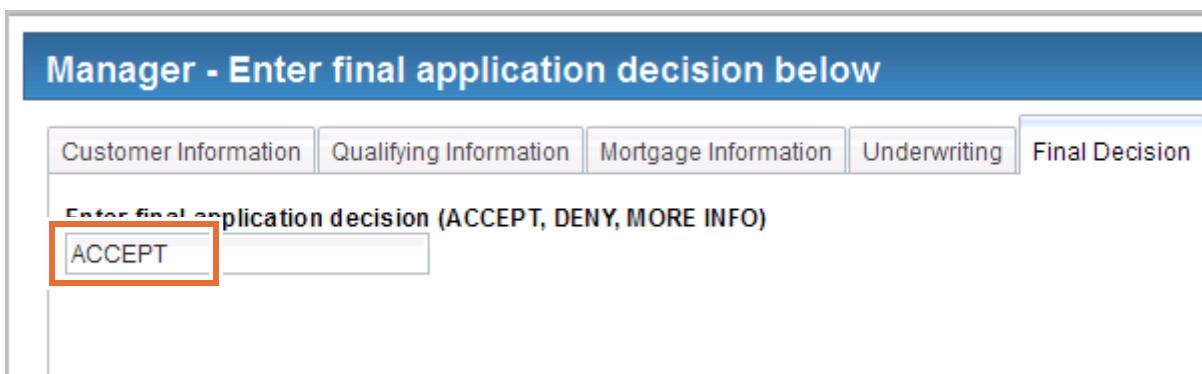
| Status   | Owner            | Subject                      | Priority |
|----------|------------------|------------------------------|----------|
| Closed   | pcdeadmin        | Step: Enter Application Data | Normal   |
| Received | (ROLE) All Users | Step: Assess Risk            | Normal   |

- \_\_ m. In the **Select User** window, select **pcdeadmin** and click **OK**.
- \_\_ n. When the coach opens, click Submit and then close the browser window.
- \_\_ o. Refresh the process state again as you did before. The third task is **Review for Approval**.
- \_\_ p. Select it and click **Runs the selected task**.



| Status   | Owner            | Subject                      | Priority |
|----------|------------------|------------------------------|----------|
| Closed   | pcdeadmin        | Step: Enter Application Data | Normal   |
| Closed   | pcdeadmin        | Step: Assess Risk            | Normal   |
| Received | (ROLE) All Users | Step: Review for Approval    | Normal   |

- \_\_ q. In the **Select User** window, select **pcdeadmin** and click **OK**.
- \_\_ r. In the coach, this time, make sure that you switch to the Final Decision tab and enter: **ACCEPT**



Manager - Enter final application decision below

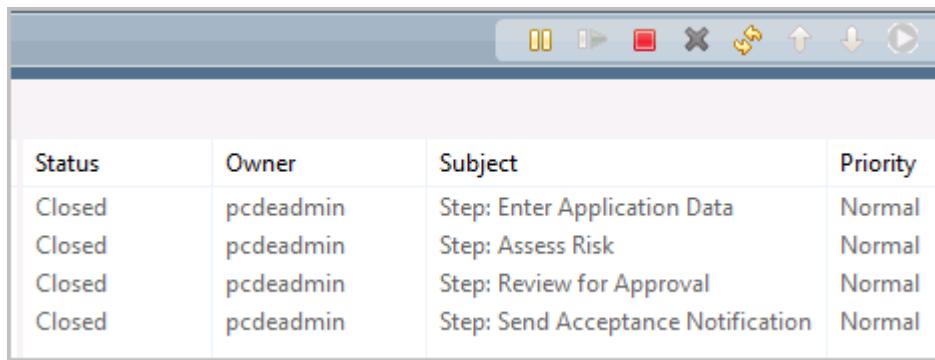
Customer Information Qualifying Information Mortgage Information Underwriting **Final Decision**

Enter final application decision (ACCEPT, DENY, MORE INFO)

**ACCEPT**

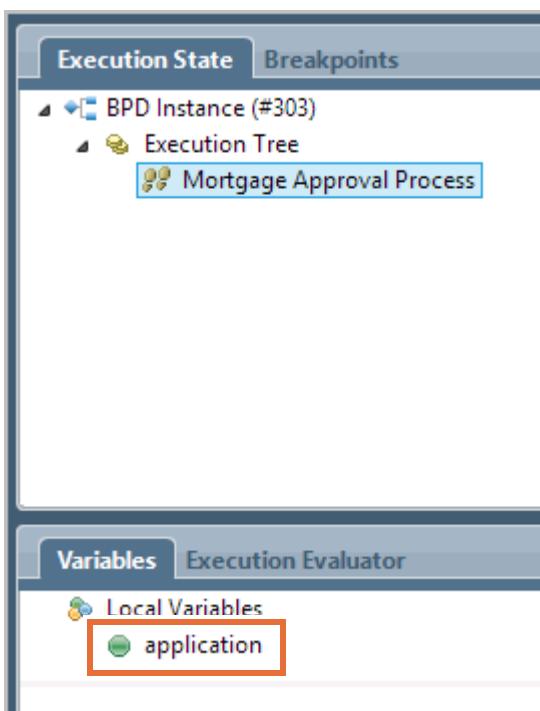
- \_\_ s. Click **Submit** and close the browser.

- \_\_ t. In the Inspector, click the refresh icon a few times until the process completes. The stack should look similar to the following image:

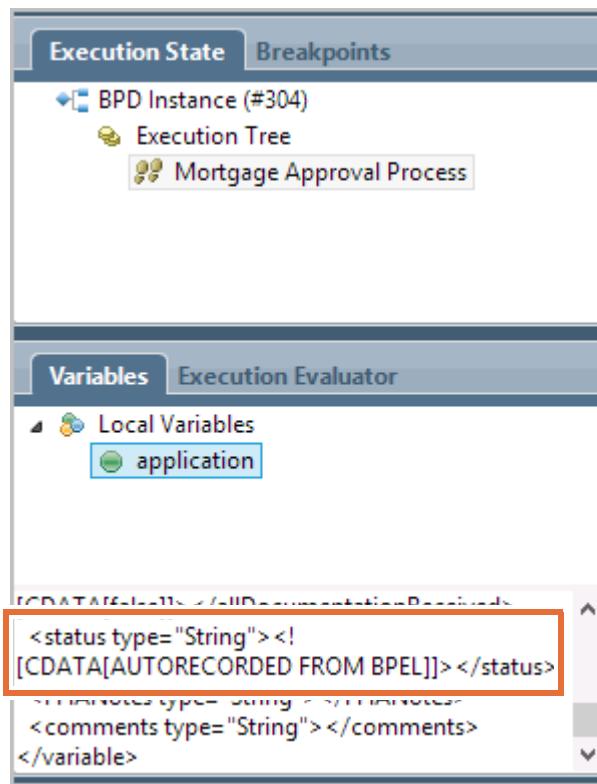


| Status | Owner     | Subject                            | Priority |
|--------|-----------|------------------------------------|----------|
| Closed | pcdeadmin | Step: Enter Application Data       | Normal   |
| Closed | pcdeadmin | Step: Assess Risk                  | Normal   |
| Closed | pcdeadmin | Step: Review for Approval          | Normal   |
| Closed | pcdeadmin | Step: Send Acceptance Notification | Normal   |

- \_\_ u. Under Execution State, select Mortgage Approval Process and under the Variables tab, double-click application.



- v. The inspector displays the XML representation of the final application business object: Scroll down in the XML pane as needed. You should notice that although the status was set to “ACCEPT” in the Review for Approval, the Advanced Integration service changed it to “RECORDED FROM BPEL”:



- w. Click Process Center at the upper right corner to switch to the Process Center view.  
— x. This portion of the test is successfully concluded, as it shows the invocation of AIS and then BPEL from a running BPD and a coach in IBM Process Designer.
2. Close Process Designer.  
3. Stop the IBM Process Center server.  
a. Double-click the **Stop Process Center cluster** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.  
b. Double-click the **Stop Process Center node agent** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.  
c. Click the **Stop Process Center deployment manager** icon on the desktop. Wait until it stops successfully. Press any key to close the command window.  
4. Stop the IBM Process Server unless it is already stopped.  
a. Right-click **IBM Process Server v8.5.7 at localhost** and click **Stop** to stop the server.  
5. Close IBM Integration Designer.  
6. Close any other open windows.

## End of exercise

## Exercise review and wrap-up

You successfully completed the Advanced Integration Lab. As you saw, the flow of a business process from the Process Designer to Advanced Integration in the Integration Designer is seamless. The capabilities of Advanced Integration provided by IBM Business Process Manager Advanced open a whole world of service orientation. These capabilities include access to process choreography through BPEL, adapters, easy incorporation of Java, and the tools that are needed to test every part of the solution.

Though the top-down approach is the most common approach, the other common approach is a bottom-up approach. This process is also straightforward, where an Integration Designer creates Advanced Integrations and publishes them to the Process Center, and they are made visible to the Process Designer to use in a Business Process Definition.

It is important to always test and take small incremental function steps. If you combine this practice with frequent playbacks, you find that it not only shortens the development lifecycle, but also the stakeholders and business users continue to be involved and invested in the application development process. This involvement allows for better communication and quick agreement on what the requirements are. If you have any more questions on any of these labs, contact your IBM Technical Professional.



# Appendix A. Re-creating the exercise environment

If you want to re-create the exercise environment, you can do so on your own machine, or you can use your own VMware image. In either case, the steps in this appendix apply.

## Installing the software

Perform the following steps to install the software.

### ***Part 1: Installing the IBM Integration Designer test environment***

To install the IBM Integration Designer V8.5.7 test environment:

- \_\_\_ 1. Log in to Windows as `Administrator` or log in with an account that is a member of the Administrators group.
- \_\_\_ 2. Insert the installation media, or in Windows Explorer, browse to the folder that contains the extracted installation files, for example: `C:\IIDExtract\`
- \_\_\_ 3. Right-click `launchpad64.exe` and click **Run as administrator** to start the installation.

4. Verify that **IBM Integration Designer for IBM Business Process Manager Advanced - Process Server** is selected and click **Next**. This option includes an IBM Process Server test environment.

## Welcome to the IBM Integration Designer installation

The typical installations provide preselected configurations and determine the environment that is enabled when IBM® Integration Designer starts. You can change the configuration selections during the installation or run Installation Manager later to make changes. You can also change your environment later in IBM Integration Designer.

Select one of the following typical installations:

**IBM Integration Designer for IBM Business Process Manager Advanced - Process Server**

Includes an IBM Process Server test environment.

**IBM Integration Designer for IBM Business Monitor**

Includes an IBM Business Monitor test environment (Windows only).

**IBM Integration Designer for IBM Business Process Manager Advanced - Process Server and IBM Business Monitor**

Includes an IBM Process Server test environment and also supports IBM Business Monitor.

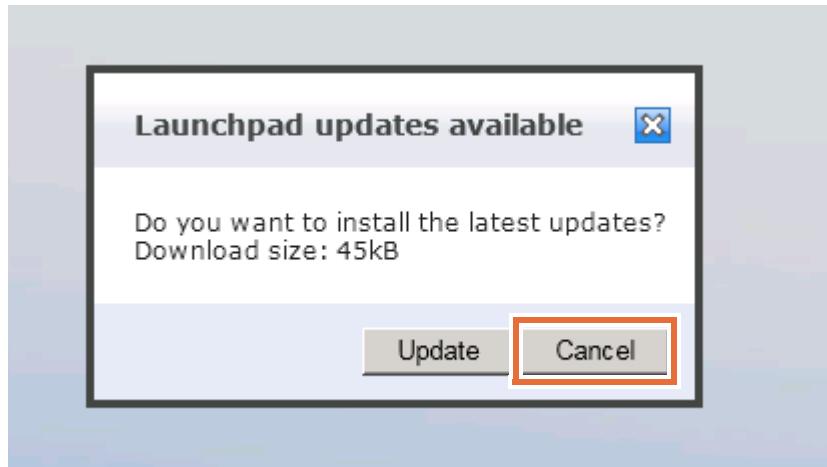
**IBM Integration Designer**

Does not include a test environment.

**IBM Integration Designer for WebSphere DataPower (Deprecated)**

**Warning**

If at any time during the installation of any software, you receive a dialog box with a message that prompts you to install the newest updates, click **Cancel**. It is important that you do not download any software updates during the installation, or else unexpected behavior might occur with the installation and the student lab exercises.



- \_\_\_ 5. In the **IBM Integration Designer Features** panel, under “Select the features to install”, take the following actions:
  - \_\_\_ b. Clear **Monitor Model editor**.
  - \_\_\_ c. Select **Client development tools**.
  - \_\_\_ d. Select **Additional development tools**. It might be necessary to scroll down to view this option.

- \_\_ e. Verify your selections and click **Next**.

**IBM Integration Designer features**

1. Specify the installation location:
2. Select the features to install:  


- \_\_ 6. In the **Install a Process Server** panel, take the following actions:

- \_\_ a. Enter the `<host name>` in the **Hostname** field. Notice that the host name that you enter is different from the one entered in the screen capture that is shown at the end of this step. The host name is unique to your environment and is associated with the computer or environment where you are installing the software. In the course lab environment that is used, the host name is: `ws2012r2x64`
- \_\_ b. For the **cell administrative account**, enter `admin` in the **Username** field and `web1sphere` in the **Password** and **Confirm password** fields.
- \_\_ c. For the **deployment environment administrative account**, enter `deadmin` in the **Username** field and `web1sphere` in the **Password** and **Confirm password** fields.
- \_\_ d. Verify your entries and click **Next**.

7. In the **Set up the database server** panel, enter `admin` in the **Username** field. Enter `Web1Sphere` (note the case) in the **Password** and **Confirm** fields and click **Next**.

## Set up the database server

DB2 Express will be installed for you. Choose a username and password for the database:

|           |                    |           |       |
|-----------|--------------------|-----------|-------|
| Username: | <code>admin</code> | Password: | ••••• |
| Confirm:  | <code>•••••</code> |           |       |

**Back**

**Next**

8. In the **Installation summary** panel, clear **Use your support account to include updates with the installation**.  
 9. Select **I have read and accepted the license agreement and notices** and click **Install Software**.

### Installation summary

The following options will be installed. If this list is not correct, click **Back** to make corrections.

- Integration Designer installation directory: C:\IBM\IID\v8.5
- Process Server installation directory: C:\IBM\IID\PS\v8.5
- Process Server environment: Non-production
- IBM Integration Designer features:
  - Email, Flat File, FTP, and JDBC IBM WebSphere Adapters
  - Web development tools
  - Additional development tools

Use your support account to include updates with the installation. [I don't have a support account.](#)

User Name:  Password:

I read and accepted the [license agreement](#) and [notices](#)

- \_\_\_ 10. In the **Installation information** panel, select **Show passwords** to view the passwords that are used during the installation. The installation is now started, and a status bar is displayed at the bottom of the panel. Depending on the resources, it might take a while for the installation to complete.

**Installation information**

While the installation is proceeding, print the following information for future reference.

Show passwords

**Integration Designer**

Installation location: C:\IBM\ID\v8.5

- Email, Flat File, FTP, and JDBC IBM WebSphere Adapters
- Web development tools
- Additional development tools

IBM Integration Designer features:

**Process Server**

Installation location: C:\IBM\ID\PS\v8.5

**Cell administrative account**

User name: admin Password: webisphere

**Deployment environment administrative account**

User name: deadmin Password: webisphere

**Embedded DB2 database server**

User name: admin Password: Web1Sphere

Common database: QCMNDB  
 Process database: QBPMDB  
 Performance Data Warehouse database: QPDWDB

Installing IBM Integration Designer...

**Cancel**  7%

- \_\_\_ 11. When the installation completes, a message is displayed that the installation is successful. Verify that the message **Installation completed successfully** is displayed. Click **No** when prompted to start the Integration Designer. You verify the installation later.

**Installation results**

**Installation completed successfully**

Launch Integration Designer?

**Yes** **No**

- 
- \_\_\_ 12. Click **Exit** to close the launchpad.
  - \_\_\_ 13. Click **OK** to confirm.

## Part 2: *Installing the IBM Business Process Manager Advanced V8.5.7*

Download the IBM Business Process Manager Advanced V8.5 installation files. To install the test environment:

- \_\_\_ 1. After downloading the installation files, use 7-zip software to extract each of the archives in the same directory.
- \_\_\_ 2. Log in to Windows as **Administrator** or log in with an account that is a member of the Administrators group.
- \_\_\_ 3. Insert the installation media, or in Windows Explorer, browse to the folder that contains the extracted installation files, for example: C:\BPMExtract\
- \_\_\_ 4. Right-click **launchpad64.exe** and click **Run as administrator** to start the installation.
- \_\_\_ 5. In the IBM Business Process Manager Advanced V8.5.7 Launchpad, click **Typical installation**. If prompted to download updates any time, remember to click **Cancel**.
- \_\_\_ 6. Make sure that **Install Process Center** is selected and click **Next**.
- \_\_\_ 7. In the **Install the Process Center** panel, take the following actions:
  - \_\_\_ a. Enter the `<host name>` into the **Hostname** field. Notice that the host name that you enter is different from the one entered in the screen capture at the end of this step. The host name is unique to your environment and is associated with the computer or environment where you are installing the software. In the course lab environment that is used, the host name is: ws2012r2x64
  - \_\_\_ b. For the cell administrative account, enter `bpmadmin` in the **Username** field. Enter `web1sphere` in the **Password** and **Confirm password** fields.
  - \_\_\_ c. For the deployment environment administrative account, enter `pcdeadmin` in the **Username** field. Enter `web1sphere` in the **Password** and **Confirm password** fields.
  - \_\_\_ d. Verify your entries and click **Next**.
- \_\_\_ 8. In the “Set up the database server” panel, take the following actions:
  - \_\_\_ a. Enter `admin` in the **Username** field and `web1Sphere` in the **Password** field for database connectivity. Notice that the password is case-sensitive for the database.
  - \_\_\_ b. Leave the Hostname and Port to the default values.
  - \_\_\_ c. For database names, enter the prefix `PC` at the beginning of each database name.
  - \_\_\_ d. Select **Create and initialize these databases during installation**.

9. Verify your entries and click Next.

Would you like to use an existing database? [Learn More](#)

Yes     No, I need one installed for me. (You must run the launchpad as administrator.)

1. Select the database that you want to use:
2. Specify the user name and password for database connectivity:  
User name:  Password:
3. Specify the host name and port used for database connectivity:  
Host name:  Port:
4. Specify database names: [Learn More](#)

|                                      |                                      |
|--------------------------------------|--------------------------------------|
| Common database:                     | <input type="text" value="PCCMNDB"/> |
| Process database:                    | <input type="text" value="PCBPMDB"/> |
| Performance Data Warehouse database: | <input type="text" value="PCPDWDB"/> |
| Cell only configuration database:    | <input type="text" value="PCCMNDB"/> |

Create and initialize these databases during installation.

10. In the Installation summary panel, clear Use your support account to include updates with the installation.
11. Select I have read and accepted the license agreement and notices, and click Install Software.

### Installation summary

The following options will be installed. If this list is not correct, click Back to make corrections.

- Process Center installation directory: C:\IBM\BPM\v8.5
- An existing DB2 database instance will be used for this server.

Use your support account to include updates with the [I don't have a support account.](#)

User Name:  Password:

I read and accepted the [license agreement](#) and [notices](#)

12. Click Install Software.

13. In the **Installation information** panel, select **Show passwords** to view the passwords that are used during the installation. The installation is now started and a status bar is displayed at the bottom of the panel. Depending on the resources, it might take a while for the installation to complete.

**Installation information**

While the installation is proceeding, print the following information for future reference.

Show passwords

**Process Center**  
Installation location: C:\IBM\BPM\v8.5

**Cell administrative account**  
User name: bpmadmin Password: web1sphere

**Deployment environment administrative account**  
User name: pcdeadmin Password: web1sphere

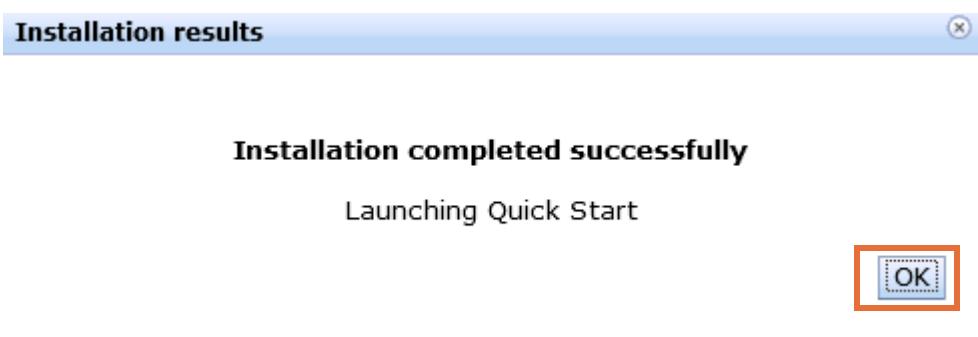
**Using existing DB2 database product**  
Host name: localhost Port: 50000  
User name: admin Password: Web1Sphere

Common database: PCCMNDB  
Process database: PCBPMDB  
Performance Data Warehouse database: PCPDWDB  
Cell only configuration database: PCCMNDB

Installing IBM Business Process Manager Advanced V8.5.5...

Cancel | 31%

14. When the installation completes, a message is displayed that the installation is successful. Verify that the message **Installation completed successfully** is displayed. Click **OK**.



15. Click **Exit** to close the launchpad.  
 16. Click **OK** to confirm.  
 17. If the IBM Business Process Manager Quick Start window opens, you can close it by clicking the **X** icon at the upper-right corner. Remember not to download any updates to the software as unpredictable results might occur.

## Part 3: Installing IBM Process Designer

The installation file for Process Designer is available for download from the Process Center Console. To do that, the Process Center server must be started.

- \_\_\_ 1. Start the IBM Process Center deployment manager.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Start the deployment manager**.
  - \_\_\_ b. A command window opens. The command window closes when the deployment manager starts.
- \_\_\_ 2. Start the IBM Process Center node agent.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Start the node agent**.
  - \_\_\_ b. A command window opens. The command window closes when the node agent starts.
- \_\_\_ 3. Start the cluster.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Administrative console**.
  - \_\_\_ b. Log in with `bpmadmin` as the **User ID** and `web1sphere` as the **Password**.
  - \_\_\_ c. Click **Servers > Deployment Environments**.
  - \_\_\_ d. Select the **ProcessCenter** check box and click **Start**. It takes several minutes to start.

Cell=PCCell1, Profile=DmgrProfile

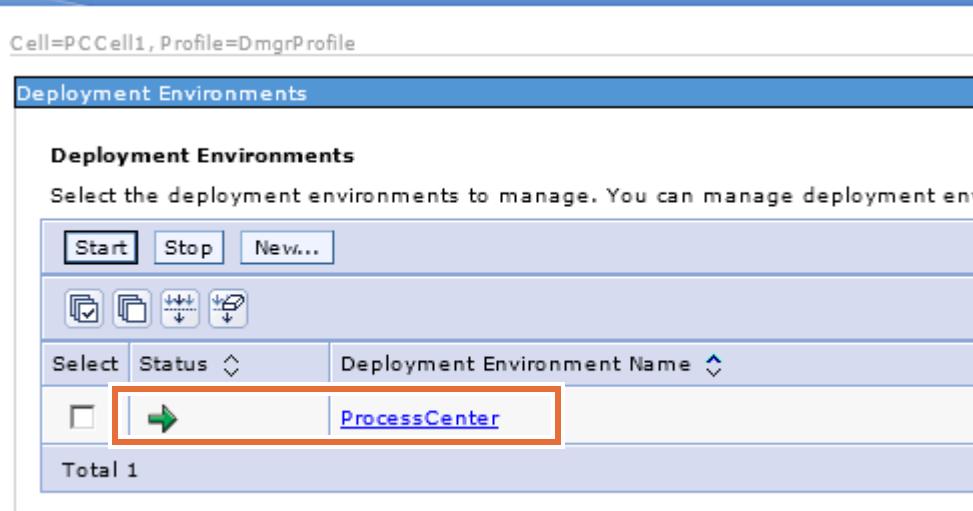
**Deployment Environments**

**Deployment Environments**  
Select the deployment environments to manage. You can manage deployment environment that are created using pattern

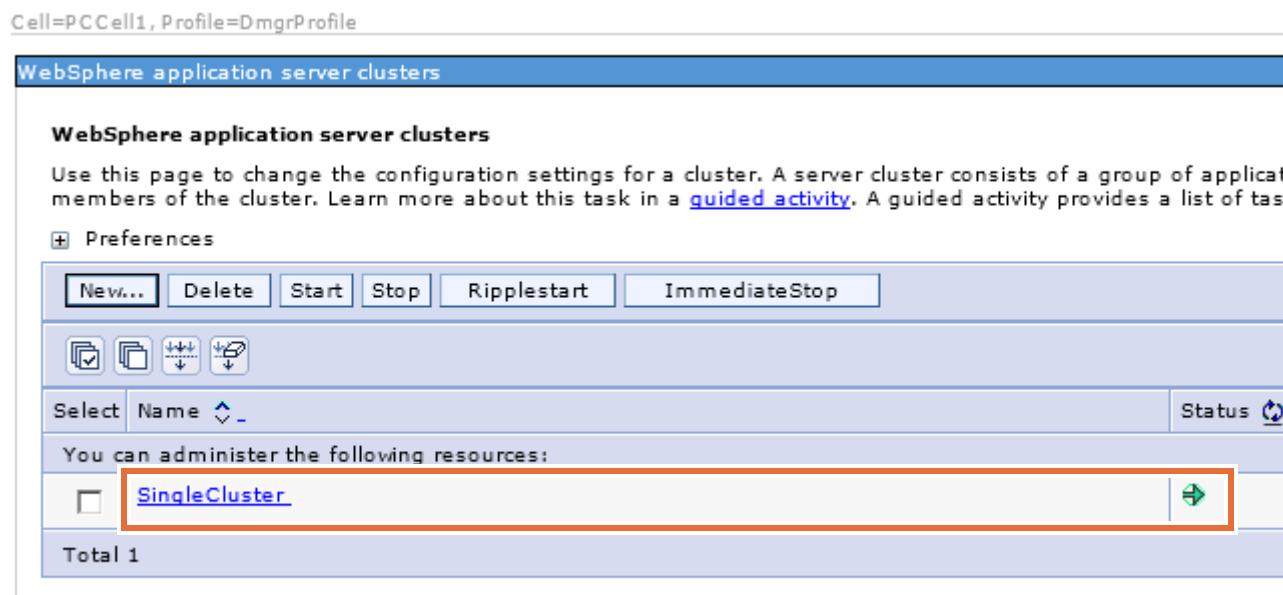
| Select                              | Status | Deployment Environment Name   | Features                        |
|-------------------------------------|--------|-------------------------------|---------------------------------|
| <input checked="" type="checkbox"/> |        | <a href="#">ProcessCenter</a> | IBM BPM Advanced Process Center |
| Total 1                             |        |                               |                                 |

**Start** **Stop** **New...**

- \_\_\_ e. When the Process Center starts, the Status icon turns green. Verify that the status is green. You can also press F5 a few times to refresh the console and go to **Servers > Deployment Environments** to view the status.



