

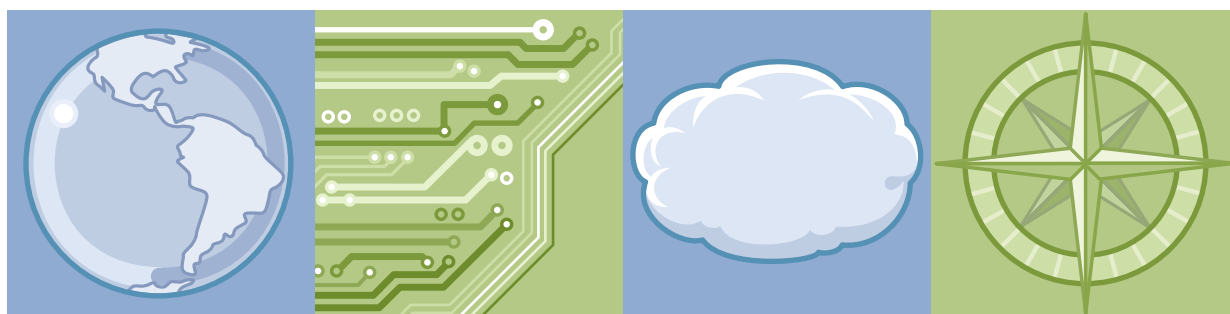


IBM Training

Student Exercises

Developing Solutions with IBM Decision Server Insights V8.7

Course code WB394 ERC 2.0



WebSphere Education

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:

CICS®	developerWorks®	Express®
IBM SmartCloud®	ILOG®	Insight™
Insights™	PartnerWorld®	Redbooks®
SPSS®	WebSphere®	z/OS®

Intel, Intel Xeon and Xeon 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.

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.

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

July 2015 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.

© Copyright International Business Machines Corporation 2015.

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

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

Contents

Trademarks	vii
Exercises description	ix
Exercise 1. Getting started with Decision Server Insights	1-1
Section 1. Installing Decision Server Insights V8.7	1-2
1.1. Exploring the installation steps	1-2
Section 2. Exploring your Decision Server Insights installation	1-8
Section 3. Setting up the Decision Insight perspective	1-10
3.1. Opening Insight Designer	1-10
3.2. Switching to the Decision Insight perspective	1-10
3.3. Setting the target platform	1-11
Section 4. Starting the server from Insight Designer	1-13
Section 5. Setting the debug port	1-14
Exercise 2. Creating a solution in Insight Designer	2-1
Section 1. Creating a solution project	2-2
Exercise 3. Defining the business model	3-1
Section 1. Modeling the domain	3-2
Section 2. Creating the business model definition	3-6
Exercise 4. Creating a rule agent	4-1
Section 1. Setting up your workspace	4-2
1.1. Opening the workspace	4-2
Section 2. Creating a rule agent	4-3
2.1. Creating the product recommendation rule agent	4-4
2.2. Writing the agent descriptor	4-4
Section 3. Writing the San Francisco rule	4-7
Section 4. Creating a Java agent	4-9
Exercise 5. Writing and testing rules	5-1
Section 1. Setting up your workspace	5-2
1.1. Opening the workspace	5-2
1.2. Setting the target platform	5-3
1.3. Importing the projects	5-5
1.4. Verifying properties	5-5
1.5. Verifying the Java build path	5-5
Section 2. Adding the New York recommendation rule	5-7
2.1. Creating the New York rule	5-7
Section 3. Deploying the solution	5-9
3.1. Starting the server	5-9
3.2. Exporting the solution archive	5-10
3.3. Deploying automatically with Ant tasks	5-12
3.4. Verifying deployment	5-13
3.5. Preparing the recording of events in Insight Inspector	5-13
Section 4. Testing the solution	5-14