- \_\_\_ f. Click **Servers > Clusters > WebSphere application server clusters** to verify that the cluster is running.



- \_\_\_ g. Click **Logout** to exit.  
 \_\_\_ h. Close the browser.  
 \_\_\_ 4. Start Process Center Console.  
   \_\_\_ a. Start Internet Explorer and enter `http://localhost:9081/ProcessCenter` in the URL field.  
   \_\_\_ b. Enter `pcdeadmin` in the **User Name** field and `web1sphere` in the **Password** field.

- \_\_\_ 5. In the Process Center console, click **Download** at the lower-right corner.
- \_\_\_ 6. When prompted, save the **Process Designer.zip** file locally on your file system.
- \_\_\_ 7. Use 7-zip to extract the compressed file into an empty folder.
- \_\_\_ 8. Right-click **InstallProcessDesigner\_admin.bat** and click **Run as administrator** to start the installation.
- \_\_\_ 9. If you receive a security warning, then click **Run**.
- \_\_\_ 10. A command window opens and the installation starts.
- \_\_\_ 11. Press any key when the installation completes.
- \_\_\_ 12. Click **Start > IBM > IBM Process Designer > IBM Process Designer 8.5** to start Process Designer.
- \_\_\_ 13. Enter **pcdeadmin** in the **User Name** field and **web1sphere** in the **Password** field, and click **Logon**.
- \_\_\_ 14. If prompted with a security alert window, click **Yes** two times.
- \_\_\_ 15. You can close the Process Designer now that you are able to log in successfully. Installation of Process Designer is complete. When you are done, close all open windows.

#### **Part 4: Define users with Process Administrative Console**

- \_\_\_ 1. Start the administrative console.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Administrative Console**.
  - \_\_\_ b. Enter **bpmadmin** in the **User ID** field and **web1sphere** in the **Password** field and click **Login**.
- \_\_\_ 2. Create users.
  - \_\_\_ a. Expand **Users and Groups** and click **Manage Users**.
  - \_\_\_ b. Create the following users.

| User Name | Full Name      | Password |
|-----------|----------------|----------|
| afrank    | Frank Conrad   | password |
| ajack     | Jack Mall      | password |
| ajane     | Jane McCormick | password |
| ajohn     | John Miller    | password |
| amaria    | Maria Master   | password |
| abob      | Bob Smith      | password |

- \_\_\_ c. Log out of the administrative console
- \_\_\_ 3. Start the Process Admin Console.
  - \_\_\_ a. Start Internet Explorer and enter **http://localhost:9081/ProcessAdmin** in the URL field.
  - \_\_\_ b. When the Process Admin Console login page is displayed, enter **pcdeadmin** in the **User Name** field and enter **web1sphere** in the **Password** field.
  - \_\_\_ c. Click **Login**.

**Important**

Make sure that you log in by using `pcdeadmin` as the user, not the `bpmadmin` user. Recall that `pcdeadmin` is the deployment admin user that you created during the installation. For the Process Center labs in this course, you use the `pcdeadmin` user.

- \_\_\_ 4. At the upper right of the Process Admin Console, verify that `pcdeadmin` is listed as the user that is logged in.



- \_\_\_ 5. Expand User Management.



\_\_\_ 6. Click **User Management** to view users.

\_\_\_ a. Verify that the newly added users are listed in the view.

| Internal IBM BPM Users |  |
|------------------------|--|
| abob                   |  |
| sfrank                 |  |
| ajack                  |  |
| ajane                  |  |
| sjohn                  |  |
| amaria                 |  |
| bpmadmin               |  |
| podeadmin              |  |

\_\_\_ 7. Click **Group Management** to add new groups.

\_\_\_ a. Click **New Group**.

\_\_\_ b. Enter **GeneralManagers** as the **Name**, and click **Save**.

\_\_\_ c. Enter **GeneralManagers** in the **Select Group to Modify** field. The new group is listed at the end of this step.

\_\_\_ d. Select **GeneralManagers**, and click **Add Users**.

### User Management > Group Management

| New Group |                 | Remove |
|-----------|-----------------|--------|
|           | GeneralManagers |        |

**GeneralManagers**

Team Manager Group (deprecated): No Team Manager Group

[Add Users](#) [Add Groups](#)

\_\_\_ e. Enter **bob** in the **Search For Name** field.

\_\_\_ f. Make sure to select the check box for **Bob Smith (abob)**, and click **Add Selected**.

Now Bob Smith belongs to the General Managers group.

\_\_\_ 8. Repeat the previous step to create the following groups and then add members to the group.

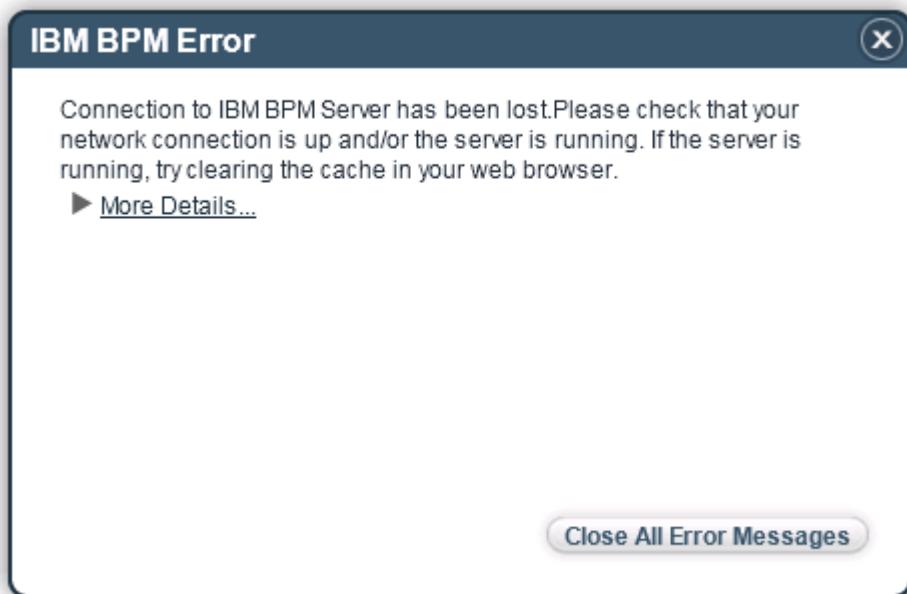
| Group Name     | Member(s)                              |
|----------------|----------------------------------------|
| HiringManagers | Jack Mall, Jane McCormick, John Miller |
| HRDepartment   | Frank Conrad, Maria Master             |
| HRExperts      | Maria Master                           |

\_\_\_ 9. Log out of the Process Admin console and close the browser.



## Information

While working with the Process Designer or Process Center console, you might receive an IBM Business Process Manager error message that the connection to the server is lost. If you get that message, do not be concerned. Click **Close All Error Messages** each time you get that message, and you can continue with your work. It is possible for the session to time out, and you might be prompted to log back in to the console. If prompted, log back in to the Process Console or Process Designer.



## **Part 5: Stop the Process Center environment**

- \_\_\_ 1. Stop the deployment environment.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Administrative console**.
  - \_\_\_ b. Log in with `bpmadmin` as the **User ID** and `web1sphere` as the **Password**.
  - \_\_\_ c. Select **Servers > Deployment Environments**.
  - \_\_\_ d. Select the **ProcessCenter** check box and click **Stop**. It takes several minutes to stop.
  - \_\_\_ e. Log out of the administrative console and close the browser.
- \_\_\_ 2. Stop IBM Process Center node agent.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Stop the node agent**.
  - \_\_\_ b. A command window opens. When prompted, enter `bpmadmin` in the **User Identity** field and `web1sphere` in the **User Password** fields. The command window closes in a few minutes when the node agent stops.

- \_\_\_ 3. Stop the IBM Process Center deployment manager.
  - \_\_\_ a. Click **Start** and then click **IBM** under **Apps**. Then, click **Stop the deployment manager**.
  - \_\_\_ b. A command window opens. When prompted, enter `bpmadmin` in the **User Identity** field and `web1sphere` in the **User Password** field. The command window closes in a few minutes when the deployment manager stops.

## Part 6: Download and install Adobe Flash player

- \_\_\_ 1. Start Internet Explorer and go to: <http://get2.adobe.com/flashplayer/>



### Note

This link is current as of writing of this course. The link might change at a future date.

- \_\_\_ 2. Clear the option to install any optional offers.
- \_\_\_ 3. Click **Install Now**.
- \_\_\_ 4. If prompted, click **Run**.
- \_\_\_ 5. In the **Update Flash Player Preferences** window, click the **Never check for updates (not recommended)** option and click **Next**.
- \_\_\_ 6. When the installation completes, click **Finish** to close the window.

## Part 7: Download and install Adobe Reader

- \_\_\_ 1. Start Internet Explorer and go to: <http://get2.adobe.com/reader/>



### Note

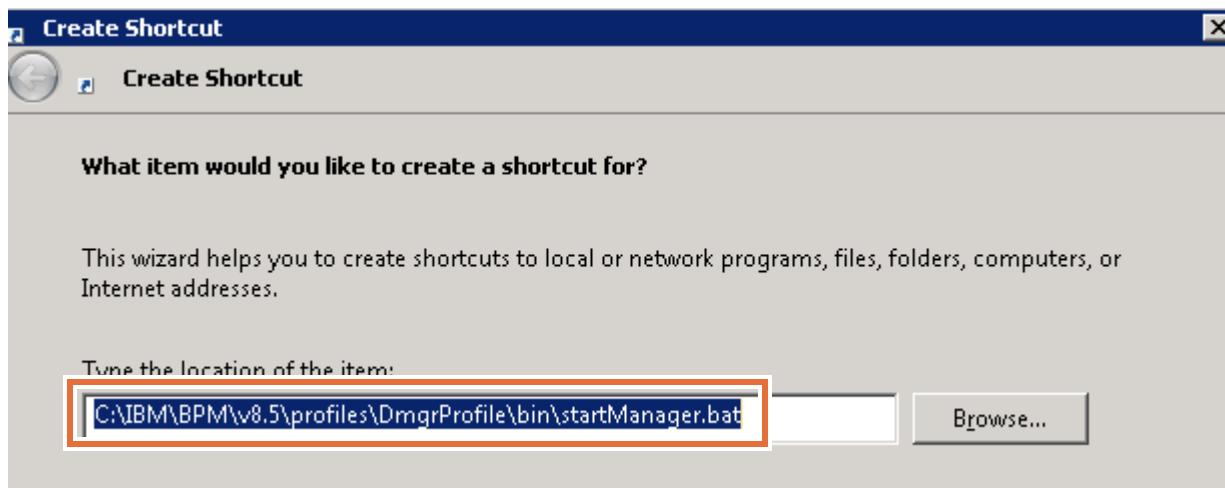
This link is current as of writing of this course. The link might change at a future date.

- \_\_\_ 2. Clear the option to install the optional offers.
- \_\_\_ 3. Click **Install Now**.
- \_\_\_ 4. Click **Run**.
- \_\_\_ 5. Click **Next**.
- \_\_\_ 6. When the installation completes, click **Finish** to close the window.

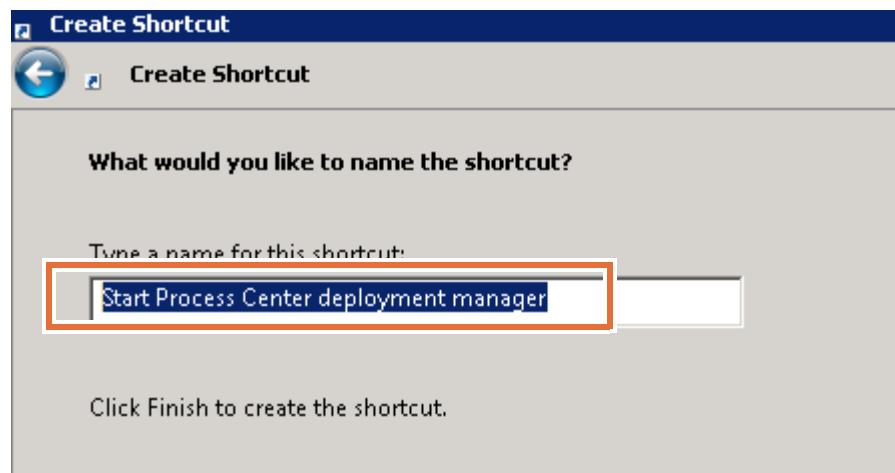
## Additional setup

### Part 8: Create desktop shortcuts for starting the Process Center environment

- \_\_\_ 1. Create a shortcut to start the Process Center deployment manager.
  - \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
  - \_\_\_ b. In the Create Shortcut window, click **Browse** and go to:  
C:\IBM\BPM\v8.5\profiles\dmgrProfile\bin\
  - \_\_\_ c. Select `startManager.bat` and click **OK**.

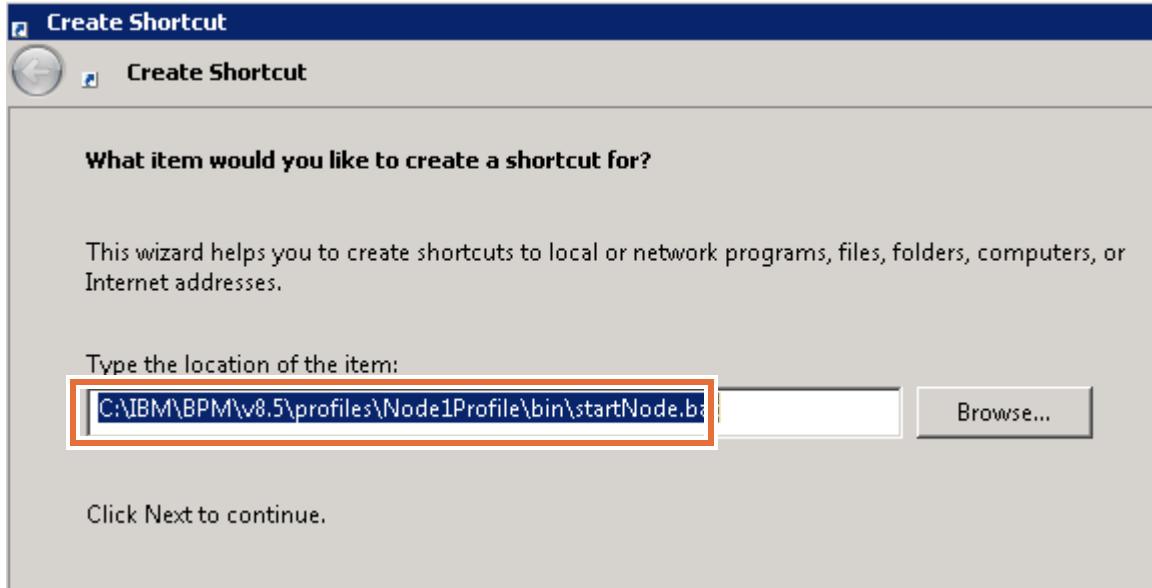


- \_\_\_ d. Click **Next**.
- \_\_\_ e. Enter Start Process Center deployment manager for the shortcut name and click **Finish**.



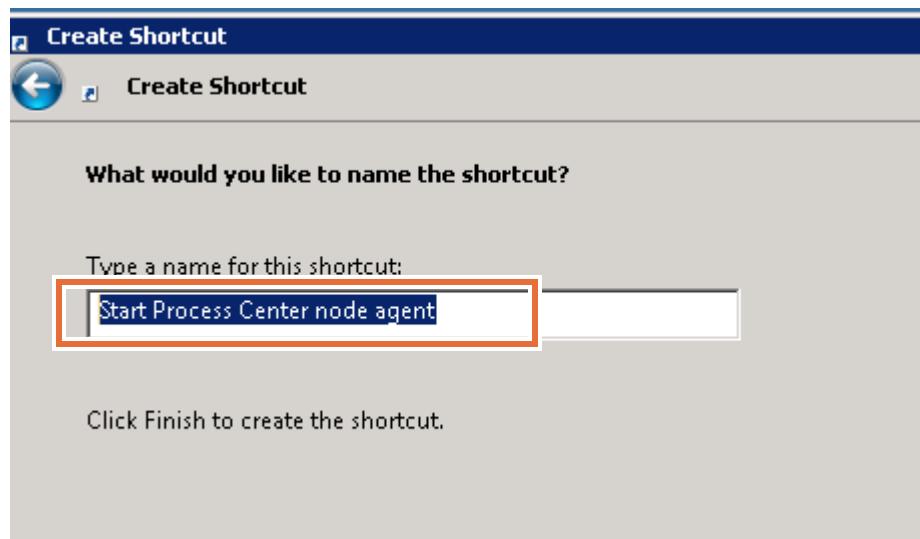
- \_\_\_ f. The new shortcut is displayed on the desktop. Do not start the server now. You are going to do that later.
- \_\_\_ 2. Create a shortcut to stop the Process Center deployment manager.
  - \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.

- \_\_\_ b. In the Create Shortcut window, click **Browse** and go to:  
C:\IBM\BPM\v8.5\profiles\dmgrProfile\bin\
  - \_\_\_ c. Select `stopManager.bat` and click **OK**.
  - \_\_\_ d. Click **Next**.
  - \_\_\_ e. Enter **Stop Process Center deployment manager** for the shortcut name and click **Finish**.
  - \_\_\_ f. The new shortcut is displayed on the desktop.
- \_\_\_ 3. Create a shortcut to start the Process Center node agent.
- \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
  - \_\_\_ b. In the Create Shortcut window, click **Browse** and go to:  
C:\IBM\BPM\v8.5\profiles\Node1Profile\bin\
  - \_\_\_ c. Select `startNode.bat` and click **OK**.



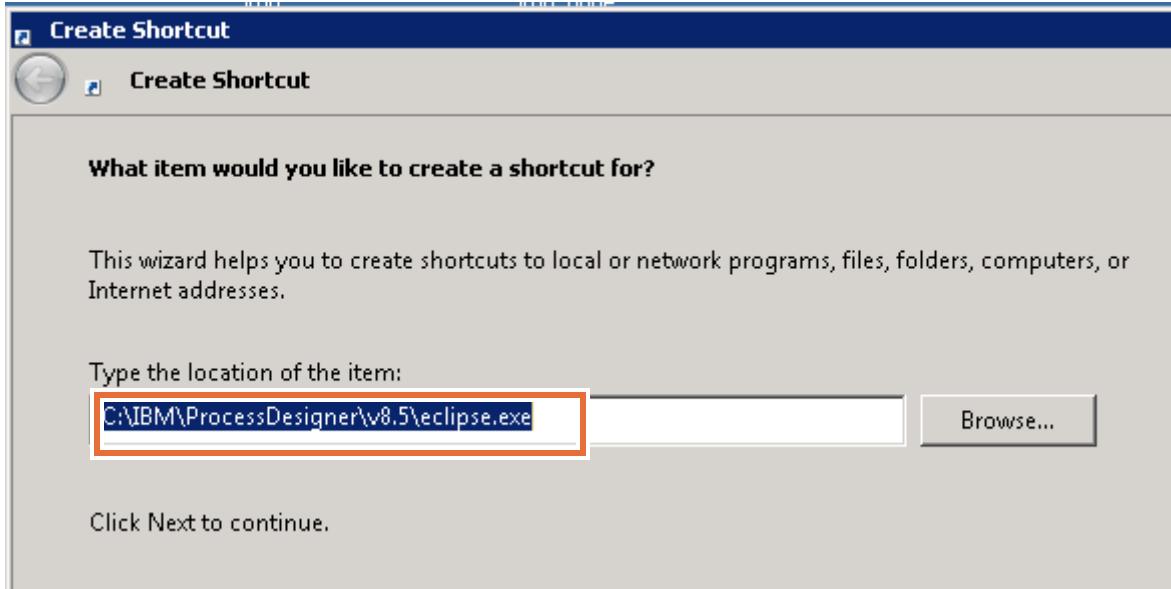
- \_\_\_ d. Click **Next**.

- \_\_\_ e. Enter Start Process Center node agent for the shortcut name and click **Finish**.



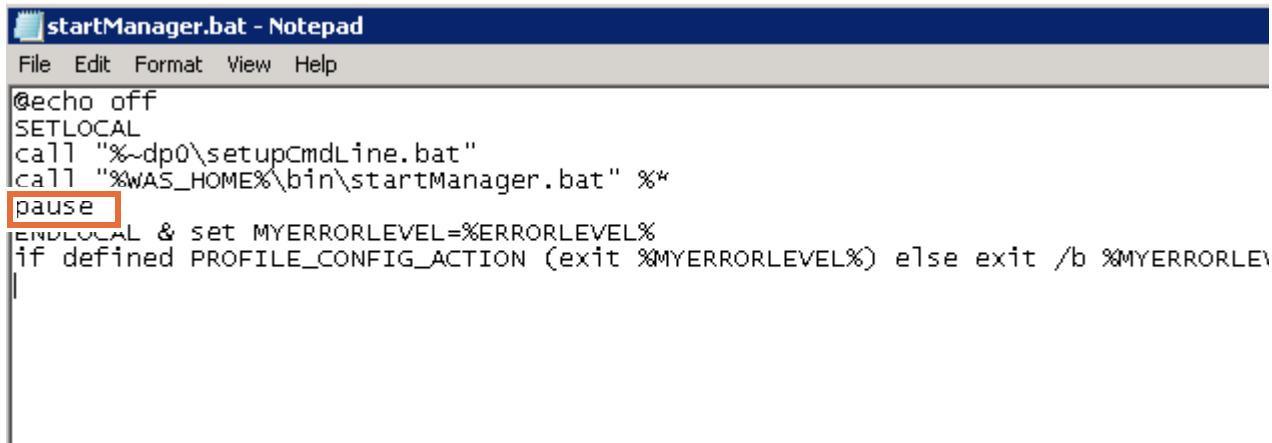
- \_\_\_ f. The new shortcut is displayed on the desktop. Do not start the server now. You are going to do that later.
- \_\_\_ 4. Create a shortcut to stop the Process Center node agent.
- \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
- \_\_\_ b. In the Create Shortcut window, click **Browse** and go to:  
C:\IBM\BPM\v8.5\profiles\Node1Profile\bin\
- \_\_\_ c. Select `stopNode.bat` and click **OK**.
- \_\_\_ d. Click **Next**.
- \_\_\_ e. Enter stop Process Center node agent for the shortcut name and click **Finish**.
- \_\_\_ f. The new shortcut is displayed on the desktop.
- \_\_\_ 5. Create a shortcut to start the Process Center cluster.
- \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
- \_\_\_ b. In the Create Shortcut window, in the "Type the location of the item field", enter:  
C:\IBM\BPM\v8.5\profiles\Node1Profile\bin\startServer.bat  
SingleClusterMember1
- \_\_\_ c. Click **Next**.
- \_\_\_ d. Enter start Process Center Cluster for the shortcut name and click **Finish**.
- \_\_\_ 6. Create a shortcut to stop the Process Center cluster.
- \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
- \_\_\_ b. In the Create Shortcut window, in the "Type the location of the item field", enter:  
C:\IBM\BPM\v8.5\profiles\Node1Profile\bin\stopServer.bat  
SingleClusterMember1 -username bpmadmin -password web1sphere
- \_\_\_ c. Click **Next**.
- \_\_\_ d. Enter Stop Process Center Cluster for the shortcut name and click **Finish**.

- \_\_\_ e. The new shortcut is displayed on the desktop.
- \_\_\_ 7. Create a shortcut to start IBM Process Designer.
  - \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
  - \_\_\_ b. In the Create Shortcut window, click **Browse**, go to `C:\IBM\ProcessDesigner\v8.5\eclipse.exe` and click **OK**.



- \_\_\_ c. Click **Next**.
- \_\_\_ d. Enter **Start IBM Process Designer** for the shortcut name and click **Finish**.
- \_\_\_ e. The new shortcut is displayed on the desktop. Do not start the server now. You are going to do that later. However, feel free to rearrange the shortcuts on the desktop to better organize them.
- \_\_\_ 8. Create a shortcut to start the IBM Integration Designer test server.
  - \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
  - \_\_\_ b. In the Create Shortcut window, in the "Type the location of the item field", enter:  
`C:\IBM\IID\PS\v8.5\profiles\qbpmaps\bin\startServer.bat server1`
  - \_\_\_ c. Click **Next**.
  - \_\_\_ d. Enter **Start UTE Process Server** for the shortcut name and click **Finish**.
  - \_\_\_ e. The new shortcut is displayed on the desktop.
- \_\_\_ 9. Create a shortcut to stop the IBM Integration Designer test server.
  - \_\_\_ a. Right-click anywhere on the desktop and click **New > Shortcut**.
  - \_\_\_ b. In the Create Shortcut window, in the "Type the location of the item field", enter:  
`C:\IBM\IID\PS\v8.5\profiles\qbpmaps\bin\stopServer.bat server1 -username admin -password web1sphere`
  - \_\_\_ c. Click **Next**.
  - \_\_\_ d. Enter **Stop UTE Process Server** for the shortcut name and click **Finish**.

- e. The new shortcut is displayed on the desktop.
- f. Feel free to rearrange the shortcuts on the desktop to better organize them.
- 10. Edit the batch scripts to add a pause command. It is a good idea to edit the batch script so that the command window does not close automatically when the server is started.
  - a. Right-click the shortcut to start Process Center deployment manager and click **Edit**.
  - b. Put the cursor at the end of the fourth line after `%*` and press Enter.
  - c. Type `pause` in the newly inserted blank line.



The screenshot shows a Windows Notepad window titled "startManager.bat - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main content area contains the following batch script code:

```

@echo off
SETLOCAL
call "%~dp0\setupCmdLine.bat"
call "%WAS_HOME%\bin\startManager.bat" %*
pause
ENDLOCAL & set MYERRORLEVEL=%ERRORLEVEL%
if defined PROFILE_CONFIG_ACTION (exit %MYERRORLEVEL%) else exit /b %MYERRORLEVEL%

```

A red box highlights the word "pause" on the fourth line of the script.

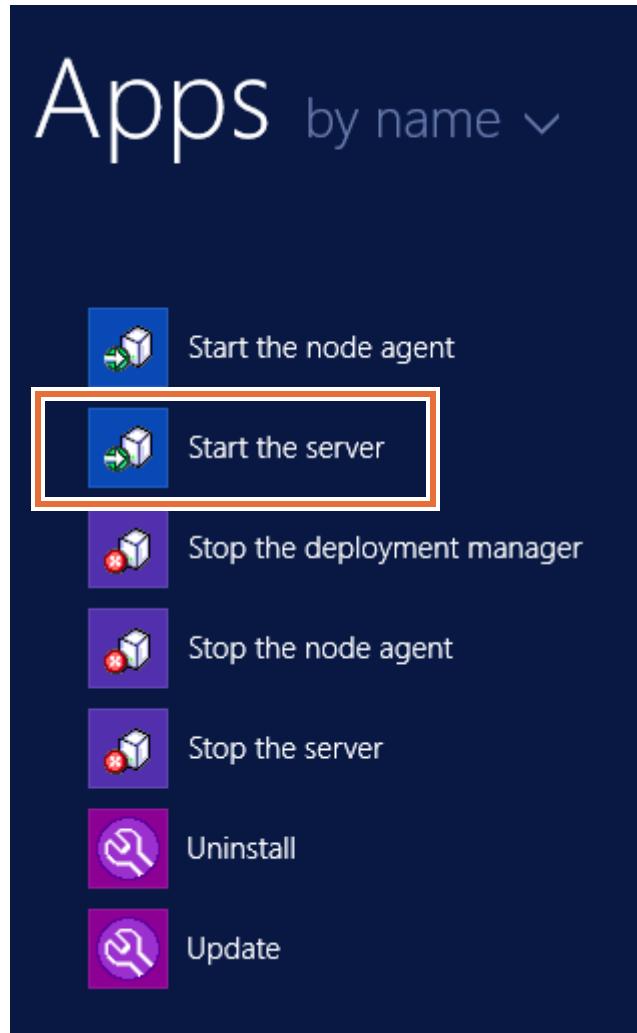
- d. Save your changes and close the batch file.
- 11. Repeat the steps to add the pause command for the shortcuts for stopping the deployment manager, starting and stopping the node agent, and starting and stopping the cluster batch scripts.

### **Part 9: Extending the default one-year expiration date for WebSphere certificate (Optional)**

By default the WebSphere certificate is set to expire in one year. While the certificates self-extend when the expiration date approaches, several SSL warnings are displayed when it happens. If you do not intend to keep this environment for a long period such as nine months or longer, then you do not need to take the steps in this section. However, if you plan to keep this environment available

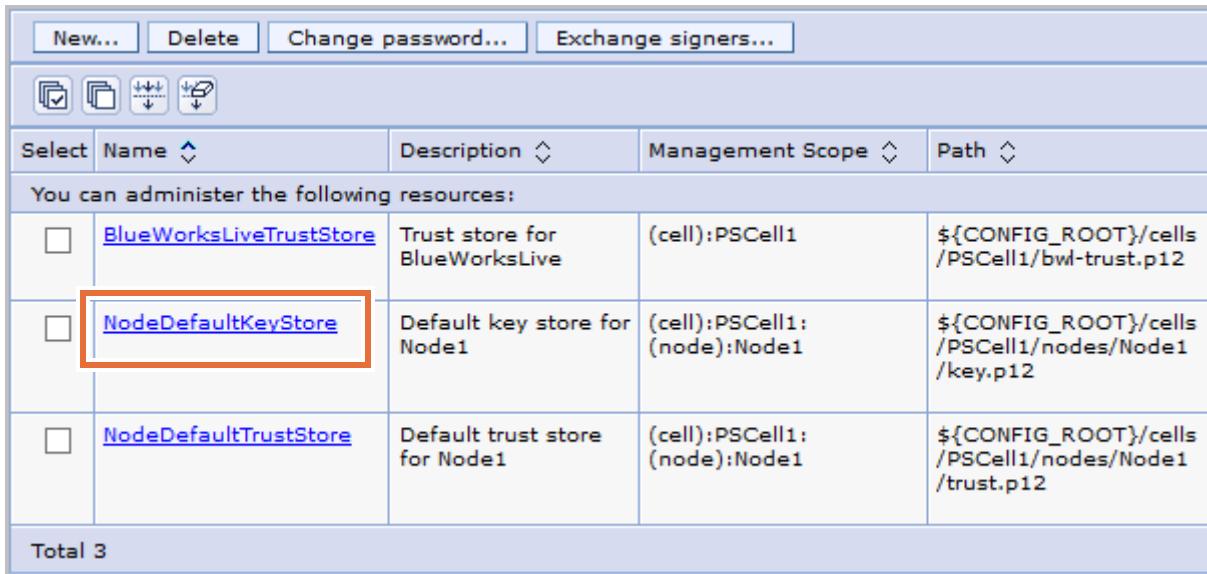
for a year or longer, then you might want to consider extending the certificate manually as described in this section.

- \_\_\_ 1. Extend the certificate for the IBM Process Server.
  - \_\_\_ a. If the IBM Process Server is not running, then start it by clicking **Start** and then under **IBM**, click **Start the server**.



- \_\_\_ b. A command window opens while the server is starting. When the server starts, the command window closes.
- \_\_\_ c. Start Firefox and go to: <http://localhost:9060/ibm/console>
- \_\_\_ d. Log in with `admin` as the **User ID** and `web1sphere` as the **Password**.
- \_\_\_ e. Select **Security > SSL certificate and key management**.
- \_\_\_ f. In the “SSL certificate and key management” panel, click **Key stores and certificates**.

- \_\_\_ g. Click the **NodeDefaultKeyStore** link.



| <input type="button" value="New..."/> <input type="button" value="Delete"/> <input type="button" value="Change password..."/> <input type="button" value="Exchange signers..."/><br><input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |                                         |                               |                             |                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-------------------------------|-----------------------------|----------------------------------------------------|
| Select                                                                                                                                                                                                                                                                                                           | Name                                    | Description                   | Management Scope            | Path                                               |
| You can administer the following resources:                                                                                                                                                                                                                                                                      |                                         |                               |                             |                                                    |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                         | <a href="#">BlueWorksLiveTrustStore</a> | Trust store for BlueWorksLive | (cell):PSCell1              | #{CONFIG_ROOT}/cells/PSCell1/bwl-trust.p12         |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                         | <a href="#">NodeDefaultKeyStore</a>     | Default key store for Node1   | (cell):PSCell1:(node):Node1 | #{CONFIG_ROOT}/cells/PSCell1/nodes/Node1/key.p12   |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                         | <a href="#">NodeDefaultTrustStore</a>   | Default trust store for Node1 | (cell):PSCell1:(node):Node1 | #{CONFIG_ROOT}/cells/PSCell1/nodes/Node1/trust.p12 |
| Total 3                                                                                                                                                                                                                                                                                                          |                                         |                               |                             |                                                    |

- \_\_\_ h. Click **Personal Certificates**.

- \_\_\_ i. Notice that the certificate expiration date for the node is one year from the date that the software was installed.



| <input type="button" value="Export..."/> <input type="button" value="Revoke..."/> <input type="button" value="Renew"/> |               |                                        |
|------------------------------------------------------------------------------------------------------------------------|---------------|----------------------------------------|
|                                                                                                                        | Serial Number | Expiration                             |
| de1,                                                                                                                   | 6576655584183 | Valid from Mar 8, 2016 to Mar 8, 2017. |
| de1,                                                                                                                   | 6573864856643 | Valid from Mar 8, 2016 to Mar 5, 2031. |

- \_\_\_ j. Select the text that is listed after **CN=** under the **Issued To** column next to the **default Alias**, right-click, and then click **Copy** to copy the text to the clipboard.

The screenshot shows a table of certificates. A context menu is open over the first row, which contains the alias "default". The menu items are: Copy, Select All, Search Yahoo for "ws2012r2x64, OU...", This Frame, View Selection Source, and Inspect Element (Q). The "Copy" option is highlighted.

| Issued To               |                                                            | Issued By                                                  | Serial Number | Expiration                                      |
|-------------------------|------------------------------------------------------------|------------------------------------------------------------|---------------|-------------------------------------------------|
| <a href="#">default</a> | CN=ws2012r2x64,<br>OU=PSCell1,<br>O=IBM, C=U               | CN=ws2012r2x64,<br>OU=Root Ce<br>OU=PSCell1,<br>O=IBM, C=U | 6576655584183 | Valid from<br>Mar 8, 2016<br>to Mar 8,<br>2017. |
|                         | CN=ws2012r2x64,<br>OU=Root Ce<br>OU=PSCell1,<br>O=IBM, C=U |                                                            | 73864856643   | Valid from<br>Mar 8, 2016<br>to Mar 5,<br>2031. |

- \_\_\_ k. Click **Create > Chained Certificate**.

The screenshot shows the "SSL certificate and key management" interface. In the top navigation bar, "SSL certificate and key management > Key stores and certificates" is selected. Below it, a message says "Manages personal certificates." There is a "+" button for "Preferences". The main area has a "Create" dropdown with three options: "Self-signed Certificate...", "CA-signed Certificate...", and "Chained Certificate...". The "Chained Certificate..." option is highlighted. Below the dropdown, there is a table with two rows. The first row has a checkbox, a green lock icon, and the alias "default". The second row has a green lock icon and an upward arrow icon. At the bottom, it says "Total 2".

- \_\_\_ l. Enter **d1** in the **Alias** field.  
 \_\_\_ m. Paste the text that was saved to the clipboard into the **Common name** field. You can do that by right-clicking in the **Common name** field and clicking **Paste**.  
 \_\_\_ n. Change the **Validity period** to **3650** by adding a zero at the end.

- \_\_\_ o. Click **OK** and save your changes.

SSL certificate and key management

[SSL certificate and key management](#) > [Key stores and certificates](#) > New...

A chained personal certificate is a personal certificate that is created using the certificate chain feature.

**General Properties**

\* Alias

Root certificate used to sign the certificate

Key size

\* Common name

\* Validity period  
 days

Organization

Organization unit

Locality

State/Province

Zip code

Country or region

- \_\_\_ p. The new **d1** certificate is now listed.
- \_\_\_ q. Select the default certificate and click **Replace**.

- \_\_\_ r. Select the two options to delete the old certificate and old signers, and click **OK** and save.

[SSL certificate and key management > Key stores and certificates > !](#)

Replaces a certificate with a new certificate. Also replaces signer certificates.

**General Properties**

Old certificate  
default

Replace with  
d1

Delete old certificate after replacement

Delete old signers

**Buttons:** Apply | OK | Reset | Cancel

- \_\_\_ s. Verify that the only certificate that is listed now is the **d1** certificate.
- \_\_\_ t. Click **Create > Chained certificate** to create another certificate.
- \_\_\_ u. Enter **default** in the **Alias** field.
- \_\_\_ v. Paste the same text as you did earlier from the clipboard in the **Common name** field by right-clicking in the field and clicking **Paste**.
- \_\_\_ w. Change the **Validity period** to **3650** and click **OK**.
- \_\_\_ x. Select the **d1** certificate and click **Replace**.
- \_\_\_ y. Select the two options to delete the old certificate and old signers, and click **OK**.

[SSL certificate and key management > Key stores and certificates >](#)

Replaces a certificate with a new certificate. Also replaces signer certificates.

**General Properties**

Old certificate  
d1

Replace with  
default

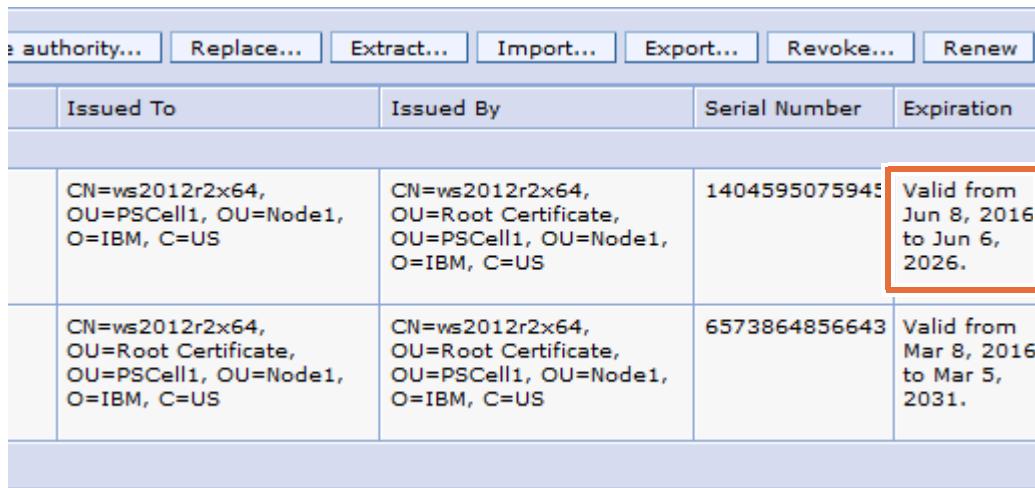
Delete old certificate after replacement

Delete old signers

**Buttons:** Apply | OK | Reset | Cancel

- \_\_\_ z. Click **Save** to save the changes to the master configuration.

- \_\_\_ aa. Click **OK**. Notice that the default certificate that had a one-year expiration date is now set to expire in 10 years.



| <input type="button" value="New authority..."/> <input type="button" value="Replace..."/> <input type="button" value="Extract..."/> <input type="button" value="Import..."/> <input type="button" value="Export..."/> <input type="button" value="Revoke..."/> <input type="button" value="Renew"/> |                                                                                 |                                                                                 |               |                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------|-------------------------------------------------|
|                                                                                                                                                                                                                                                                                                     | Issued To                                                                       | Issued By                                                                       | Serial Number | Expiration                                      |
|                                                                                                                                                                                                                                                                                                     | CN=ws2012r2x64,<br>OU=PSCell1, OU=Node1,<br>O=IBM, C=US                         | CN=ws2012r2x64,<br>OU=Root Certificate,<br>OU=PSCell1, OU=Node1,<br>O=IBM, C=US | 1404595075945 | Valid from<br>Jun 8, 2016<br>to Jun 6,<br>2026. |
|                                                                                                                                                                                                                                                                                                     | CN=ws2012r2x64,<br>OU=Root Certificate,<br>OU=PSCell1, OU=Node1,<br>O=IBM, C=US | CN=ws2012r2x64,<br>OU=Root Certificate,<br>OU=PSCell1, OU=Node1,<br>O=IBM, C=US | 6573864856643 | Valid from<br>Mar 8, 2016<br>to Mar 5,<br>2031. |

- \_\_\_ 2. Extend the certificate for the Process Center deployment manager.
- \_\_\_ a. Double-click the **Start Process Center deployment manager** shortcut. A DOS command window is displayed. When the deployment manager starts, press any key to close the window.
- \_\_\_ b. Double-click the **Start Process Center node agent** shortcut. A DOS command window is displayed. When the deployment manager starts, press any key to close the window.
- \_\_\_ c. Double-click the **Start Process Center cluster** shortcut. A DOS command window is displayed. When the deployment manager starts, press any key to close the window.
- \_\_\_ d. Click **Start** and then under **IBM**, click **Administrative console**.
- \_\_\_ e. Log in with `bpmadmin` as the **User ID** and `websphere` as the **Password**.
- \_\_\_ f. Select **Security > SSL certificate and key management**.
- \_\_\_ g. In the “SSL certificate and key management” panel, click **Key stores and certificates**.

- \_\_\_ h. You must extend the certificate for **CellDefaultKeyStore** and **NodeDefaultKeyStore** for the Process Center server. Refer to the previous steps that you followed to extend the certificate for the Process Server.

| <a href="#">New...</a> <a href="#">Delete</a> <a href="#">Change password...</a> <a href="#">Exchange signers...</a> |                                         |                                 |                             |                                                      |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------|---------------------------------|-----------------------------|------------------------------------------------------|
| Select                                                                                                               | Name                                    | Description                     | Management Scope            | Path                                                 |
| You can administer the following resources:                                                                          |                                         |                                 |                             |                                                      |
| <input type="checkbox"/>                                                                                             | <a href="#">BlueWorksLiveTrustStore</a> | Trust store for BlueWorksLive   | (cell):PCCell1              | `\${CONFIG_ROOT}/cells/PCCell1/bwl-trust.p12         |
| <input type="checkbox"/>                                                                                             | <a href="#">CellDefaultKeyStore</a>     | Default key store for PCCell1   | (cell):PCCell1              | `\${CONFIG_ROOT}/cells/PCCell1/key.p12               |
| <input type="checkbox"/>                                                                                             | <a href="#">CellDefaultTrustStore</a>   | Default trust store for PCCell1 | (cell):PCCell1              | `\${CONFIG_ROOT}/cells/PCCell1/trust.p12             |
| <input type="checkbox"/>                                                                                             | <a href="#">NodeDefaultKeyStore</a>     | Default key store for Node1     | (cell):PCCell1:(node):Node1 | `\${CONFIG_ROOT}/cells/PCCell1/nodes/Node1/key.p12   |
| <input type="checkbox"/>                                                                                             | <a href="#">NodeDefaultTrustStore</a>   | Default trust store for Node1   | (cell):PCCell1:(node):Node1 | `\${CONFIG_ROOT}/cells/PCCell1/nodes/Node1/trust.p12 |
| Total 5                                                                                                              |                                         |                                 |                             |                                                      |

## Installing the lab files

Perform the following steps to install the lab files that are provided with the course.

The course environment depends upon a number of prebuilt support files. These files are necessary to complete the course exercises as they are designed. WebSphere Education also offers a complete set of solution files, represented as project interchange files, which can be imported to IBM Integration Designer.

When you obtain the solution and support files from WebSphere Education:

- \_\_\_ 1. The solution and support files are on the Instructor CD in the file that is called `WB860_ERC1.0_LABFILES.zip`. Extract the contents of the compressed file from the Instructor CD to the host system root.
- \_\_\_ 2. Verify that after extraction of the contents, a folder that is named `labfiles` is created with two subfolders in it named `Support files` and `Solutions`.



### Note

At any point during the course, you can install the solution files as a means of helping the students keep pace with the rest of the class.

## Postinstallation tasks

Perform the following steps to complete the installation and customize the products.

### Creating default workspaces and shortcuts

As part of the instructions in the course exercises, students are asked to open default workspaces, such as `C:\Workspaces\Ex10`. If you choose to use these default workspaces, they must be created. They offer an easy way for students to all begin at the same starting point with no exceptions, which can accumulate between exercises.

- \_\_\_ 1. Create a directory in the host system root that is named `Workspaces`, for example: `C:\Workspaces`
- \_\_\_ 2. Start IBM Integration Designer and open it to a sample workspace, for example: `C:\sample`  
Make sure that the **Use this as the default and do not ask again** check box is **not** selected.
- \_\_\_ 3. When a sample workspace is created, close IBM Integration Designer.
- \_\_\_ 4. In the directory where the sample workspace was created, copy the entire workspace and paste it into `C:\Workspaces` for a total of 13 times. You need 13 workspaces, or one for each lab exercise (except Exercises 1, 2, and 15), to reproduce the environment.
- \_\_\_ 5. Rename the workspace directories from `Ex3` to `Ex16`, inclusive.
- \_\_\_ 6. Rename the `Ex4` workspace to: `Ex4b`
- \_\_\_ 7. Open IBM Integration Designer on each workspace in turn and import the project interchange file from the support files directory that represents the start of the exercise. For example, open IBM Integration Designer on `C:\Workspaces\Ex3` and import the `C:\labfiles\Support Files\EX3_START.zip` project interchange file. Repeat this process for all the exercises. You can switch workspaces by clicking **File > Switch Workspace > Other**.
- \_\_\_ 8. In the desktop, create a folder that is named: `Exercise Shortcuts`
- \_\_\_ 9. In this directory, create a shortcut to each of the prebuilt workspaces. Ensure that the shortcut has following properties:

```
<Integration_Designer_Install>\wid.exe -product
com.ibm.wbit.feature.ide -data "C:\Workspaces\<lab>"
```

`<Integration_Designer_Install>` is the full class path where IBM Integration Designer was installed and `<lab>` expresses the workspace corresponding to the exercise number.

### Import process applications for the lab

- \_\_\_ 1. Start IBM Process Designer and log in with the `pcdeadmin` user.

**Important**

Make sure that you log in by using `pcdeadmin` as the user, not the `bpmadmin` user. Recall that `pcdeadmin` is the deployment admin user that you created during the installation.

- \_\_\_ 2. After logging in Process Designer, click the **Process Apps** tab.
- \_\_\_ 3. Import the **Account Verification** process app.
  - \_\_\_ a. Click **Import Process App** on the upper-right panel.
  - \_\_\_ b. Click **Browse** to go to folder `C:\Support Files\Ex1` and select the file:  
`Account_Verification_Skeleton - Lab_1_Start.twx`
  - \_\_\_ c. Click **Open** to return to Import Process App.
  - \_\_\_ d. Click **OK**.
  - \_\_\_ e. Click **Import**.
  - \_\_\_ f. The Account Verification Skeleton (AVS) is imported successfully and is displayed in the **Process Apps** tab. It might take few minutes for the application to import.
- \_\_\_ 4. Import the **Hiring Sample** process app.
  - \_\_\_ a. Click **Import Process App** on the upper-right panel.
  - \_\_\_ b. Click **Browse** to go to folder `C:\Support Files\Ex1` and select the  
`AHS_GOES_BUE.twx` file.
  - \_\_\_ c. Click **Open** to return to Import Process App.
  - \_\_\_ d. Click **OK**.
  - \_\_\_ e. Click **Import**.
  - \_\_\_ f. The Hiring Sample (HSAV1) is imported successfully and is displayed in the **Process Apps** tab.
- \_\_\_ 5. Click **File > Exit** to close the IBM Process Designer.

## Verification procedures

Use the following information to verify the installation and configuration of the student and instructor lab environments.

- \_\_\_ 1. Start the IBM Process Center server. Refer to the steps listed earlier on how to start the server.
- \_\_\_ 2. Start IBM Process Designer, and log in by entering `pceadmin` in the **User Name** and **Password** fields. If you arrive at the Process Center perspective in IBM Process Designer, then IBM Process Center and IBM Process Designer are installed correctly.
- \_\_\_ 3. Stop the IBM Process Center server. Refer to the steps listed earlier on how to start the server.
- \_\_\_ 4. Start IBM Integration Designer. If you arrive at the default workspace for IBM Integration Designer, then IBM Process Server and IBM Integration Designer are installed correctly.
- \_\_\_ 5. Stop IBM Process Server and close the IBM Process Designer.

## Verification procedures

Use the following information to verify the installation and configuration of the student and instructor lab environments.

- \_\_\_ 1. Start the IBM Process Center server. Refer to the steps listed earlier on how to start the server.
- \_\_\_ 2. Start IBM Process Designer, and log in by entering `pceadmin` in the **User Name** and **Password** fields. If you arrive at the Process Center perspective in IBM Process Designer, then IBM Process Center and IBM Process Designer are installed correctly.
- \_\_\_ 3. Stop the IBM Process Center server. Refer to the steps listed earlier on how to start the server.
- \_\_\_ 4. Start IBM Integration Designer. If you arrive at the default workspace for IBM Integration Designer, then IBM Process Server and IBM Integration Designer are installed correctly.
- \_\_\_ 5. Stop IBM Process Server and close IBM Integration Designer.