4.1. Visualizing the event processing in Insight Inspector	5-20
Exercise 6. Using global aggregates in rules	6-1
Section 1. Setting up your workspace	6-2
Section 2. Creating the global aggregates	6-3
2.1. Creating the Average churn for PLATINUM and GOLD aggregate	6-3
2.2. Creating the Average monthly profitability for PLATINUM and GOLD aggregate	6-4
Section 3. Creating a churn prevention rule agent	6-6
3.1. Creating the rule agent	6-6
3.2. Creating a rule that offers a client reward	6-7
Section 4. Deploying the solution	6-9
4.1. Exporting the solution	6-9
4.2. Deploying automatically with Ant tasks	6-10
4.3. Accessing your aggregates through REST	6-10
Section 5. Testing the solution	6-12
Exercise 7. Using event aggregates in rules	7-1
Section 1. Setting up your workspace	7-2
Section 2. Creating the fraud detection rule agent	7-3
2.1. Creating the rule agent	7-3
2.2. Creating the rule: Check amount versus historical average	7-3
Section 3. Deploying the solution	7-5
3.1. Exporting the solution	7-5
3.2. Deploying automatically with Ant tasks	7-6
Section 4. Testing the solution	7-7
Exercise 8. Using time-based and location-based reasoning in rules	8-1
Section 1. Creating a rule to check time and location compatibility	8-2
1.1. Creating the rule: Check distance to recent events	8-2
Section 2. Deploying the solution	8-4
2.1. Exporting the solution	8-4
2.2. Deploying automatically with Ant tasks	8-5
2.3. Deploying the solution to the map tool	8-5
Section 3. Testing the solution	8-7
Exercise 9. Testing for the absence of events	9-1
Section 1. Setting up your workspace	9-2
Section 2. Creating a fraud management rule agent	9-3
2.1. Creating the rule agent	9-3
2.2. Creating a rule: Ask client to confirm within 60 minutes	9-4
2.3. Creating a rule: Remind after 30 min if no action taken	9-4
Section 3. Deploying the solution	9-6
3.1. Exporting the solution	9-6
3.2. Deploying automatically with Ant tasks	9-7
Section 4. Testing the solution	9-8
Exercise 10. Deploying solutions	10-1
Section 1. Deploying and removing a solution manually	10-2
1.1. Managing deployed solutions	10-2
1.2. Undeploying and deleting solution files	10-3

Exercise 11. Defining connectivity for a solution	11-1
Section 1. Setting up your workspace	11-2
Section 2. Defining connectivity	11-3
Section 3. Exporting the solution and connectivity	11-5
3.1. Exporting the solution	11-5
3.2. Exporting the connectivity configuration	11-5
3.3. Using solutionManager to manually deploy the solution	11-6
3.4. Using connectivityManager to deploy the connectivity	11-7

Trademarks

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

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

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

CICS®	developerWorks®	Express®
IBM SmartCloud®	ILOG®	Insight™
Insights™	PartnerWorld®	Redbooks®
SPSS®	WebSphere®	z/OS®

Intel, Intel Xeon and Xeon 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.

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.

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

Exercises description

This course includes the following exercises:

- Exercise 1, "Getting started with Decision Server Insights"
- Exercise 2, "Creating a solution in Insight Designer"
- Exercise 3, "Defining the business model"
- Exercise 4, "Creating a rule agent"
- Exercise 5, "Writing and testing rules"
- Exercise 6, "Using global aggregates in rules"
- Exercise 7, "Using event aggregates in rules"
- Exercise 8, "Using time-based and location-based reasoning in rules"
- Exercise 9, "Testing for the absence of events"
- Exercise 10, "Deploying solutions"
- Exercise 11, "Defining connectivity for a solution"

For the first four exercises, you start from scratch to become familiar with the product installation, set up your workspace, and create a solution. For the remaining exercises, some additional projects are provided for you to help you test the behavior of the business logic that you implement.

The exercises build on each other and you are encouraged to complete each exercise in order before continuing. However, solution projects are also provided if you are unable to finish an exercise.

How to follow the exercise instructions

Exercise structure

Each exercise is divided into sections with a series of numbered steps and lettered substeps:

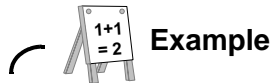
- The numbered steps (1, 2, 3) represent actions to be performed.
- The lettered substeps (a, b, c) provide detailed guidance on how to complete the action.



Information

If you already understand how to perform the action in the numbered step, you can skip the specific guidance in the lettered substeps.

Here is an example from Exercise 1 of this course.



Excerpt from Exercise 1

- __ 1. Start IBM Installation Manager.
 - __ a. Go to **Start > All Programs > IBM Installation Manager**.
 - __ b. Click **IBM Installation Manager**.

In this example, the numbered instructions say to start IBM Installation Manager. The “a” and “b” substeps provide specific guidance on the menu steps to find and start the tool.

Text highlighting in exercises

Different text styles indicate various elements in the exercises.

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

- Menu items
- Field names
- Icons

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

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

Tracking your progress

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

You are encouraged to use these markers to track your progress by checking off each step as you complete it. Tracking your progress in this manner might be useful if you are interrupted while working on an exercise.