# Appendix B. Creating the artifacts that are imported in the exercises

You can use IBM Integration Designer to create, in your own environment, the artifacts that are described in this appendix. The Hiring Solution Sample model that is used in Exercise 1 is a clone of an existing process application of the same name. The account verification model that is used in Exercise 1 is not included in this appendix.

## **Part 1: Creating the credit score web service**

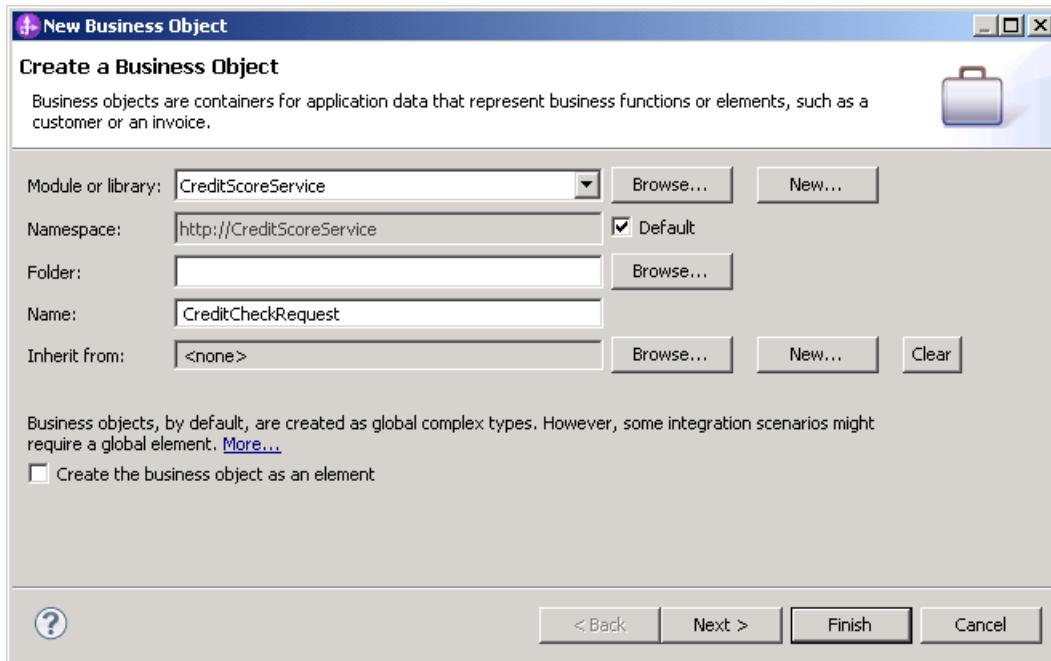
The credit score web service is an IBM Process Server integration module that contains business rules that return a credit score, which is based on the company name. For simplicity, the credit score that is returned is fixed for each company name.

To create the credit score web service:

- \_\_\_ 1. Create an integration module to house the web service.
  - \_\_\_ a. Click **File > New > Module** from the menu options.
  - \_\_\_ b. In the New Module dialog box, enter the following information:
    - Type `CreditScoreService` in the **Module name** field.
    - Leave **Use default location** selected.
    - (Optional) Leave **Open module assembly diagram** selected.
  - \_\_\_ c. Click **Finish**.
- \_\_\_ 2. Create a CreditCheckRequest business object that the service uses.
  - \_\_\_ a. Expand **CreditScoreService** and right-click **Data**.
  - \_\_\_ b. Click **New > Business Object**.

\_\_\_ c. In the New Business Object dialog box, enter the following information:

- In the **Module or library** field, leave **CreditScoreService** selected.
- Accept the default **Namespace**.
- Leave the **Folder** field blank.
- In the **Name** field, type: **CreditCheckRequest**
- Leave the **Inherit from** field empty.



\_\_\_ d. Click **Finish**.

\_\_\_ e. In the business object editor, add the following fields and their corresponding data types.

| Field name    | Data type |
|---------------|-----------|
| accountNumber | string    |
| companyName   | string    |
| creditScore   | int       |
| dateRequested | string    |

\_\_\_ f. The completed business object definition resembles the following image:



\_\_\_ g. Save your changes.

\_\_\_ 3. Create a CreditScoreService WSDL interface that is used by the service.

\_\_\_ a. Expand **CreditScoreService** and right-click **Interfaces**.

- \_\_\_ b. Click **New > Interface**.
- \_\_\_ c. In the New Interface dialog box, enter the following information:
- In the **Module or library** field, leave **CreditScoreService** selected.
  - Accept the default **Namespace**.
  - Leave the **Folder** field blank.
  - In the **Name** field, type: **CreditScoreService**
- \_\_\_ d. Click **Finish**.
- \_\_\_ e. In the interface editor, click the **Add Request Response Operation** icon.
- \_\_\_ f. Change the name of the operation to: **calculateCreditScore**
- \_\_\_ g. For **Input(s)**, in the **Name** column, type: **request**
- \_\_\_ h. For **Input(s)**, in the **Type** column, select the **CreditCheckRequest** business object.
- \_\_\_ i. For **Output(s)**, in the **Name** column, type: **calculateCreditScoreReturn**
- \_\_\_ j. For **Output(s)**, in the **Type** column, select the **CreditCheckRequest** business object.
- \_\_\_ k. Your completed interface resembles the following image:

The screenshot shows the interface editor with two main sections: Configuration and Operations.

**Configuration:**

|               |                                              |                                                                              |
|---------------|----------------------------------------------|------------------------------------------------------------------------------|
| Name          | CreditScoreService                           | <a href="#">Refactor name</a>                                                |
| Namespace     | http://CreditScoreService/CreditScoreService | <a href="#">Refactor namespace</a>                                           |
| Binding Style | document literal wrapped                     | <a href="#">Change binding style to document literal non-wrapped</a> More... |

**Operations:**

| Operations and their parameters |         | Name                       | Type               |
|---------------------------------|---------|----------------------------|--------------------|
| <b>calculateCreditScore</b>     |         |                            |                    |
|                                 | Inputs  | request                    | CreditCheckRequest |
|                                 | Outputs | calculateCreditScoreReturn | CreditCheckRequest |

- \_\_\_ l. Save your changes.
- \_\_\_ 4. Create the business rules that are used to return the credit score.
- \_\_\_ a. Expand **CreditScoreService** and right-click **Integration Logic**.
- \_\_\_ b. Click **New > Rules > Rule Group**.
- \_\_\_ c. At the **File Name** panel, type **CreditScoreRG** in the **Name** field.
- \_\_\_ d. Click **Next**.
- \_\_\_ e. At the “Select an Interface” panel, select **CreditScoreService** from the **Interface** list.
- \_\_\_ f. Click **Finish**.
- \_\_\_ g. In the rule group editor, locate the **Default Rule Logic** row from the **Scheduled Rule Logic** table.
- \_\_\_ h. Click the **Enter Rule Logic** link and select **New decision table** from the list.
- \_\_\_ i. At the **File Name** panel, type **CreditScore** in the **Name** field and click **Finish**.

- \_\_ j. In the **Initialize** section of the decision table editor, click the **Add an Initialization Action Rule** icon.
- \_\_ k. Change the rule name from **Rule1** to: **SetOutput**
- \_\_ l. In the **Action** column, type `calculateCreditScoreReturn = request` or click the **Action** link to select the items from the list.
- \_\_ m. In the **Table** section of the decision table editor, click the first **Enter Term** link.
- \_\_ n. Type `request.companyName` or select an option from the list.
- \_\_ o. In the first **Enter Value** column, type: "IBM"
- \_\_ p. In the second **Enter Value** column, type: "AbcCo"
- \_\_ q. Click the **Add a New Condition Value** icon and in the third **Enter Value** column, type: "ACME"
- \_\_ r. Right-click the cell that contains "ACME" and click **Add Condition Otherwise** from the menu.
- \_\_ s. Click the second (lower) **Enter Term** link and type `calculateCreditScoreReturn.creditScore` or select the items from the list.
- \_\_ t. Click the **Enter Term** link under "IBM" and type: 11
- \_\_ u. Click the **Enter Term** link under "AbcCo" and type: 1
- \_\_ v. Click the **Enter Term** link under "ACME" and type: 6
- \_\_ w. Click the **Enter Term** link under **Otherwise** and type: 1
- \_\_ x. The completed decision table resembles the following image:

The screenshot shows the IBM BPM Decision Table Editor interface. The top section, labeled 'Initialize', contains three rows: Name (SetOutput), Presentation (empty), and Action (calculateCreditScoreReturn = request). The bottom section, labeled 'Table', has a 'Conditions' header row with columns for 'request.companyName' and four condition values ("IBM", "AbcCo", "ACME", and 'Otherwise'). Below this is a 'Actions' header row with columns for 'calculateCreditScoreReturn.creditScore' and four corresponding numerical values (11, 1, 6, and 1). The 'Actions' row is currently selected.

| Initialize                             |                                      |
|----------------------------------------|--------------------------------------|
| Name                                   | SetOutput                            |
| Presentation                           |                                      |
| Action                                 | calculateCreditScoreReturn = request |
| request.companyName                    | "IBM" "AbcCo" "ACME" Otherwise       |
| calculateCreditScoreReturn.creditScore | 11 1 6 1                             |

- \_\_ y. Save your changes and close the decision table editor.
  - \_\_ z. In the rule group editor tab, verify that you see the **CreditScore** decision table in the **Default Rule Logic** field, save your changes, and close the tab.
5. Add the **CreditScore** rule group to the CreditScoreService assembly diagram.
- \_\_ a. If the assembly diagram is not open already, expand **CreditScoreService** and double-click **Assembly Diagram**.

- \_\_\_ b. Expand **Integration Logic > Rule Groups** and drag **CreditScoreRG** onto the assembly diagram.
- \_\_\_ c. Save your changes and close the assembly diagram.

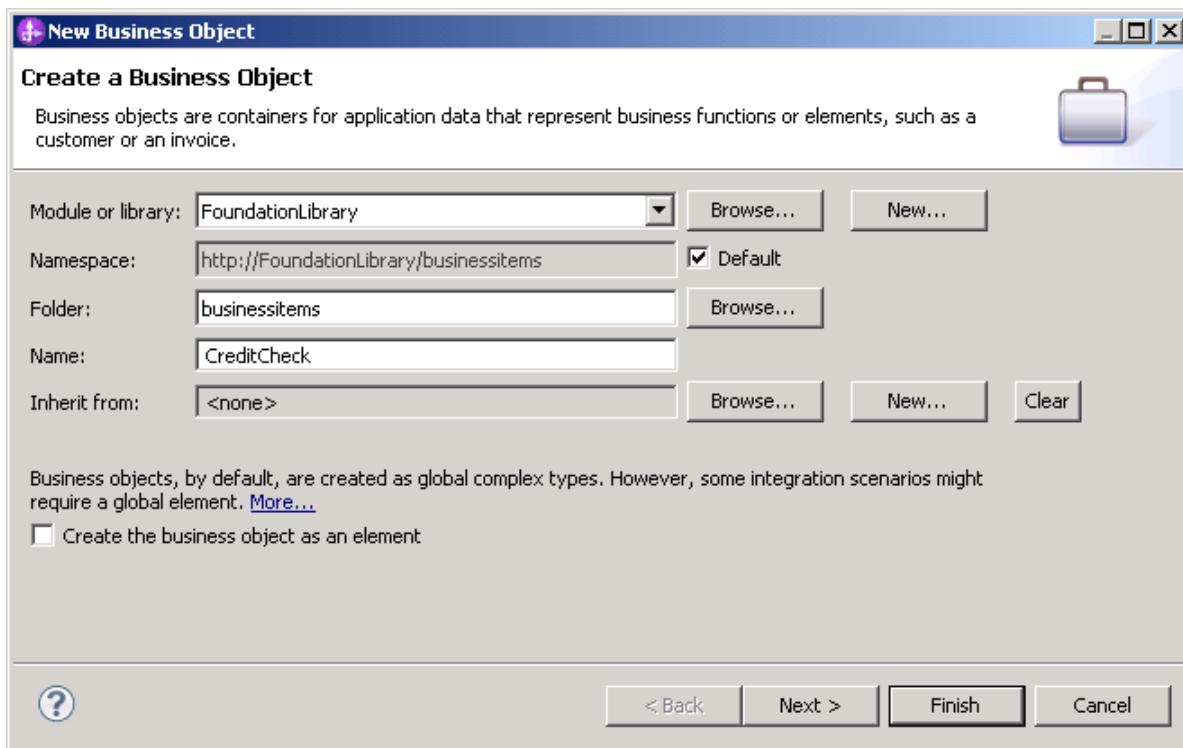
## **Part 2: Creating the artifacts in the FoundationLibrary**

The FoundationLibrary contains a number of shared business object and interface definitions that are used throughout the project. These business objects and interfaces are used in Exercise 5.

The following artifacts are all placed in the FoundationLibrary library. If you do not have one already, you must create a library by this name before proceeding.

To create the library artifacts:

- \_\_\_ 1. Create a CreditCheck business object in the `businessitems` folder.
  - \_\_\_ a. Expand **FoundationLibrary** and right-click **Data**.
  - \_\_\_ b. Click **New > Business Object**.
  - \_\_\_ c. In the New Business Object dialog box, enter the following information:
    - In the **Module or library** field, leave **FoundationLibrary** selected.
    - Accept the default **Namespace**.
    - In the **Folder** field, type `businessitems` (which changes the **Namespace** field correspondingly).
    - In the **Name** field, type: `CreditCheck`
    - Leave the **Inherit from** field empty.



- \_\_\_ d. Click **Finish**.

- \_\_\_ e. In the business object editor, add the following fields and their corresponding data types.

| Field name    | Data type |
|---------------|-----------|
| accountNumber | string    |
| companyName   | string    |
| creditScore   | int       |

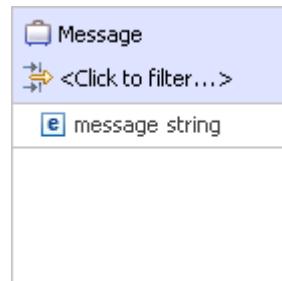
- \_\_\_ f. The completed business object definition resembles the following image:



- \_\_\_ g. Save your changes.

- \_\_\_ 2. Create a Message business object in the `businessitems` folder.

- \_\_\_ a. Repeat the previous steps to add the `Message` business object in the `businessitems` folder.
- \_\_\_ b. The `Message` business object has one field (`message`) of type `string`.
- \_\_\_ c. Save your changes. The business object definition resembles the following image:



- \_\_\_ 3. Create a CustomerApplication business object in the `creditserviceitems` folder.

- \_\_\_ a. Repeat the previous steps to add the `CustomerApplication` business object in the `creditserviceitems` folder.
- \_\_\_ b. The `CustomerApplication` business object has the following fields and corresponding data types:

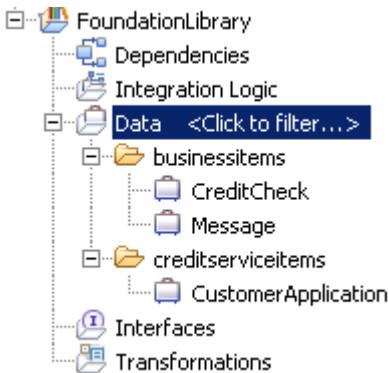
| Field name          | Data type |
|---------------------|-----------|
| accountNumber       | string    |
| applicationDate     | string    |
| applicationDecision | boolean   |
| comments            | string    |
| companyName         | string    |
| contactFirstName    | string    |
| contactLastName     | string    |
| contactPhoneNumber  | string    |
| creditRating        | string    |
| creditReportNeeded  | boolean   |
| creditRisk          | string    |

| Field name           | Data type |
|----------------------|-----------|
| creditScore          | int       |
| customerCity         | string    |
| customerCountry      | string    |
| eligibleApplication  | boolean   |
| ineligibleReason     | string    |
| pricingCode          | string    |
| pricingScore         | string    |
| productName          | string    |
| requestAccountAmount | int       |

- \_\_\_ c. Save your work.
- \_\_\_ d. The completed business object definition resembles the following image:

| CustomerApplication                 |                             |
|-------------------------------------|-----------------------------|
| <Click to filter...>                |                             |
| <input checked="" type="checkbox"/> | accountNumber string        |
| <input checked="" type="checkbox"/> | applicationDate string      |
| <input checked="" type="checkbox"/> | applicationDecision boolean |
| <input checked="" type="checkbox"/> | comments string             |
| <input checked="" type="checkbox"/> | companyName string          |
| <input checked="" type="checkbox"/> | contactFirstName string     |
| <input checked="" type="checkbox"/> | contactLastName string      |
| <input checked="" type="checkbox"/> | contactPhoneNumber string   |
| <input checked="" type="checkbox"/> | creditRating string         |
| <input checked="" type="checkbox"/> | creditReportNeeded boolean  |
| <input checked="" type="checkbox"/> | creditRisk string           |
| <input checked="" type="checkbox"/> | creditScore int             |
| <input checked="" type="checkbox"/> | customerCity string         |
| <input checked="" type="checkbox"/> | customerCountry string      |
| <input checked="" type="checkbox"/> | eligibleApplication boolean |
| <input checked="" type="checkbox"/> | ineligibleReason string     |
| <input checked="" type="checkbox"/> | pricingCode string          |
| <input checked="" type="checkbox"/> | pricingScore string         |
| <input checked="" type="checkbox"/> | productName string          |
| <input checked="" type="checkbox"/> | requestAccountAmount int    |

- \_\_\_ 4. After adding the business objects, the **Data** section of the **FoundationLibrary** resembles the following image:



- \_\_\_ 5. Create the AccountVerification interface in the `accountverification` folder.
- Expand **FoundationLibrary** and right-click **Interfaces**.
  - Click **New > Interface**.
  - In the New Interface dialog box, enter the following information:
    - In the **Module or library** field, leave **FoundationLibrary** selected.
    - Accept the default **Namespace**.
    - In the **Folder** field, type `accountverification` (entering this folder alters the **Namespace** field correspondingly).
    - In the **Name** field, type: `AccountVerification`
  - Click **Finish**.
  - In the interface editor, click the **Add Request Response Operation** icon.
  - Change the name of the operation to: `InputCriterion`
  - Change the name of the input to: `Input`
  - Change the name of the output to: `Output`
  - For **Input(s)**, in the **Type** column, select the **CustomerApplication** business object.
  - For **Output(s)**, in the **Type** column, select the **Message** business object.
  - Save your work.
  - Your completed operation resembles the following image:

| Name    |                | Type                |
|---------|----------------|---------------------|
| ▼       | InputCriterion |                     |
| Inputs  | Input          | CustomerApplication |
| Outputs | Output         | Message             |

- \_\_\_ 6. Create the CreditRiskAssessment interface in the `creditriskassessment` folder.
- Follow the previous steps to create the `CreditRiskAssessment` interface. The interface operation is named `InputCriterion`.
  - Name the inputs: `Input`

- c. Name the outputs: Output
- d. Save your work.
- e. The completed operation resembles the following image:

| Name                                                                                             | Type   |
|--------------------------------------------------------------------------------------------------|--------|
|  InputCriterion |        |
| Inputs                                                                                           | Input  |
| Outputs                                                                                          | Output |

- 7. Use the previous steps to create the following interface definitions. For these interfaces, do not specify a folder. Leave the **Folder** field empty. Remember to save your interfaces.
  - a. Create the `DetermineApplicantEligibility` interface. Name the two-way operation of the interface: `InputCriterion`  
The Input (named `credappin`) and Output (named `credappout`) are `CustomerApplication` types of business objects.

| Name                                                                                             | Type                |
|--------------------------------------------------------------------------------------------------|---------------------|
|  InputCriterion |                     |
| Inputs                                                                                           | CustomerApplication |
| Outputs                                                                                          | CustomerApplication |

- b. Create the `FinalApplicationReview` interface. Name the two-way operation of the interface: `InputCriterion`  
The input (named `Input`) and output (named `Output`) are `CustomerApplication` types of business objects.

| Name                                                                                               | Type                |
|----------------------------------------------------------------------------------------------------|---------------------|
|  InputCriterion |                     |
| Inputs                                                                                             | CustomerApplication |
| Outputs                                                                                            | CustomerApplication |

- c. Create the `GenerateDecline` interface. Name the two-way operation of the interface: `InputCriterion`  
The input (named `Input`) is a `CustomerApplication` type of business object, and the output (named `Output`) is a `Message` type of business object.

| Name                                                                                               | Type                |
|----------------------------------------------------------------------------------------------------|---------------------|
|  InputCriterion |                     |
| Inputs                                                                                             | CustomerApplication |
| Outputs                                                                                            | Message             |

- \_\_\_ d. Create the `RecordDeclinedApplication` interface. Name the one-way operation of the interface: `InputCriterion`  
 The input (named `Input`) is a `CustomerApplication` type of business object.

|                   | Name  | Type                |
|-------------------|-------|---------------------|
| ▼  InputCriterion |       |                     |
| Inputs            | Input | CustomerApplication |

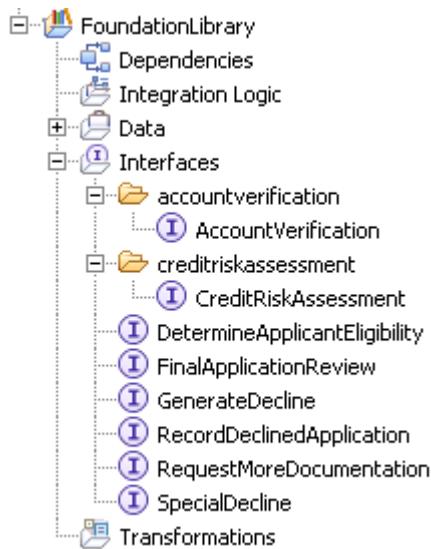
- \_\_\_ e. Create the `RequestMoreDocumentation` interface. Name the two-way operation of the interface: `InputCriterion`  
 The input (named `Input`) and output (named `Output`) are `CustomerApplication` types of business objects.

|                   | Name   | Type                |
|-------------------|--------|---------------------|
| ▼  InputCriterion |        |                     |
| Inputs            | Input  | CustomerApplication |
| Outputs           | Output | CustomerApplication |

- \_\_\_ f. Create the `SpecialDecline` interface. The interface has a two-way operation that is named `InputCriterion`. The interface has an input, named `Input`, of type `CustomerApplication` and an output, named `Output`, of type `Message`.

|                   | Name   | Type                |
|-------------------|--------|---------------------|
| ▼  InputCriterion |        |                     |
| Inputs            | Input  | CustomerApplication |
| Outputs           | Output | Message             |

- \_\_\_ 8. As soon as the interfaces are created, the **Interfaces** section of your `FoundationLibrary` resembles the following image:



# Appendix C. Account verification solution overview

The “account verification” scenario that is used in the exercises is composed of a business process model and solution that are designed to automate the method of validating applications on customer accounts. This solution is based on the proposition that customer accounts currently exist in diverse locations (such as EIS systems) and that they require validation. The process has four basic paths through it:

1. The IBM WebSphere Adapter for Flat Files is used to archive applications that are ineligible.
2. Applications that are eligible and are created for customers that are deemed to be a low credit risk are automatically approved.
3. Applications that are eligible and are created for customers that are deemed to be a high credit risk require supplemental documentation and manual approval or denial.
4. Applications that are eligible and are created for customers that are deemed to be a medium credit risk also require manual approval or denial.

A predefined test case (with corresponding test data) is used to test each path through the process.

- An application for customer AbcCo is used to test the ineligible path (path 1)
- An application for customer IBM is used to test the low credit risk path (path 2)
- An application for customer TestCo is used to test the high credit risk path (path 3)
- An application for customer ACME is used to test the medium credit risk path (path 4)

## ***Part 1: The business process model***

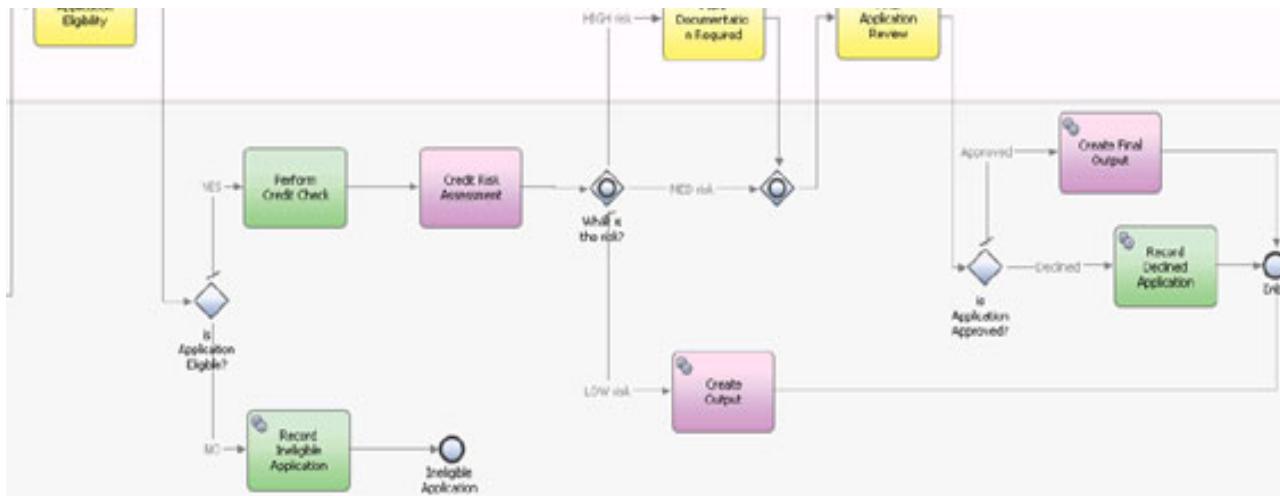
The business process model is introduced in an early exercise. The model is based on a written script, which functions as a sequence of use cases. The script is called a “process narrative.”

### **Process narrative for account verification**

- As soon as the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and terminate the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is done against the customer’s credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.
- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.

- If the customer is determined to be high risk, the customer must submit more documentation, and then the customer must seek final approval for the application.
- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

The narrative is built into a business process diagram (BPD) in IBM Process Designer.



In the exercises, WS-BPEL is used to transform this model into a business process. The model also requires SCA components to complete its implementation and execution.

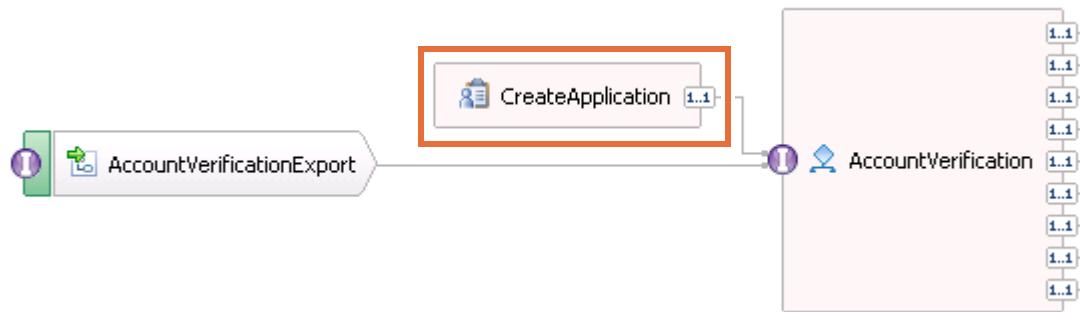
## **Part 2: Paths of the business process solution**

### **Path one: Ineligible applications**

When you test the applications that use the company name AbcCo, the eligibleApplication attribute is set to `false`. When you submit an application that uses `companyName AbcCo`, the application flows through these activities: **Account Verification Receive > Determine Application Eligibility > Map to Ineligible > Record Ineligible Application > Account Verification Reply**. Because the application is ineligible, the flat file adapter is used to archive the application to the file system for later processing or auditing.

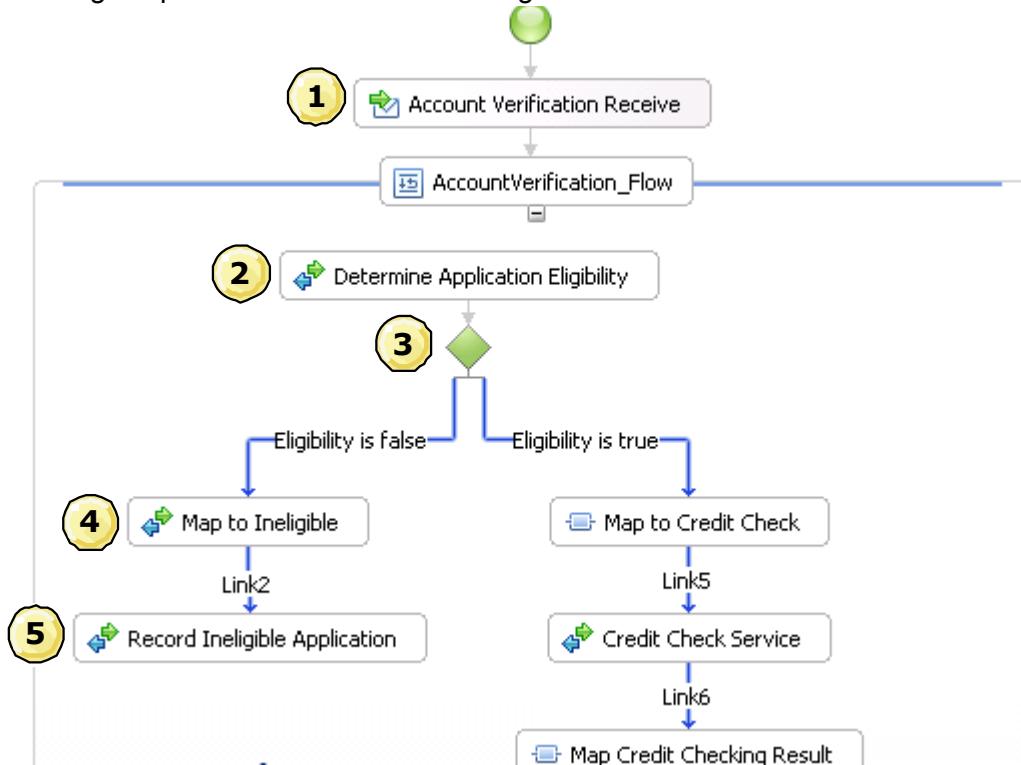
The first path through the process is as follows:

- 1. The AccountOpeningUI, a JSF user interface for the CreateApplication human task, is used to create an application and to trigger a new instance of the AccountVerification business process.



(Before the CreateApplication task is created, you use the IBM Integration Designer integrated test client to test your solution by invoking the AccountVerification SCA component or the AccountVerificationExport component.)

- 2. When the application for AbcCo is received, it takes the ineligible path through the process. The ineligible path consists of the following activities:



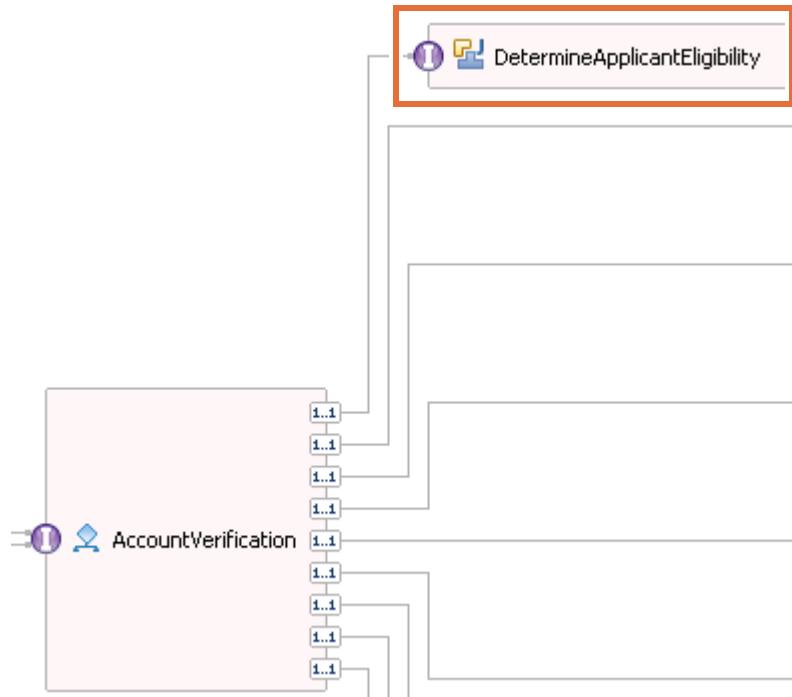
- 1) The Account Verification Receive activity receives the message (the application) sent by the CreateApplication invocation task.

The process follows the link to the next activity, the AccountVerification\_Flow structured activity (a generalized flow).

- 2) The first activity in AccountVerification\_Flow is the Determine Application Eligibility invoke activity. Determine Application Eligibility invokes the

Determine Applicant Eligibility service through the Determine Application Eligibility reference partner.

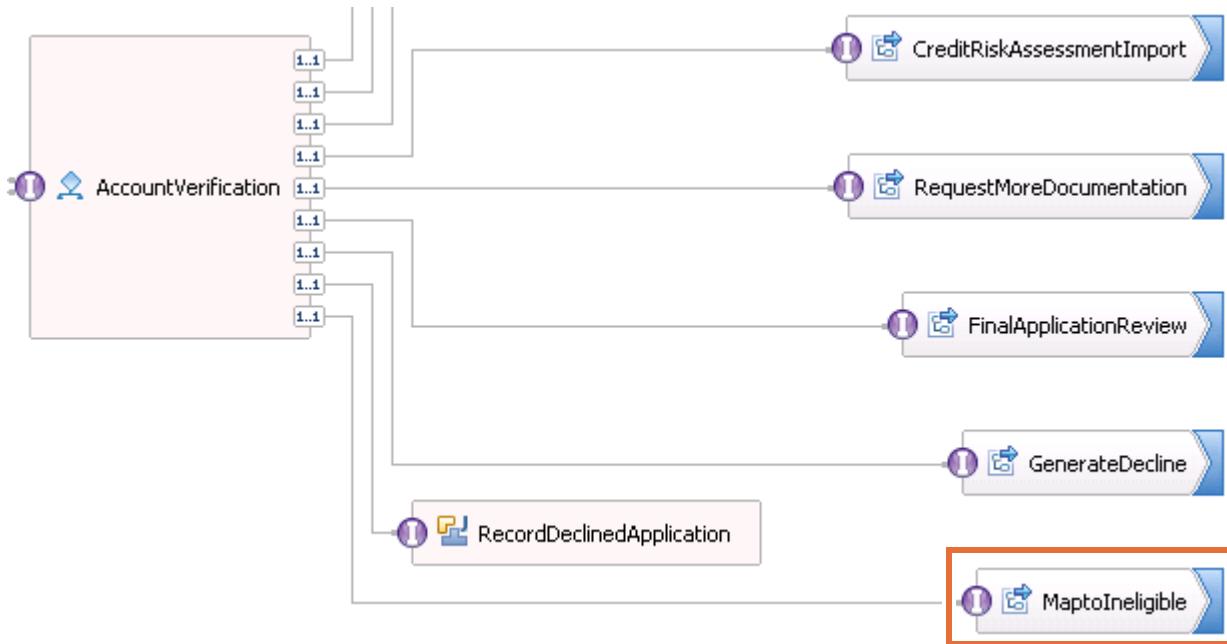
The Determine Applicant Eligibility service (in the FoundationModule) is an SCA component with a Java implementation. It uses simple Java code to return sample data for each of the four test cases.



By default, the Java code sets the `eligibleApplication` element in `CustomerApplicationVariable` to `false` for AbcCo and to `true` for ACME, IBM, and TestCo. In practice, this element would not be set automatically; a human applicant would likely set it.

- 3) The `eligibleApplication` element in `CustomerApplicationVariable` is examined. If `eligibleApplication` is `false` (which is always the case for AbcCo), then the “Eligibility is false” link is followed to the Map to Ineligible activity.
- 4) The Map to Ineligible activity uses the `MaptoIneligiblePartner` reference partner to invoke the services of the `IneligibleMediationService` module.

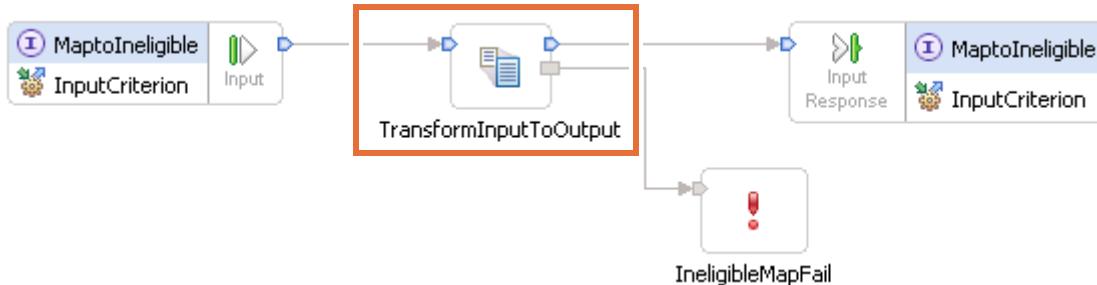
The MaptоНеligiblePartner reference partner invokes the MaptоНеligible import component on the FoundationModule assembly diagram.



The MaptоНеligible import component calls the IneligibleMediationExport component in the IneligibleMediationService module.

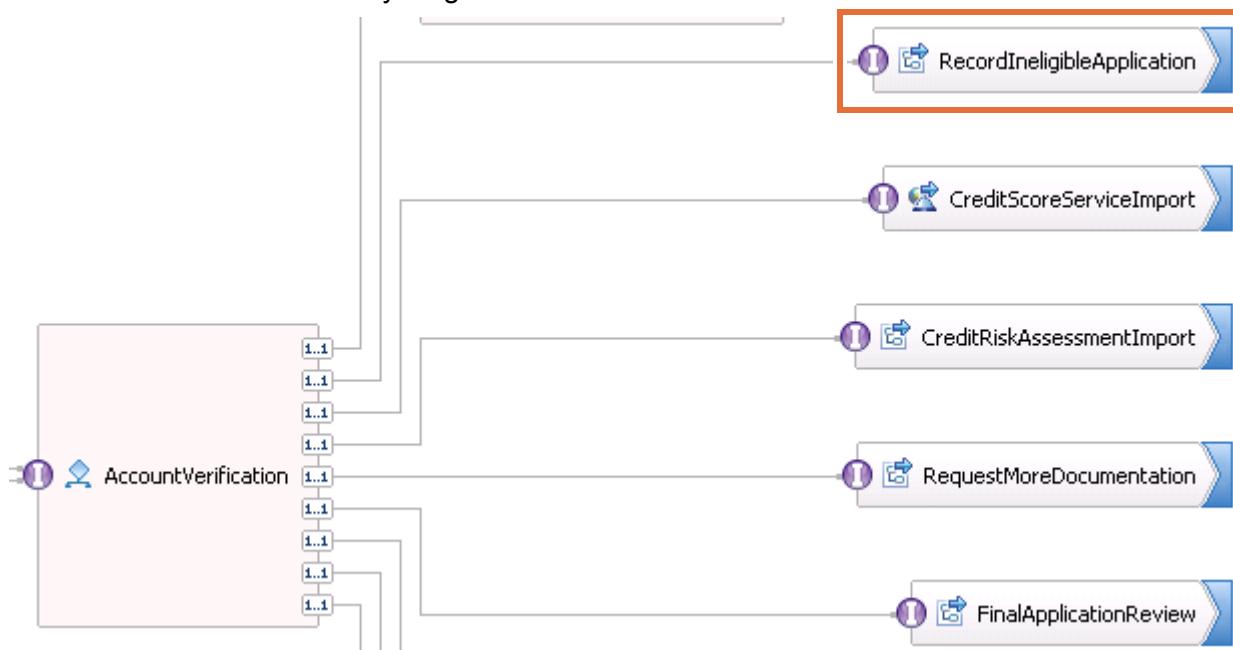


The IneligibleMediation mediation flow (exposed by the IneligibleMediationExport component) transforms the business object from a CustomerApplication into an IneligibleApplication business object. (The service that archives the application uses an IneligibleApplication input, so an XSL transformation primitive is used to transform the data.)

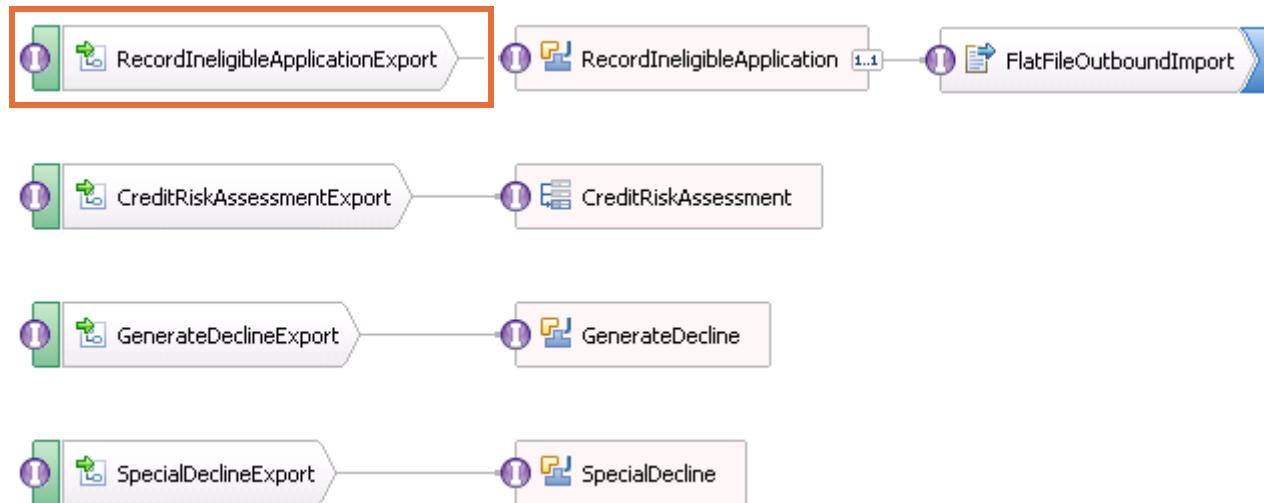


- 5) As soon as the data is transformed, the next link is followed to the Record Ineligible Application invoke activity. The IneligibleApplication business object that the IneligibleMediationService returns is sent to the Record Ineligible Application service through the RecordIneligibleApplication reference partner.

The RecordIneligibleApplication reference partner calls the RecordIneligibleApplication import component on the FoundationModule assembly diagram.



The RecordIneligibleApplication import component invokes the RecordIneligibleApplicationExport component in the FoundationServices module.



In turn, the FlatFileOutboundImport component is invoked. It calls the FlatFile adapter code in the CWYFF\_FlatFile module to write the IneligibleApplication to the file system.

- 6) RecordIneligibleApplication is the final activity in AccountVerification\_Flow. The final link is followed from AccountVerification\_Flow to the Account Verification Reply activity.

Account Verification Reply returns a Message business object to the client that contains a message element, which reads:

Account Verification recorded this application as ineligible for the customer: AbcCo.

The RecordIneligibleApplication Java component on the FoundationServices assembly diagram sets this message text.

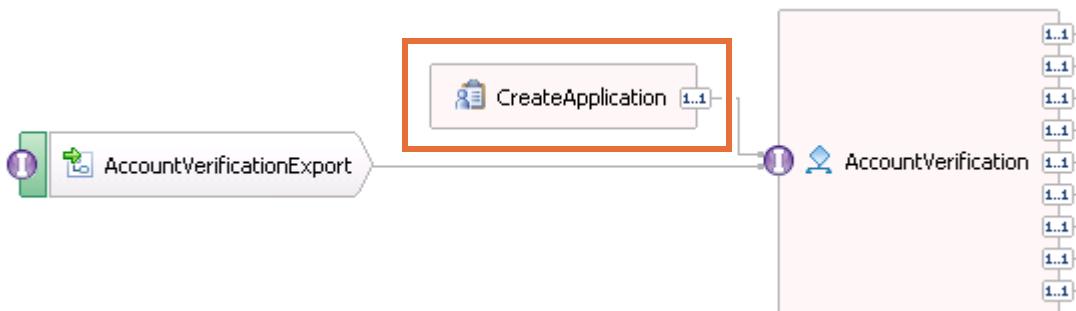


## Path two: Eligible applications with LOW credit risk

When you use the company name IBM to test the solution, the eligibleApplication attribute is set to true. When the Credit Check Service is called, a creditScore of 11 is returned, which represents a `LOW` creditRisk. As soon as you submit an application by using the companyName IBM, the application flows through these activities: **Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment > Create Output > Account Verification Reply**. Because the application is eligible and the creditRisk is `LOW`, the application is automatically approved and a reply is sent from the AccountVerification process.

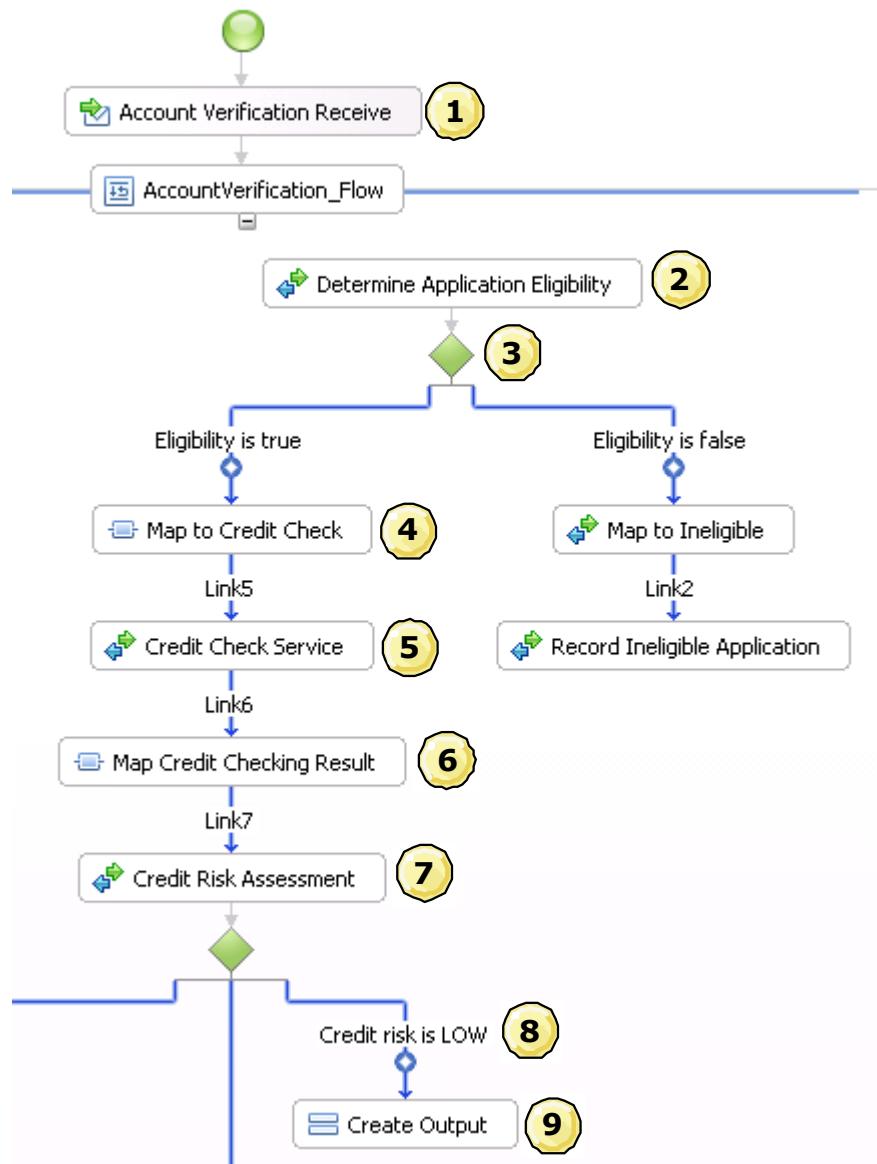
The second path through the process is as follows:

- 1. The AccountOpeningUI, a JSF user interface for the CreateApplication human task, is used to create an application and to trigger a new instance of the AccountVerification business process.



Before the `CreateApplication` task is created, use the IBM Integration Designer integrated test client to test your solution by invoking the `AccountVerification` SCA component or the `AccountVerificationExport` component.

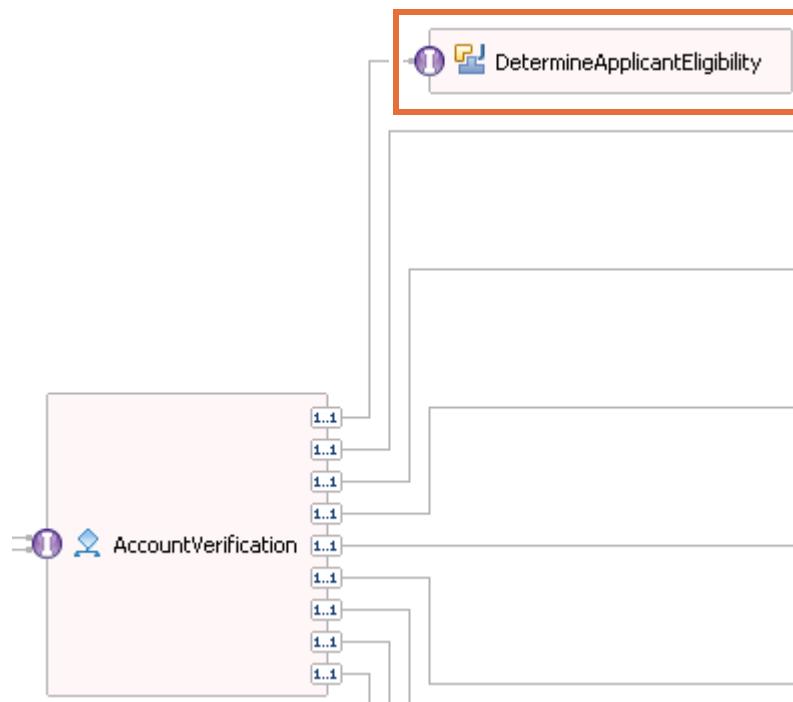
2. When the application for IBM is received, it takes an eligible path through the process. The eligible path for this test case (a low credit risk) consists of the following activities:



- 1) The Account Verification Receive activity receives the message (the application) sent by the CreateApplication invocation task.  
The process follows the link to the next activity, the AccountVerification\_Flow structured activity (a generalized flow).
- 2) The first activity in AccountVerification\_Flow is the Determine Application Eligibility invoke activity. Determine Application Eligibility invokes the

Determine Applicant Eligibility service through the Determine Application Eligibility reference partner.