Required exercise sections

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

Dependencies between exercises are listed in the exercise introduction. If you are unable to complete an exercise, you can use the solution project for that exercise as the starting point for the next one.

File references

Exercise steps contain references to files or projects to open or import. Two directories are used in these references:

- **InstallDir:** This directory is the IBM Decision Server Insights V8.7 installation directory, which includes two subdirectories.
- **TrainingDir:** This directory contains the files that are required during demonstrations, exercises, and the workshop steps, such as samples of code that you can copy and paste.

If you are using the VMware image that is provided with this course:

- **InstallDir** is: C:\IBM\ODMInsights87

This folder is the default IBM Operational Decision Manager installation directory on Windows.

- **TrainingDir** is: C:\labfiles

If you are not using the VMware image that is provided with this course, ask the installer of your environment, or your instructor, where to find the **InstallDir** and **TrainingDir** directories.



Stop

Make sure that you identify the **InstallDir** and **TrainingDir** directories before you proceed with the exercises in this course.

Projects for exercises

Most of the exercises for this course are done in Insight Designer, which uses the Decision Insight perspective of Eclipse.

The exercise projects are provided for you to import into Eclipse.

To open Insight Designer, you click **Start > All Programs > IBM > Decision Server Insights V8.7 > Rule Designer** (or you can use the shortcut on the desktop).



When prompted for a workspace, you can type the path directly in the Workspace Launcher, for example:

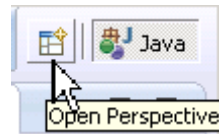
`C:\labfiles\myWorkspace`

When you type a path, an empty workspace is created and opens in the Java perspective.

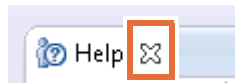


Troubleshooting

With Eclipse 4.2.2, the Rule perspective does not open directly, so you must close the Welcome view and manually switch perspectives by clicking the **Open Perspective** icon in the upper-right corner of the Eclipse window.



When you switch to the Decision Insight perspective, to give yourself more area to work in, you can close the **Help** pane by clicking the **X**.



Using the product documentation

The product documentation is installed locally on the VMware image that is provided with this course.

To access the local documentation while working in Rule Designer, you must first start it by clicking **Start > All Programs > IBM > Decision Server Insights V8.7 > Help and Support > Start Information Center (local)**.

You can also access the product documentation from a web browser at this web address:

`www.ibm.com/support/knowledgecenter/SSQP76_8.7.0/
com.ibm.odm.distrib/kc_welcome_odm_distrib.html`



Information

If you are not using the VMware image that is provided with this course, you must either install the local help as described in the product documentation, or use the online version.

Exercise 1. Getting started with Decision Server Insights

What this exercise is about

This exercise explores the installation and configuration of Decision Server Insights.

What you should be able to do

After completing this exercise, you should be able to:

- Install Decision Server Insights with IBM Installation Manager
- Prepare a workspace in Insight Designer
- Set the debug port for your installation

Introduction

This exercise includes these sections:

- Section 1, "Installing Decision Server Insights V8.7"
- Section 2, "Exploring your Decision Server Insights installation"
- Section 3, "Setting up the Decision Insight perspective"
- Section 4, "Starting the server from Insight Designer"
- Section 5, "Setting the debug port"

Requirements

This exercise requires that you have the Decision Server Insights V8.7 installation package that is downloaded on your computer.



Important

The exercises in this course use a set of lab files that include start projects, solution files, and code snippets. The course lab files are in the following directory:

C:\labfiles (also referred to as TrainingDir)

The exercises point you to the lab files as you need them. For this first exercise, you do not use the TrainingDir files.