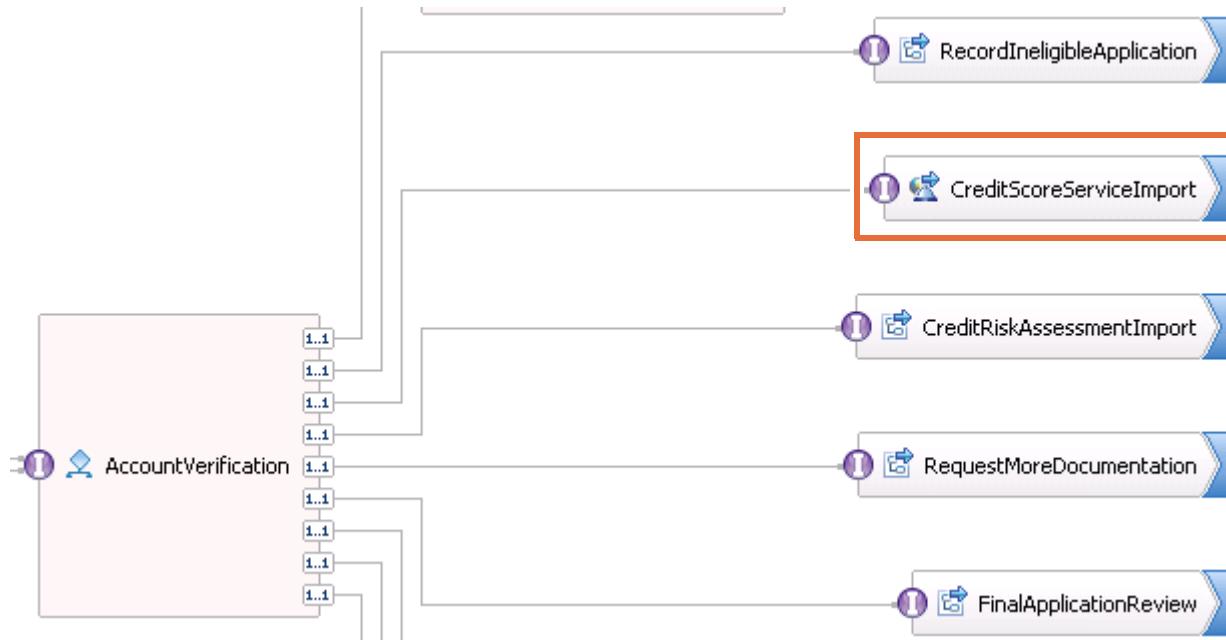
The Determine Applicant Eligibility service (in the FoundationModule) is an SCA component with a Java implementation. It uses simple Java code to return sample data for each of the four test cases.



By default, the Java code sets the eligibleApplication element in the CustomerApplicationVariable to `false` for AbcCo and to `true` for ACME, IBM, and TestCo. In practice, this element would not be set automatically; a human would likely set it.

- 3) The eligibleApplication element in CustomerApplicationVariable is examined. If eligibleApplication is `true` (which is always the case for IBM), then the “Eligibility is true” link is followed to the Map to Credit Check activity.
- 4) The Map to Credit Check activity is a data map activity that transforms the CustomerApplication business object into a CreditCheckRequest business object input. CreditCheckRequest is the input type for the CreditScoreService.
- 5) After the data is transformed, the link to the Credit Check Service invoke activity is followed.

The Credit Check Service activity uses the CreditCheckServicePartner reference partner to invoke the CreditScoreServiceImport component.



The CreditScoreServiceImport component calls the CreditScoreServiceExport component on the CreditScoreService module assembly diagram. The CreditScoreServiceExport is used to expose the services of the CreditScoreRG rule group.



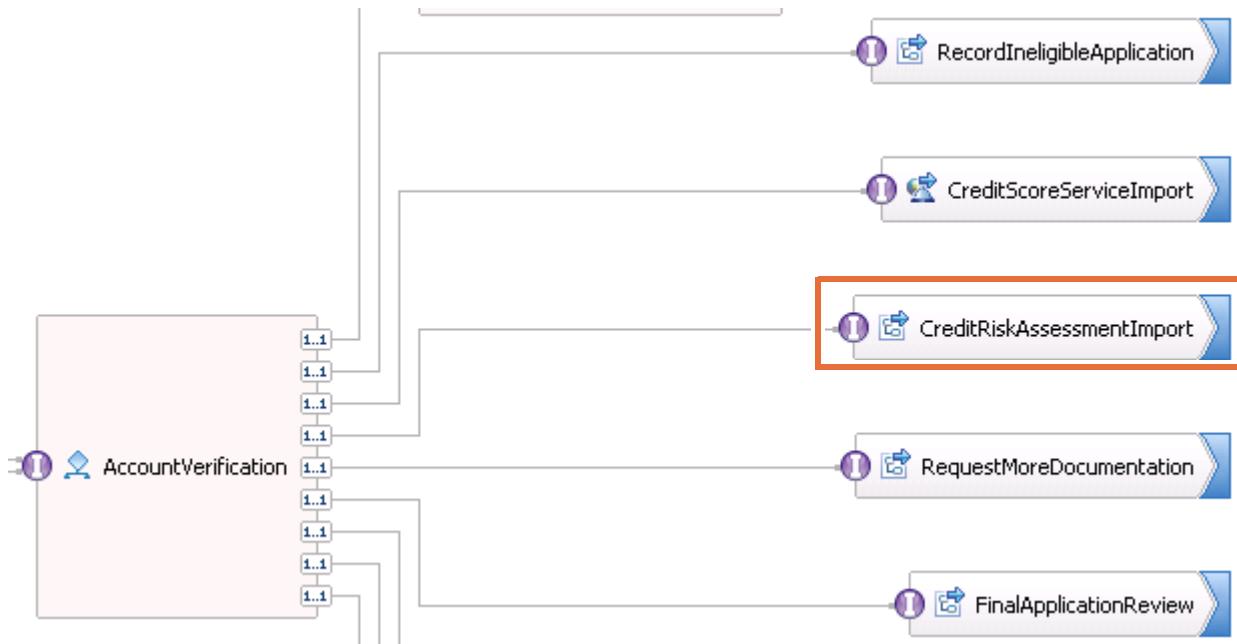
The CreditScoreRG rule group contains a decision table that returns a fixed creditScore based on the companyName element in the CreditCheckRequest input business object. For IBM, the creditScore is always 11, which represents a low creditRisk.

| Conditions                             |       |         |        |           |
|----------------------------------------|-------|---------|--------|-----------|
| request.companyName                    | "IBM" | 'AbcCo' | "ACME" | Otherwise |
| calculateCreditScoreReturn.creditScore | 11    | 1       | 6      | 1         |
| Actions                                |       |         |        |           |

- 6) As soon as the CreditScoreService returns the creditScore, the link to the Map Credit Checking Result data map activity is followed. The CustomerApplication variable is updated with the creditScore by transforming the data from a CreditCheckRequest business object to a CustomerApplication business object.

- 7) After the Map Credit Checking Result data map activity updates the CustomerApplication business object, the link to the Credit Risk Assessment invoke activity is followed.

Credit Risk Assessment uses the CreditRiskAssessmentPartner reference partner to call the CreditRiskAssessmentImport component on the FoundationModule assembly diagram.



The CreditRiskAssessmentImport component invokes the CreditRiskAssessmentExport component on the FoundationServices module assembly diagram. The CreditRiskAssessmentExport component exposes the services of the CreditRiskAssessment rule group.



The CreditRiskAssessment rule group contains a rule set that returns a creditRisk value that is based on the value in the creditScore element of the CustomerApplication. Because the creditScore for IBM is always 11, the creditRisk returned by the rule set is LOW. The rule set updates the contents

of the creditRisk element in the CustomerApplication business object with the value that is returned by the fired rule.

|              |                                                                                                                                                                                    |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskHIGH                                                                                                                                                                           |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/> |
| Name         | RiskMED                                                                                                                                                                            |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="3"/> and less than <input type="text" value="8"/> then the credit risk is <input type="text" value="MED"/>  |
| Name         | RiskLOW                                                                                                                                                                            |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="7"/> and less than <input type="text" value="12"/> then the credit risk is <input type="text" value="LOW"/> |

- 8) As soon as the creditRisk element in the CustomerApplication variable is updated, the creditRisk is evaluated. Because the risk is LOW, the Credit risk is LOW link is followed.
- 9) The final activity in this path is the CreateOutput assign activity. This activity assigns the value for the message element of the Message business object that the process returns. Because the creditRisk is LOW, the output message is: Risk was LOW. Application automatically approved.
- 10) As soon as the message is assigned, the AccountVerification\_flow is complete, and the process follows the link to Account Verification Reply.

Account Verification Reply returns a message business object to the client; it contains the message element that Create Output assigns: Risk was LOW. Application automatically approved.



## Path three: Eligible applications with HIGH credit risk

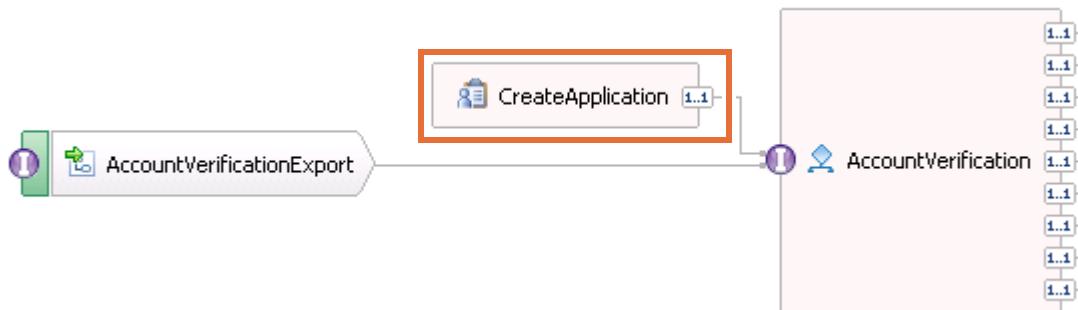
When you use the company name TestCo to test the solution, the eligibleApplication attribute is set to `true` and the creditRisk evaluates to `HIGH` (the creditScore returned is 1). As soon as you submit an application by using **companyName** TestCo, the application flows through these activities: **Account Verification Receive** > **Determine Application Eligibility** > **Map to Credit Check** > **Credit Check Service** > **Map Credit Checking Result** > **Credit Risk Assessment** > **Assign Variable** > **While More Documents Required** > **Request More Documentation**.

A user interface for Request More Documentation is used to change the **comment** field from `None` to `Complete`. After leaving the While More Documents Required loop, the application flows through **Merge Assign** > **Final Application Review**. A user interface for Final Application Review is used to update the **applicationDecision** field. If **applicationDecision** is `true`, the application flows through **Create Output** > **Account Verification Reply**. If **applicationDecision** is `false`, the application

flows through **Generate Decline > Record Declined Application > Create Output > Account Verification Reply**.

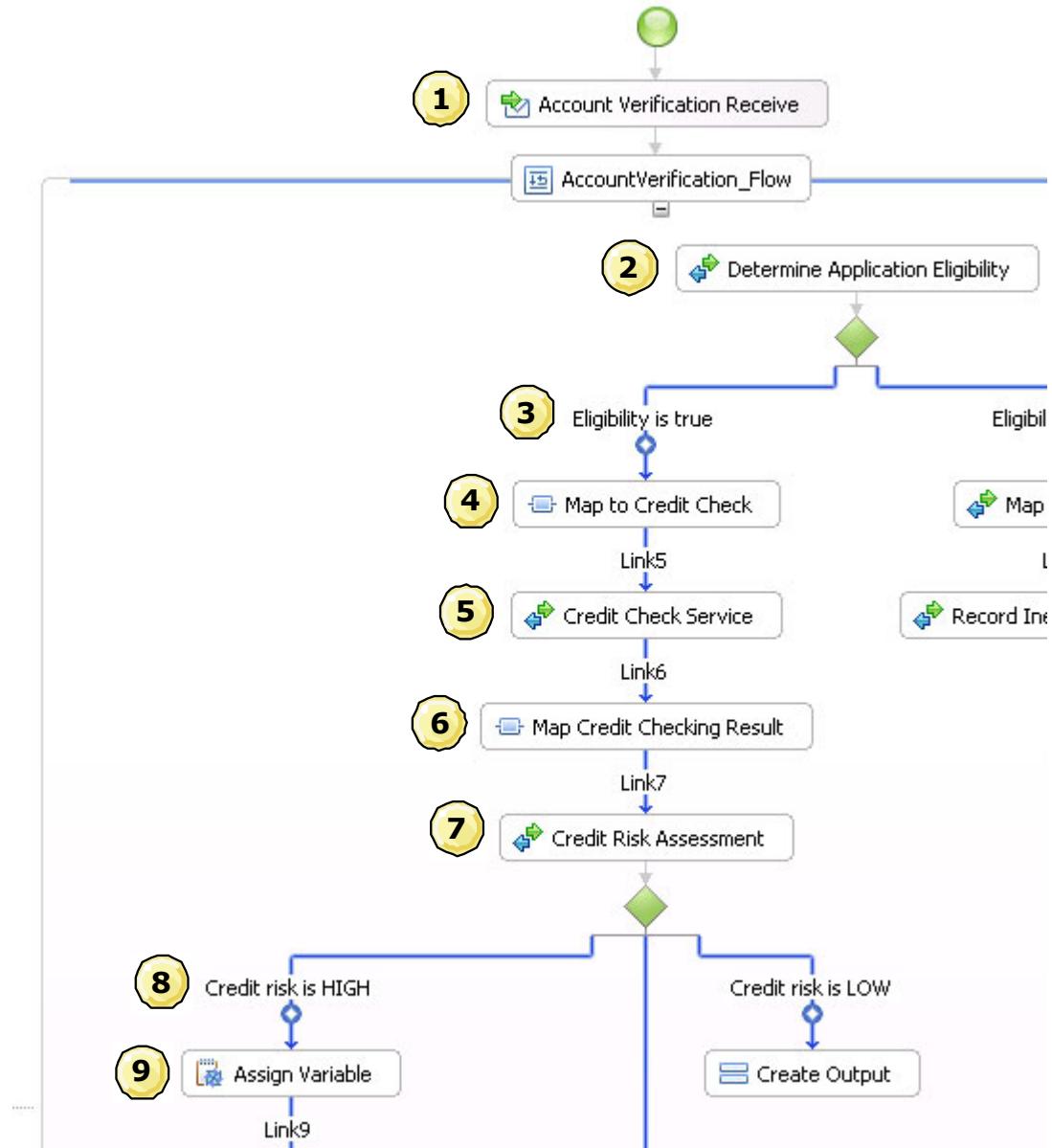
The third path through the process is as follows:

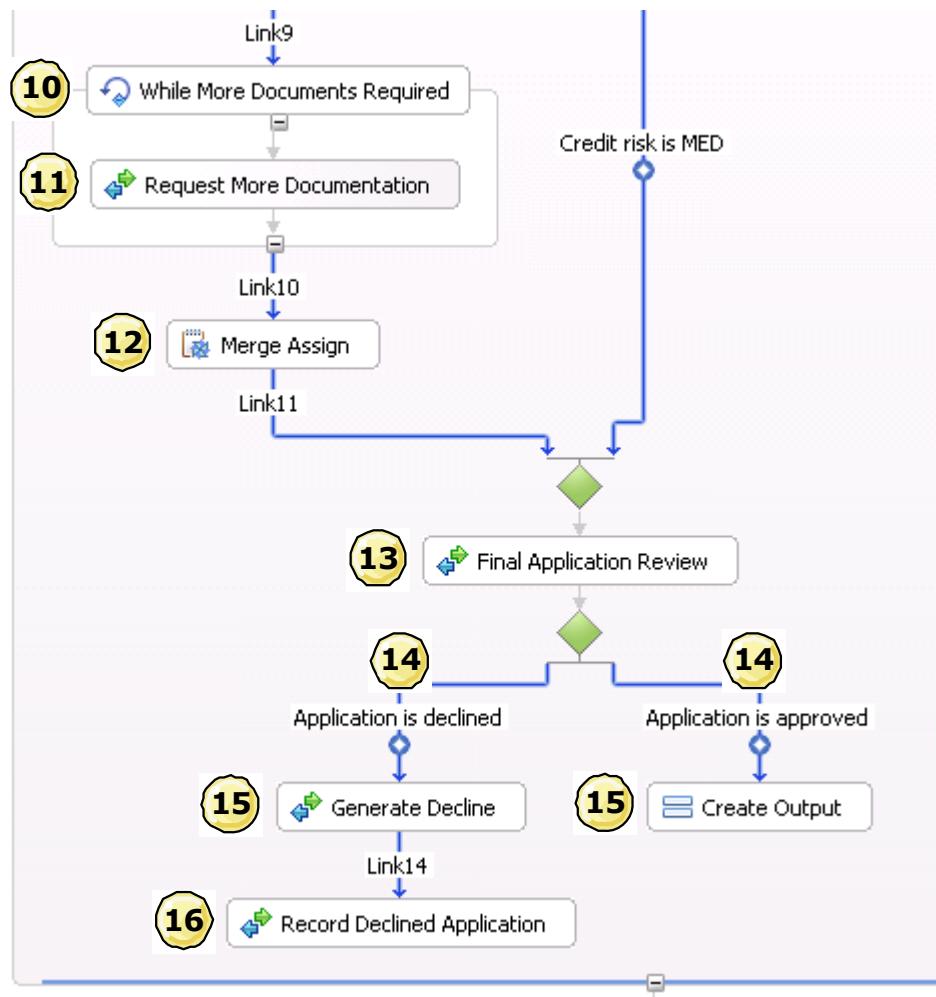
- 1. The AccountOpeningUI, a JSF user interface for the CreateApplication human task, is used to create an application and to trigger a new instance of the AccountVerification business process.



(Before the CreateApplication task is created, you use the IBM Integration Designer integrated test client to test your solution by invoking the AccountVerification SCA component or the AccountVerificationExport component.)

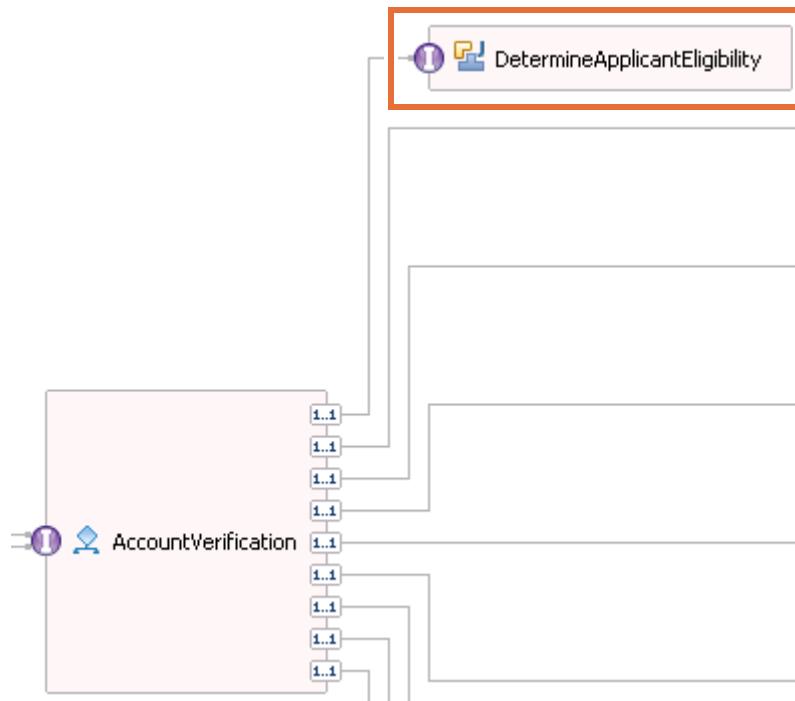
- 2. When the application for TestCo is received, it takes an eligible path through the process. The eligible path for this test case (a high credit risk) consists of the following activities (the diagram is split for readability):





- 1) The Account Verification Receive activity receives the message (the application) sent by the CreateApplication invocation task.  
The process follows the link to the next activity, the AccountVerification\_Flow structured activity (a generalized flow).
- 2) The first activity in AccountVerification\_Flow is the Determine Application Eligibility invoke activity. Determine Application Eligibility invokes the Determine Applicant Eligibility service through the Determine Application Eligibility reference partner.

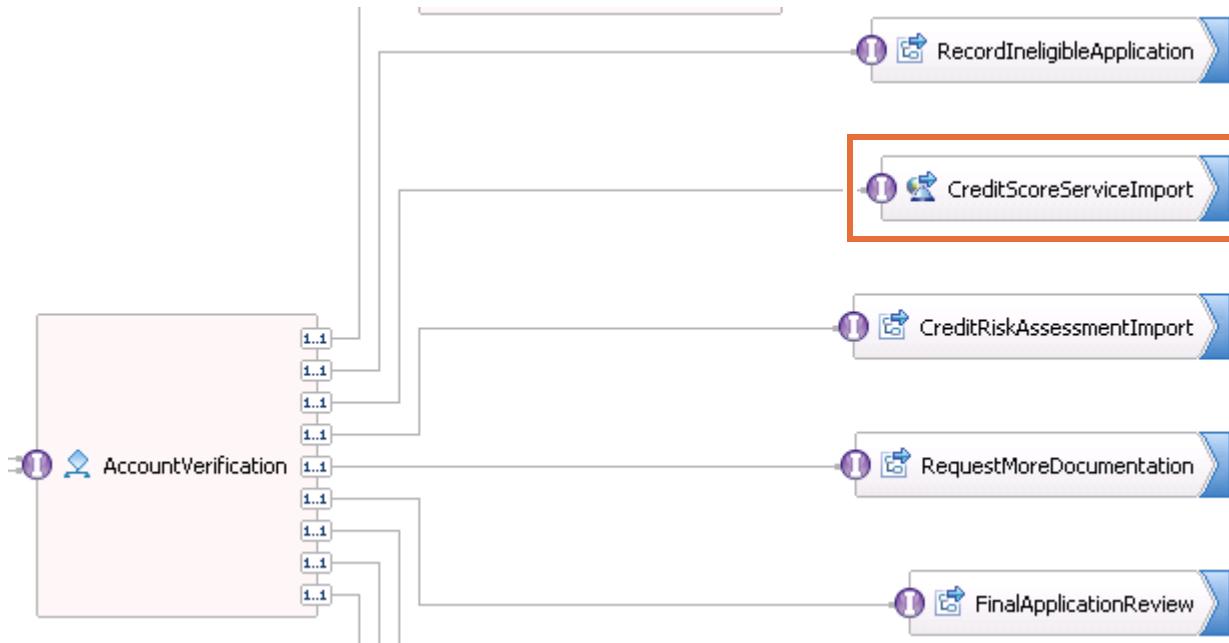
The Determine Applicant Eligibility service (in the FoundationModule) is an SCA component, with a Java implementation, that uses simple Java code to return sample data for each of the four test cases.



By default, the Java code sets the `eligibleApplication` element in `CustomerApplicationVariable` to `false` for `AbcCo` and to `true` for `ACME`, `IBM`, and `TestCo`. In practice, this element would not be set automatically; a human would likely set it.

- 3) The `eligibleApplication` element in `CustomerApplicationVariable` is examined. If `eligibleApplication` is `true` (which is always the case for `TestCo`), then the `Eligibility is true` link is followed to the Map to Credit Check activity.
- 4) The Map to Credit Check activity is a data map activity that transforms the `CustomerApplication` business object into a `CreditCheckRequest` business object input. `CreditCheckRequest` is the input type for the CreditScoreService.
- 5) After the data is transformed, the link to the Credit Check Service invoke activity is followed.

The Credit Check Service activity uses the CreditCheckServicePartner reference partner to invoke the CreditScoreServiceImport component.



The **CreditScoreServiceImport** component calls the **CreditScoreServiceExport** component on the **CreditScoreService** module assembly diagram. The **CreditScoreServiceExport** is used to expose the services of the **CreditScoreRG** rule group.



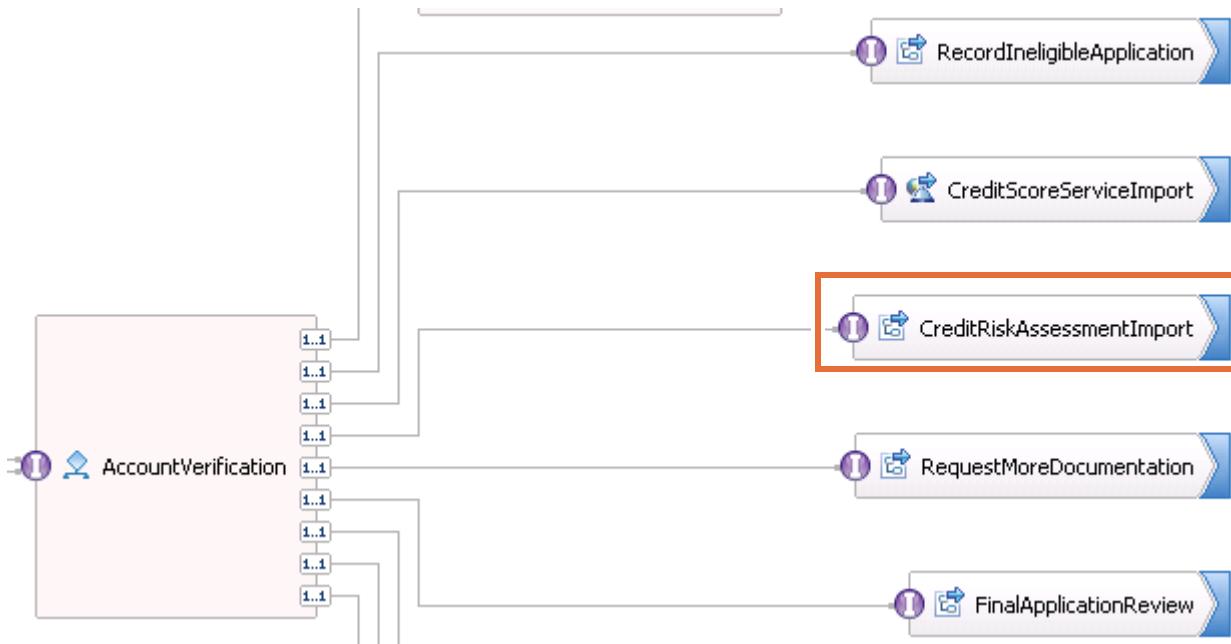
The **CreditScoreRG** rule group contains a decision table that returns a fixed creditScore based on the **companyName** element in the **CreditCheckRequest** input business object. For TestCo, the creditScore is always 1 because the “otherwise” condition is evaluated, which represents a high creditRisk.

| Conditions                             |       |         |        |           | Actions |
|----------------------------------------|-------|---------|--------|-----------|---------|
| request.companyName                    | "IBM" | "AbcCo" | "ACME" | Otherwise |         |
| calculateCreditScoreReturn.creditScore | 11    | 1       | 6      | 1         |         |

- 6) As soon as the CreditScoreService returns the credit score, the link to the Map Credit Checking Result data map activity is followed. The CustomerApplication variable is updated with the creditScore by transforming the data from a CreditCheckRequest business object to a CustomerApplication business object.

- 7) After the Map Credit Checking Result data map activity updates the CustomerApplication business object, the link to the Credit Risk Assessment invoke activity is followed.

Credit Risk Assessment uses the CreditRiskAssessmentPartner reference partner to call the CreditRiskAssessmentImport component on the FoundationModule assembly diagram.



The CreditRiskAssessmentImport component invokes the CreditRiskAssessmentExport component on the FoundationServices module assembly diagram. The CreditRiskAssessmentExport component exposes the services of the CreditRiskAssessment rule group.



The CreditRiskAssessment rule group contains a rule set that returns a creditRisk value that is based on the value in the creditScore element of the CustomerApplication. Because the creditScore for TestCo is always 1, the creditRisk returned by the rule set is HIGH. The rule set updates the contents

of the creditRisk element in the CustomerApplication business object with the value that is returned by the fired rule.

|              |                                                                                                                                                                                    |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskHIGH                                                                                                                                                                           |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/> |

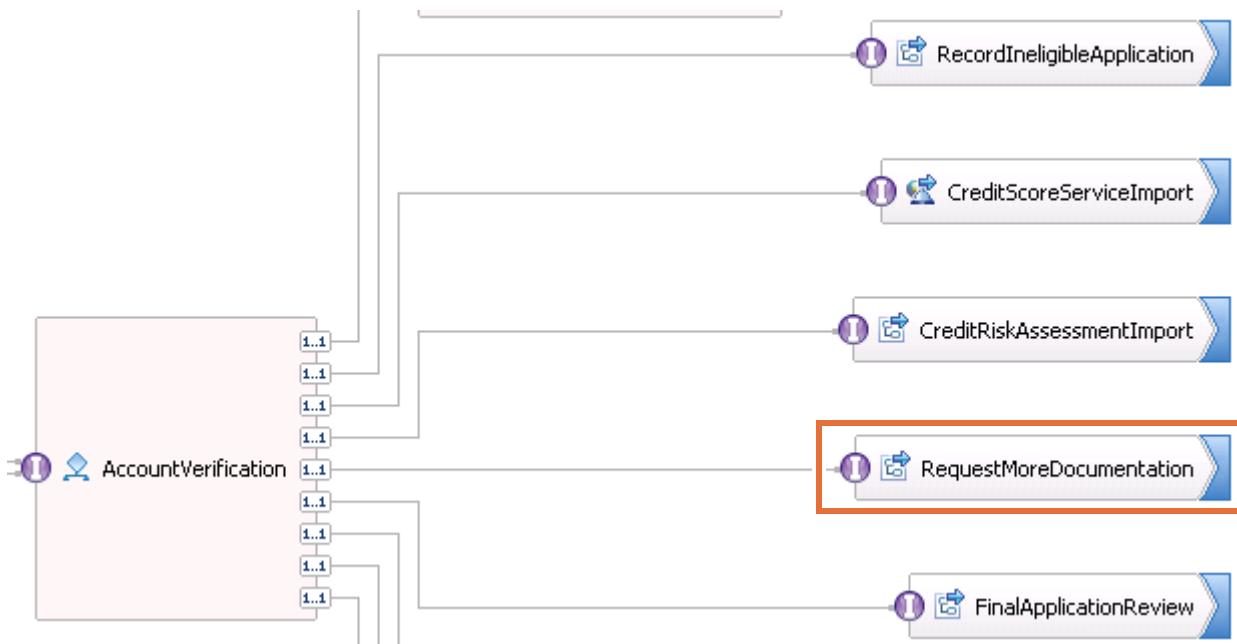
|              |                                                                                                                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskMED                                                                                                                                                                           |
| Template     | CreditRiskTemplate                                                                                                                                                                |
| Presentation | If the customer credit score is greater than <input type="text" value="3"/> and less than <input type="text" value="8"/> then the credit risk is <input type="text" value="MED"/> |

|              |                                                                                                                                                                                    |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskLOW                                                                                                                                                                            |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="7"/> and less than <input type="text" value="12"/> then the credit risk is <input type="text" value="LOW"/> |

- 8) As soon as the creditRisk element in the CustomerApplication variable is updated, the creditRisk is evaluated and because the risk is HIGH, the Credit risk is HIGH link is followed.
- 9) After the Credit risk is HIGH link is followed, the Assign Variable activity is processed. Assign Variable is a Java snippet that generates a new CustomerApplication business object (CustomerApplicationVariable2) and assigns the contents of the original CustomerApplication variable to it.
- 10) After the variable is created and populated, the link to While More Documents Required is followed. While More Documents Required is a while loop that processes the activities in its scope while the comments element of CustomerApplicationVariable2 is set to None.

- 11) The only activity inside the While More Documents Required while loop is the Request More Documentation invoke activity.

Request More Documentation uses the RequestMoreDocumentationPartner reference partner to call the RequestMoreDocumentation import component on the FoundationModule assembly diagram.



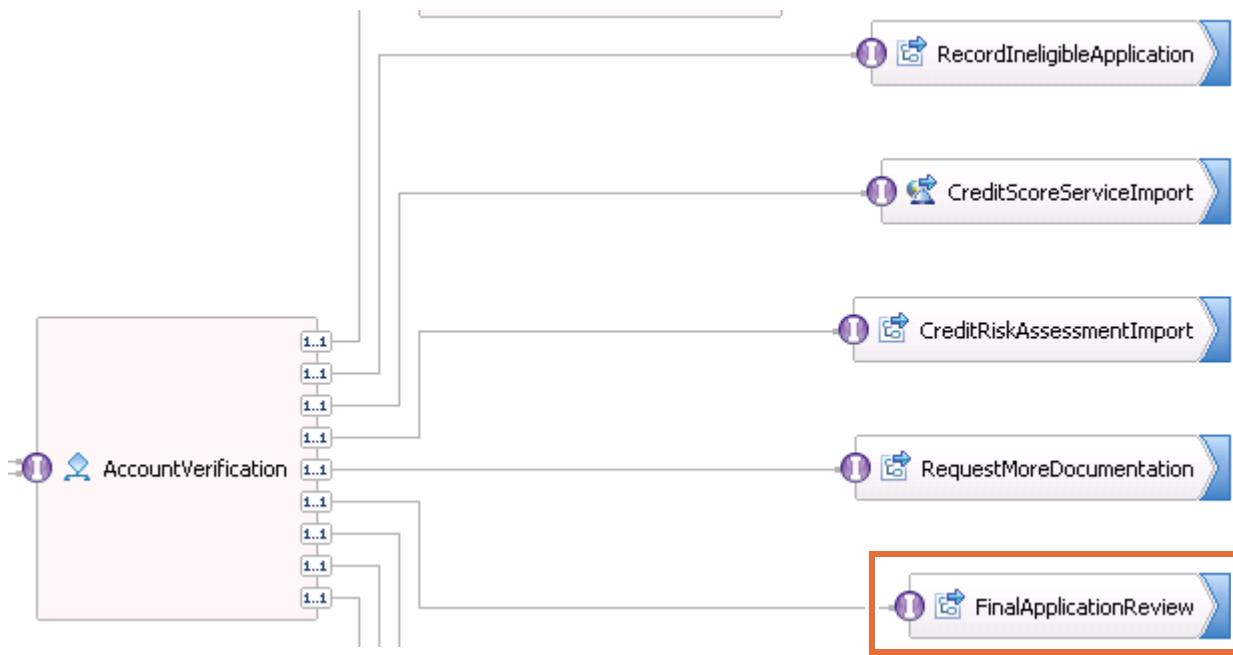
The RequestMoreDocumentation import component invokes the RequestMoreDocumentationExport component on the HumanTaskServices module assembly diagram.



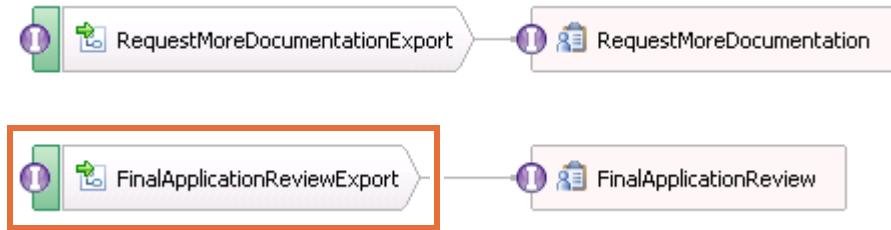
The RequestMoreDocumentationExport component exposes the services of the RequestMoreDocumentation stand-alone human task. Using a human task user interface such as the BPEL Process Choreographer Explorer, an employee updates the comments field from None to Complete. This information update indicates that the supplemental documentation that is required by the high risk application is received.

- 12) As soon as the comments field is updated, the loop is broken, and the link to Merge Assign is followed. Merge Assign is a Java snippet that creates a variable (CustomerApplicationVariable) and assigns the contents of CustomerApplicationVariable2 to it.
- 13) After Merge Assign is complete, the link to Final Application Review is followed.

Final Application Review is an invoke activity that uses the FinalApplicationReviewPartner reference partner to call the FinalApplicationReview import component on the FoundationModule assembly diagram.



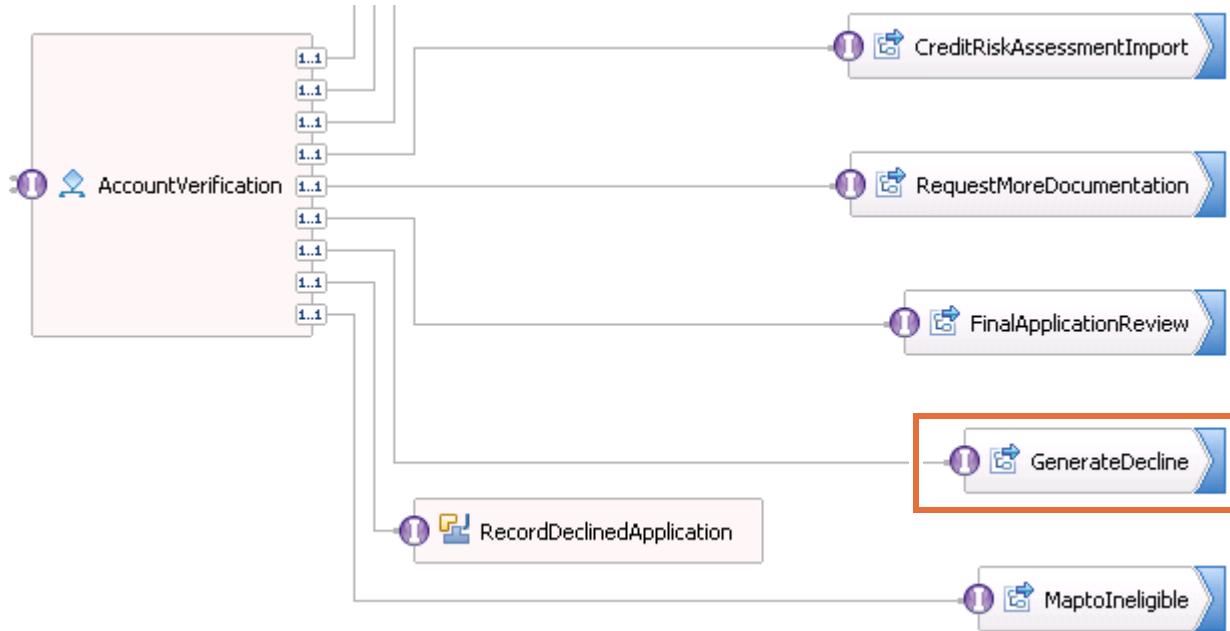
The FinalApplicationReview import invokes the FinalApplicationReviewExport component on the HumanTaskServices assembly diagram.



The FinalApplicationReviewExport component exposes the services of the FinalApplicationReview stand-alone human task. Using a human task user interface such as AccountProcessingUI, an employee does a final review of the high risk application and supplemental documentation to decide whether the application is approved or declined. Using the user interface, the employee updates the `applicationDecision` field to `true` (approved) or `false` (declined).

- 14) Depending on the action that the employee takes, the applicationDecision field is examined and either the Application is declined link is followed or the Application is approved link is followed.
  - 15) If the Application is approved link is followed, the Create Output activity is processed. Create Output is an assign activity that assigns the message element of the Message business object to: Application was approved.

If the Application is declined link is followed, the Generate Decline invoke activity is processed. The Generate Decline activity uses the GenerateDeclinePartner reference partner to call the GenerateDecline import component on the FoundationModule assembly diagram.

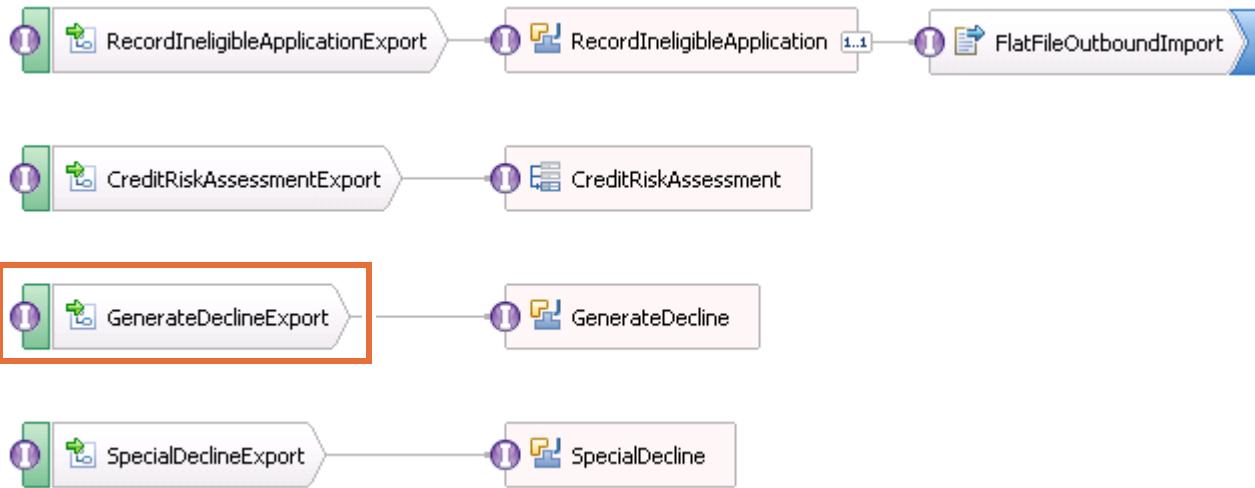


The GenerateDecline import component invokes the RouteRequestExport component on the RouterMediationService assembly diagram.



**RouteRequestExport** exposes the services of the **RouteRequest** mediation flow. If the creditRisk is `HIGH` (which is the case for TestCo) and applicationDecision is `false`, the message is routed to the **GenerateDeclineService** import component. The **GenerateDeclineService**

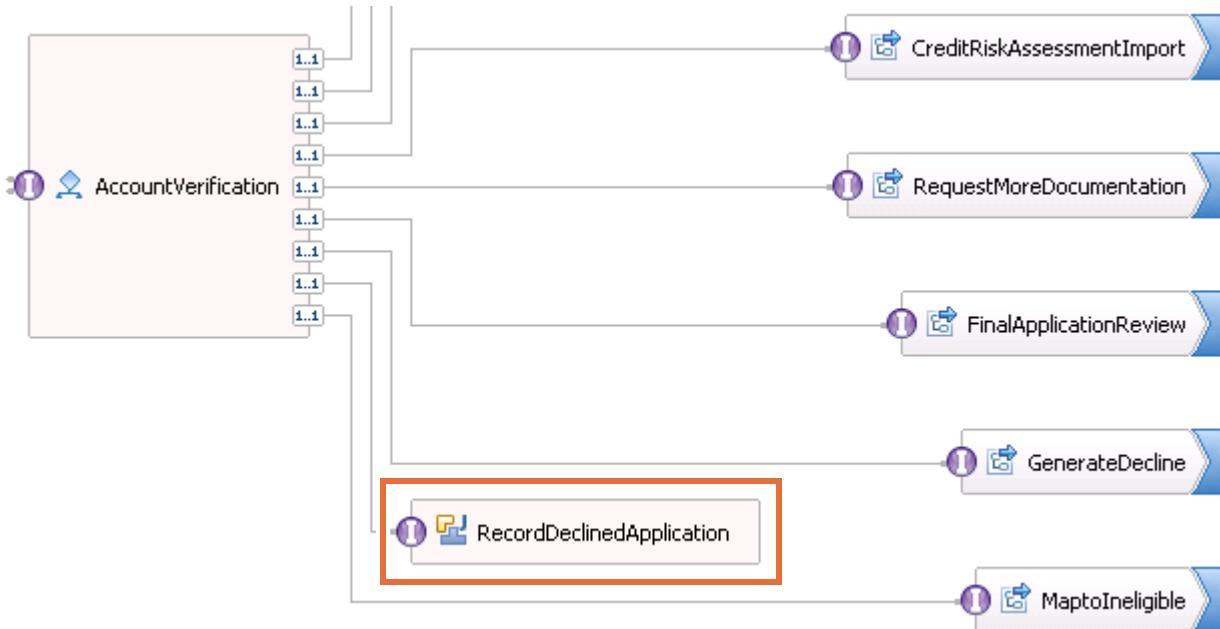
import invokes The GenerateDeclineExport component on the FoundationServices assembly diagram.



This GenerateDeclineExport exposes the services of the GenerateDecline Java component. This component sets the message element of the Message business object to: Account for customer TestCo was declined and the credit risk was HIGH.

- 16) As soon as the GenerateDecline activity is processed, the link to Record Declined Application is followed.

The Record Declined Application invoke activity uses the RecordDeclinedApplicationPartner reference partner to invoke the RecordDeclinedApplication Java component on the FoundationModule assembly diagram.



RecordDeclinedApplication writes messages to the console and to the server log indicating that the application was declined.

- 17) As soon as the message is assigned through the Create Output activity, or the RecordDeclinedApplication is processed, the AccountVerification\_flow is complete, and the process follows the link to Account Verification Reply.

Account Verification Reply returns one of two messages to the client:

- If the application was approved, a Message business object is returned to the client; it contains the message element that Create Output assigns: Application was approved.
- If the application was denied, a Message business object is returned to the client; it contains the message element that the GenerateDecline service assigns: Account for customer TestCo was declined and the credit risk was HIGH.



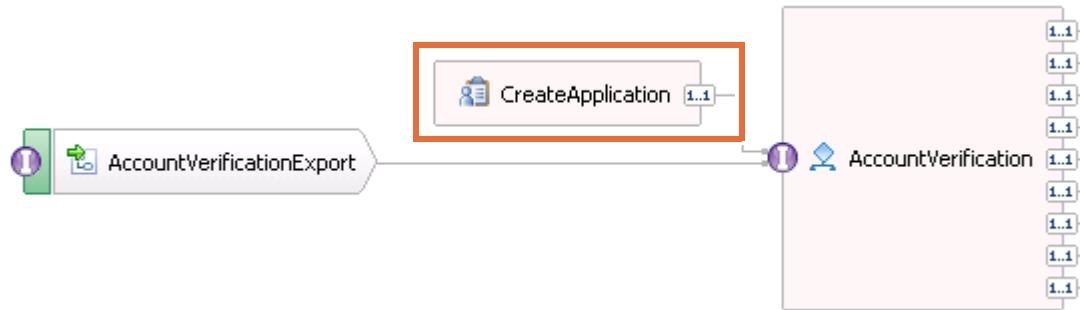
## Path four: Eligible applications with MED credit risk

When you use the company name ACME to test the applications, the eligibleApplication attribute is set to true and the creditRisk evaluates to MED (the creditScore returned is 6). As soon as you submit an application by using companyName ACME, the application flows through these activities: **Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment >Final Application Review > Create Output > Account Verification Reply.**

A user interface for Final Application Review is used to update the **applicationDecision** field. If **applicationDecision** is true, the application flows to **Create Output > Account Verification Reply**. If **applicationDecision** is false, the application flows through **Generate Decline > Record Declined Application > Create Output > Account Verification Reply**.

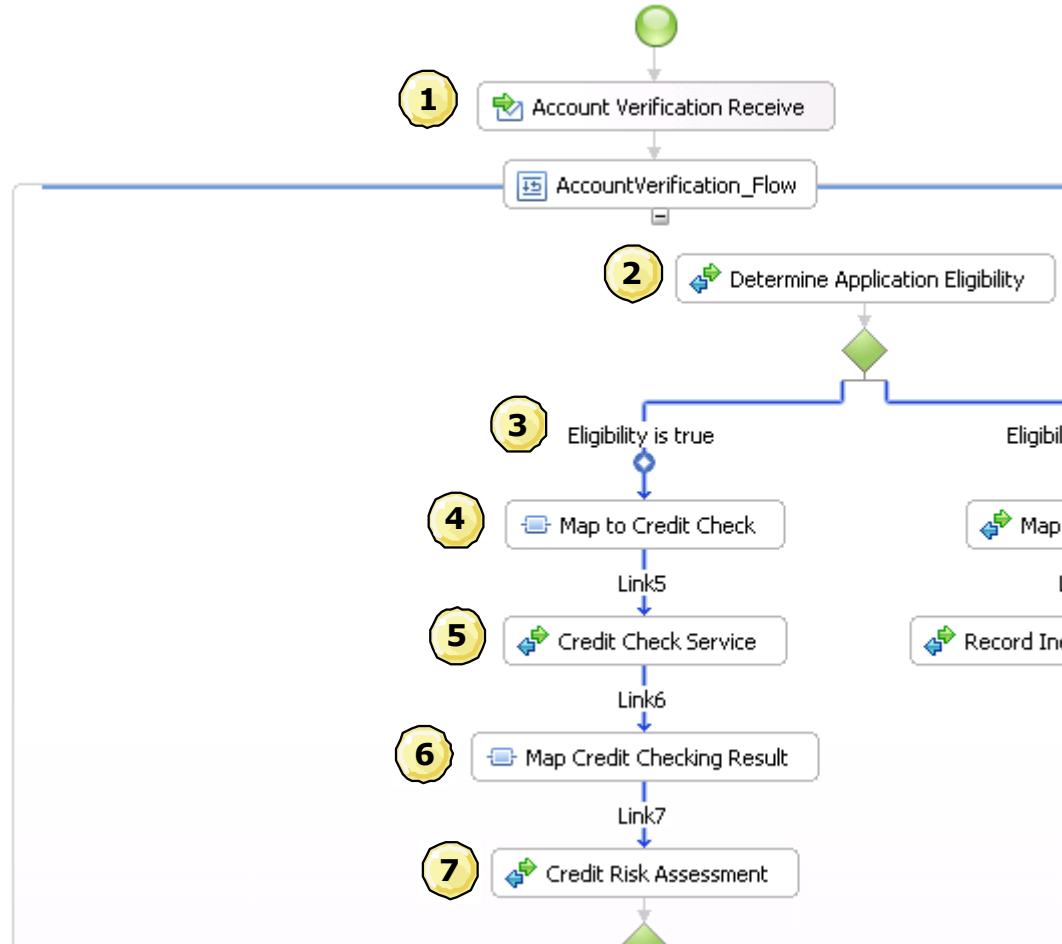
The fourth path through the process is as follows:

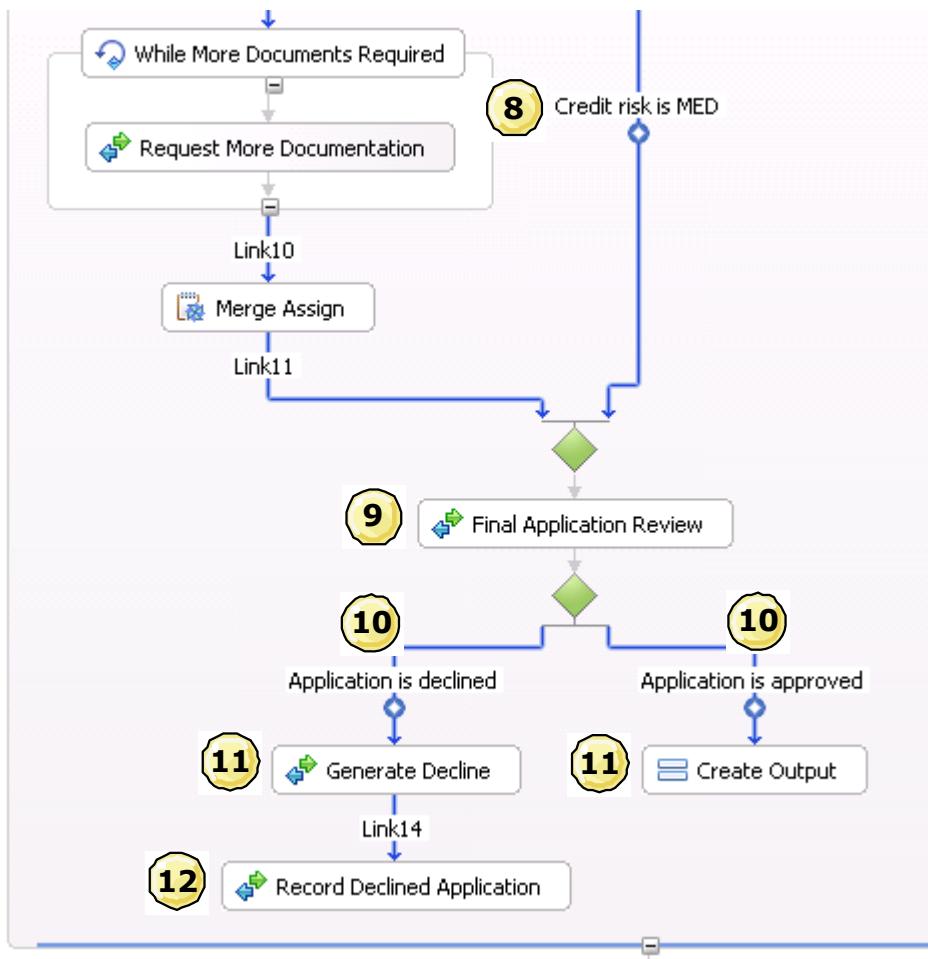
- 1. The AccountOpeningUI, a JSF user interface for the CreateApplication human task, is used to create an application and to trigger a new instance of the AccountVerification business process.



(Before the CreateApplication task is created, you use the IBM Integration Designer integrated test client to test your solution by invoking the AccountVerification SCA component or the AccountVerificationExport component.)

- 2. When the application for ACME is received, it takes an eligible path through the process. The eligible path for this test case (a high credit risk) consists of the following activities (the diagram is split for readability):

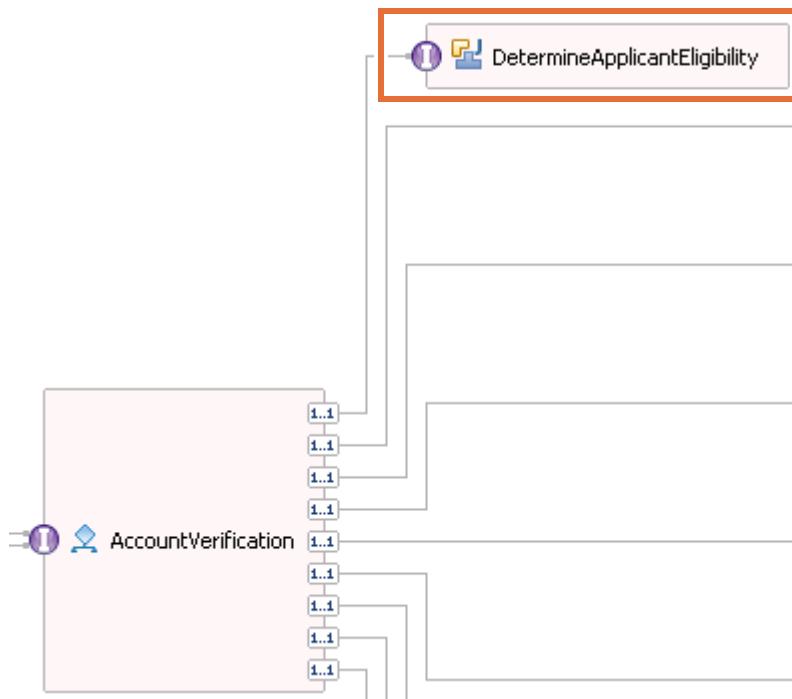




- 1) The Account Verification Receive activity receives the message (the application) sent by the CreateApplication invocation task.  
The process follows the link to the next activity, the AccountVerification\_Flow structured activity (a generalized flow).
- 2) The first activity in AccountVerification\_Flow is the Determine Application Eligibility invoke activity. Determine Application Eligibility invokes the

Determine Applicant Eligibility service through the Determine Application Eligibility reference partner.

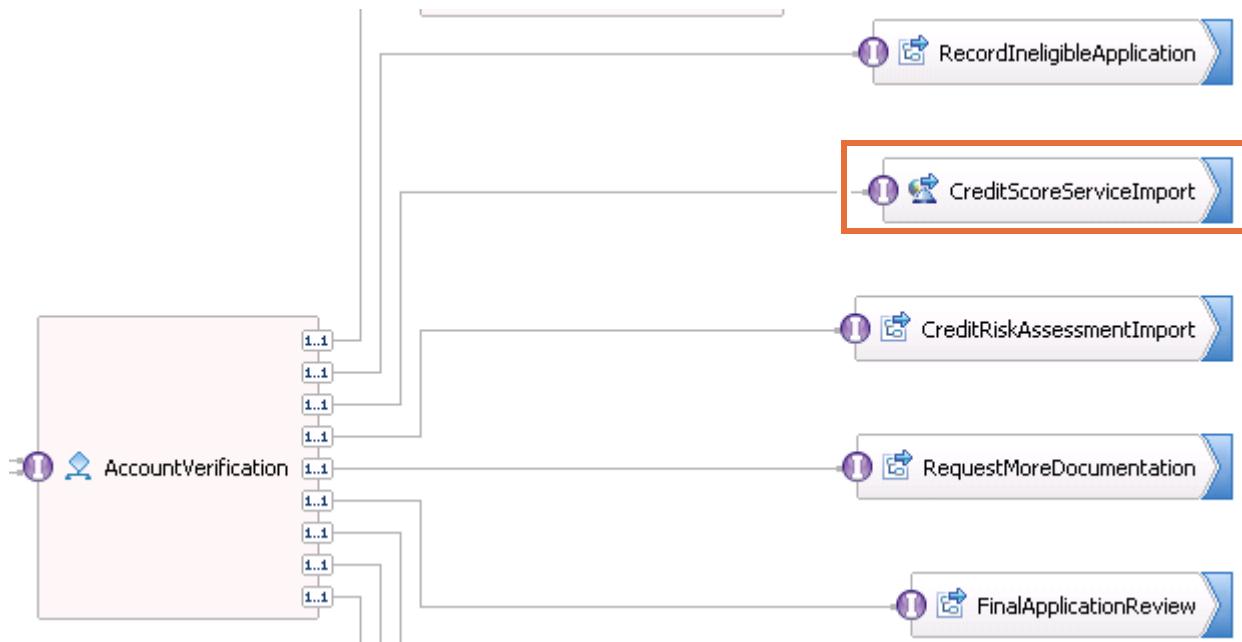
The Determine Applicant Eligibility service (in the FoundationModule) is an SCA component, with a Java implementation, that uses simple Java code to return sample data for each of the four test cases.



By default, the Java code sets the `eligibleApplication` element in the `CustomerApplicationVariable` to `false` for `AbcCo` and to `true` for `ACME`, `IBM`, and `TestCo`. In practice, this element would not be set automatically; a human would likely set it.

- 3) The eligibleApplication element in CustomerApplicationVariable is examined. If eligibleApplication is true (which is always the case for ACME), then the Eligibility is true link is followed to the Map to Credit Check activity.
  - 4) The Map to Credit Check activity is a data map activity that transforms the CustomerApplication business object into a CreditCheckRequest business object input. CreditCheckRequest is the input type for the CreditScoreService.
  - 5) After the data is transformed, the link to the Credit Check Service invoke activity is followed.

The Credit Check Service activity uses the CreditCheckServicePartner reference partner to invoke the CreditScoreServiceImport component.



The CreditScoreServiceImport component calls the CreditScoreServiceExport component on the CreditScoreService module assembly diagram. The CreditScoreServiceExport is used to expose the services of the CreditScoreRG rule group.



The CreditScoreRG rule group contains a decision table that returns a fixed creditScore based on the companyName element in the CreditCheckRequest input business object. For ACME, the creditScore is always 6 (which represents a medium creditRisk).

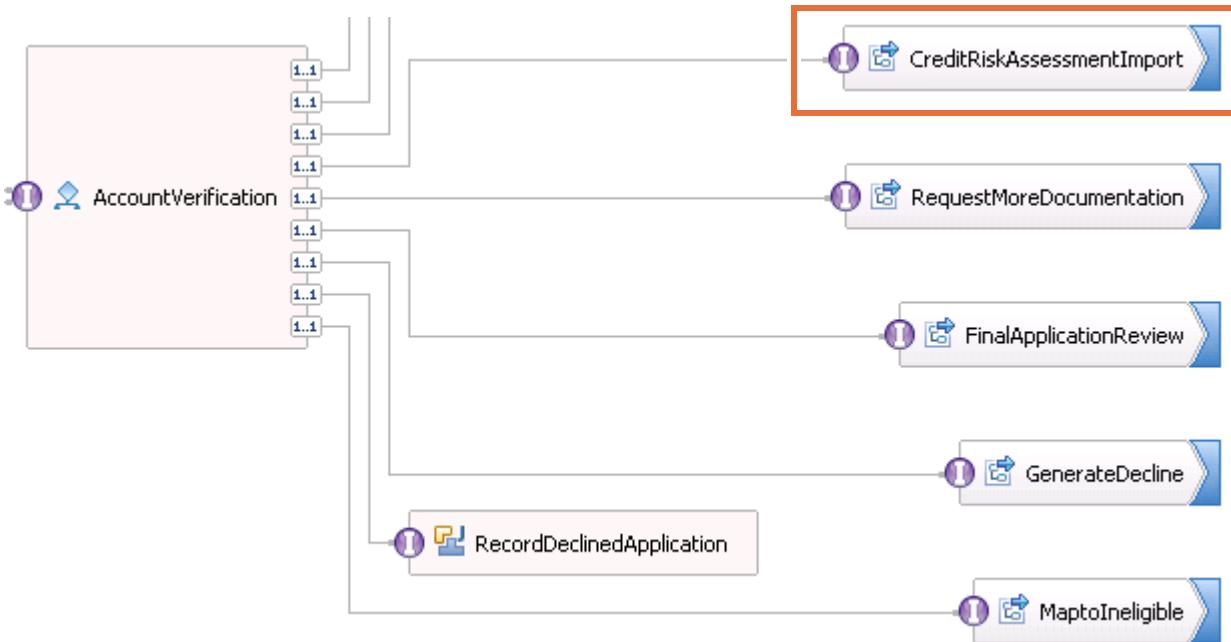
▼Table

| Conditions                             |       |         |           |
|----------------------------------------|-------|---------|-----------|
| request.companyName                    | "IBM" | "AbcCo" | "ACME"    |
| calculateCreditScoreReturn.creditScore | 11    | 1       | 6         |
|                                        |       |         | Otherwise |

Actions

- 6) As soon as the CreditScoreService returns the creditScore, the link to the Map Credit Checking Result data map activity is followed. The CustomerApplication variable is updated with the creditScore by transforming the data from a CreditCheckRequest business object to a CustomerApplication business object.
  - 7) After the Map Credit Checking Result data map activity updates the CustomerApplication business object, the link to the Credit Risk Assessment invoke activity is followed.

Credit Risk Assessment uses the CreditRiskAssessmentPartner reference partner to call the CreditRiskAssessmentImport component on the FoundationModule assembly diagram.



The CreditRiskAssessmentImport component invokes the CreditRiskAssessmentExport component on the FoundationServices module assembly diagram. The CreditRiskAssessmentExport component exposes the services of the CreditRiskAssessment rule group.



The CreditRiskAssessment rule group contains a rule set that returns a creditRisk value that is based on the value in the creditScore element of the CustomerApplication. Because the creditScore for ACME is always 6, the creditRisk returned by the rule set is MED (short for medium). The rule set updates the contents of the creditRisk element in the CustomerApplication business object with the value that is returned by the fired rule.

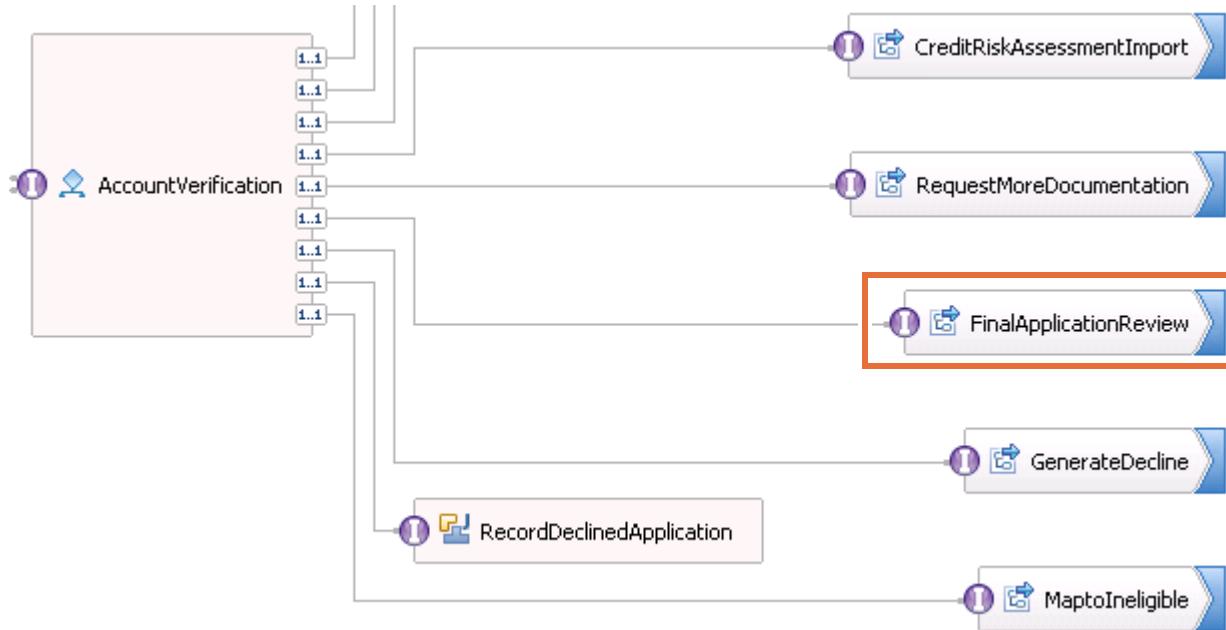
|              |                                                                                                                                                                                    |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskHIGH                                                                                                                                                                           |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/> |

|              |                                                                                                                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskMED                                                                                                                                                                           |
| Template     | CreditRiskTemplate                                                                                                                                                                |
| Presentation | If the customer credit score is greater than <input type="text" value="3"/> and less than <input type="text" value="8"/> then the credit risk is <input type="text" value="MED"/> |

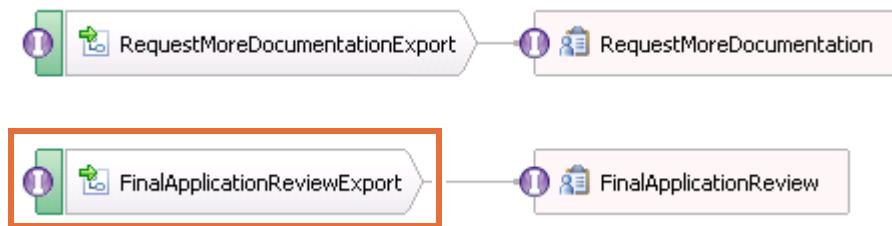
|              |                                                                                                                                                                                    |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name         | RiskLOW                                                                                                                                                                            |
| Template     | CreditRiskTemplate                                                                                                                                                                 |
| Presentation | If the customer credit score is greater than <input type="text" value="7"/> and less than <input type="text" value="12"/> then the credit risk is <input type="text" value="LOW"/> |

- 8) As soon as the creditRisk element in the CustomerApplication variable is updated, the creditRisk is evaluated and because the risk is MED, the Credit risk is MED link is followed.
- 9) After the Credit risk is MED link is followed, the Final Application Review activity is processed.

Final Application Review is an invoke activity that uses the FinalApplicationReviewPartner reference partner to call the FinalApplicationReview import component on the FoundationModule assembly diagram.



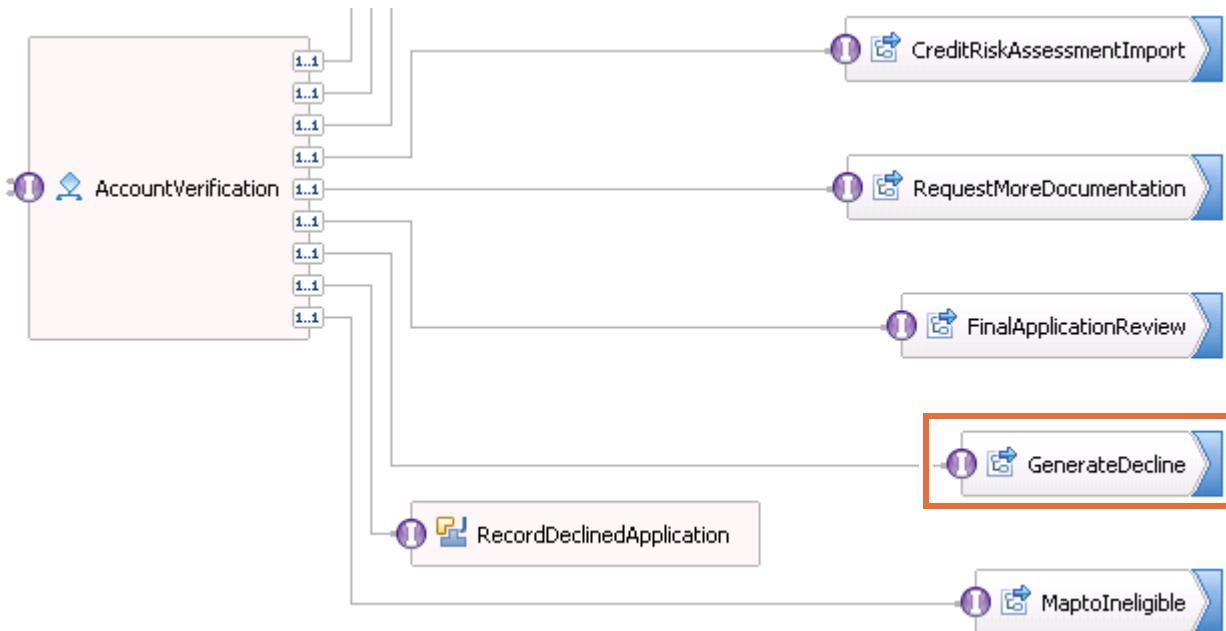
The FinalApplicationReview import invokes the FinalApplicationReviewExport component on the HumanTaskServices assembly diagram.



The FinalApplicationReviewExport component exposes the services of the FinalApplicationReview stand-alone human task. Using a human task user interface such as AccountProcessingUI, an employee does a final review of the medium risk application to decide whether the application is approved or declined. Using the user interface, the employee updates the applicationDecision field to true (approved) or false (declined).

- 10) Depending on the action that the employee takes, the applicationDecision field is examined and either the Application is declined link is followed or the Application is approved link is followed.
  - 11) If the Application is approved link is followed, the Create Output activity is processed. Create Output is an assign activity that assigns the message element of the Message business object to: Application was approved.  
If the Application is declined link is followed, the Generate Decline

If the Application is declined link is followed, the Generate Decline invoke activity is processed. The Generate Decline activity uses the GenerateDeclinePartner reference partner to call the GenerateDecline import component on the FoundationModule assembly diagram.

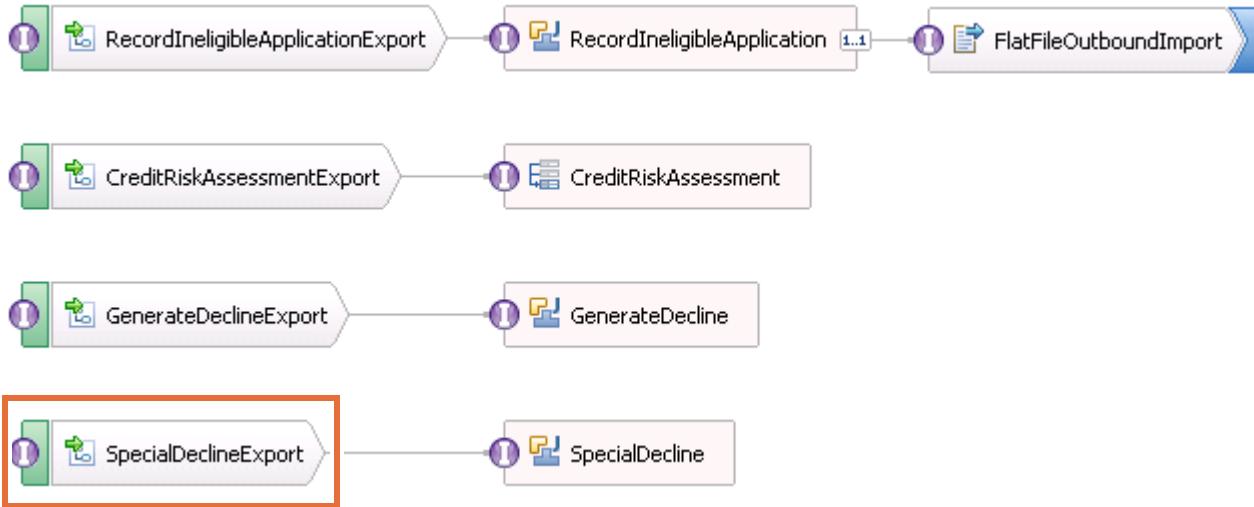


The GenerateDecline import component invokes the RouteRequestExport component on the RouterMediationService assembly diagram.



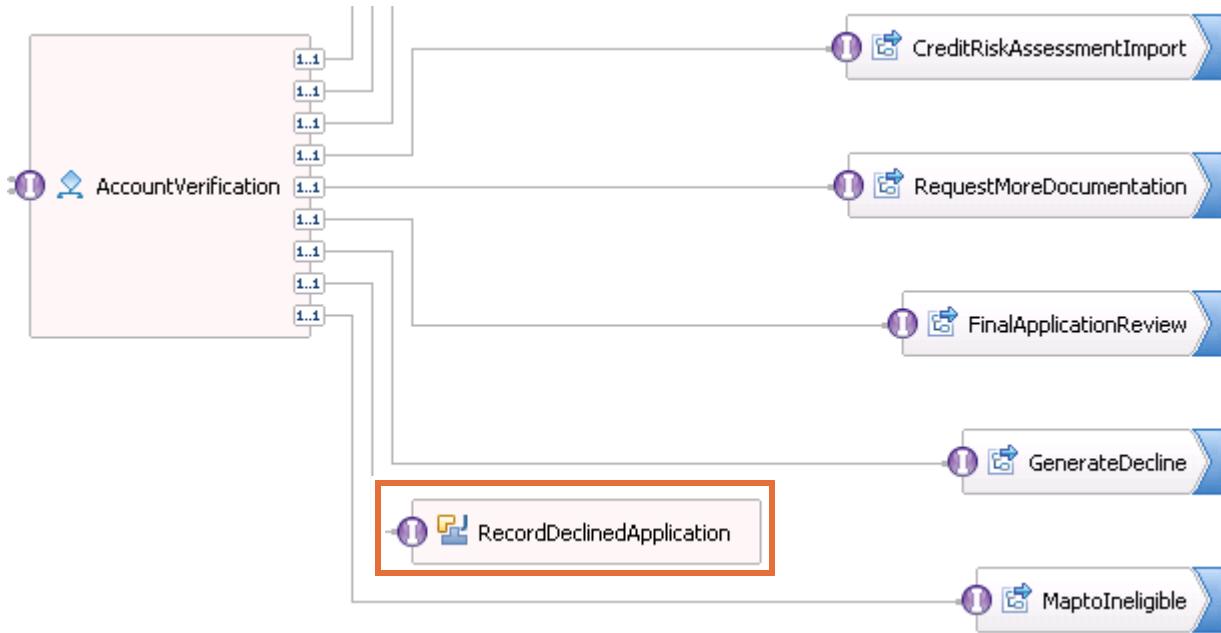
RouteRequestExport exposes the services of the RouteRequest mediation flow. If the creditRisk is `MED` (which is the case for ACME) and applicationDecision is `false`, the message is routed to the SpecialDeclineService import component. The SpecialDeclineService import

invokes the SpecialDeclineExport component on the FoundationServices assembly diagram.



SpecialDeclineExport exposes the services of the SpecialDecline Java component. This component sets the message element of the Message business object to: Account for customer ACME was routed through special decline because the credit risk was MED.

- 12) As soon as the GenerateDecline activity is processed, the link to Record Declined Application is followed. The Record Declined Application invoke activity uses the RecordDeclinedApplicationPartner reference partner to invoke the RecordDeclinedApplication Java component on the FoundationModule assembly diagram.



- 13) As soon as the message is assigned through the Create Output activity, or the RecordDeclinedApplication is processed, the AccountVerification\_flow is complete, and the process follows the link to Account Verification Reply.

Account Verification Reply returns one of two messages to the client:

- If the application was approved, a Message business object is returned to the client; it contains the message element that Create Output assigns: Application was approved.
- If the application was denied, a Message business object is returned to the client; it contains the message element that the SpecialDecline service assigns: Account for customer ACME was routed through special decline because the credit risk was MED.







**IBM**  
®