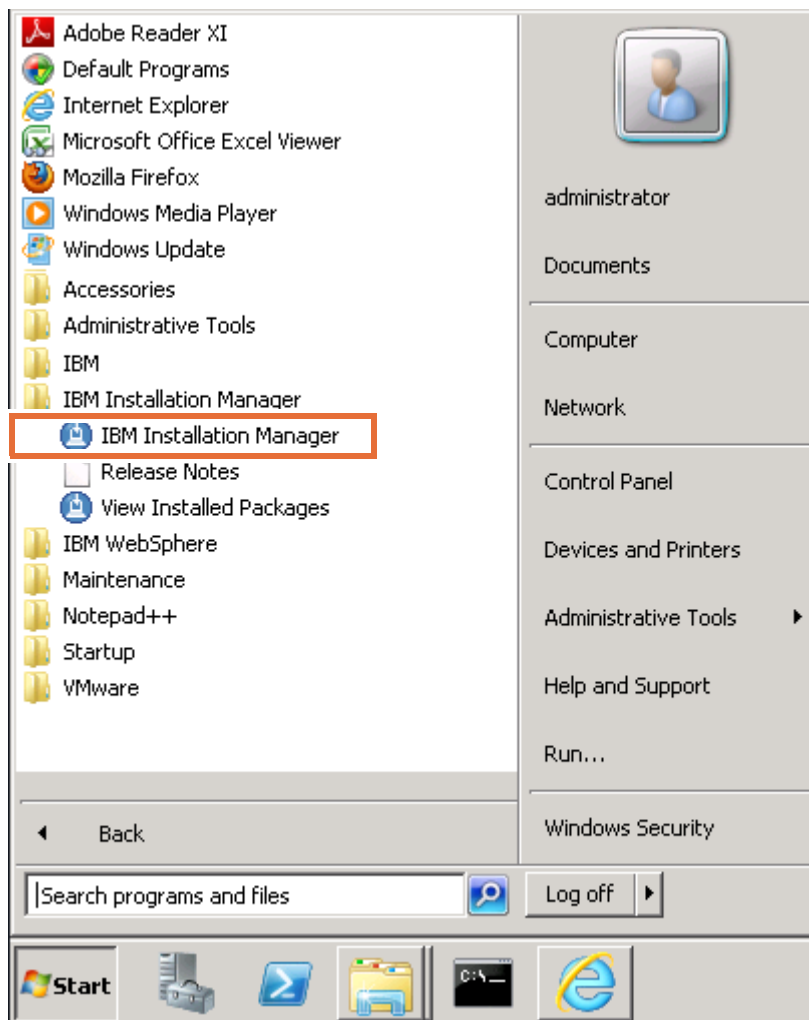
Section 1. Installing Decision Server Insights V8.7

The Decision Server Insights V8.7 installation file is packaged separately from the installation files for the main IBM Operational Decision Manager product. You install Decision Server Insights by using IBM Installation Manager.

1.1. Exploring the installation steps

To install Decision Server Insights, you must first add the Operational Decision Manager Insights installation file as a repository in IBM Installation Manager.

- ___ 1. Start IBM Installation Manager.
 - ___ a. Click **Start > All Programs > IBM Installation Manager**.
 - ___ b. Click **IBM Installation Manager**.



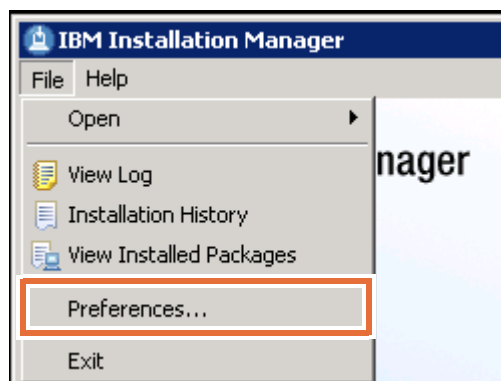
IBM Installation Manager opens.

- ___ 2. View how to add the Decision Server Insights installer file as an IBM Installation Manager repository.

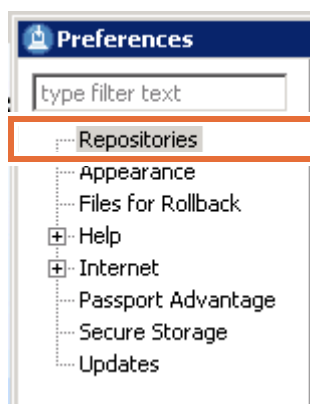
**Note**

The Decision Server Insights installation file is packaged as a TAR file that you extract. The extracted folder includes a **disk5** folder with a `repository.config` file. To install Decision Server Insights with Installation Manager, you add the path to the `repository.config` file as a repository.

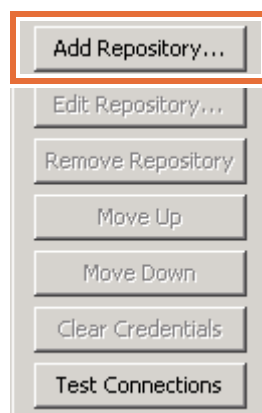
- ___ a. In the IBM Installation Manager main window, click **File > Preferences**.



- ___ b. Make sure that you are on the Repositories page.



- ___ c. Click **Add Repository**.



**Note**

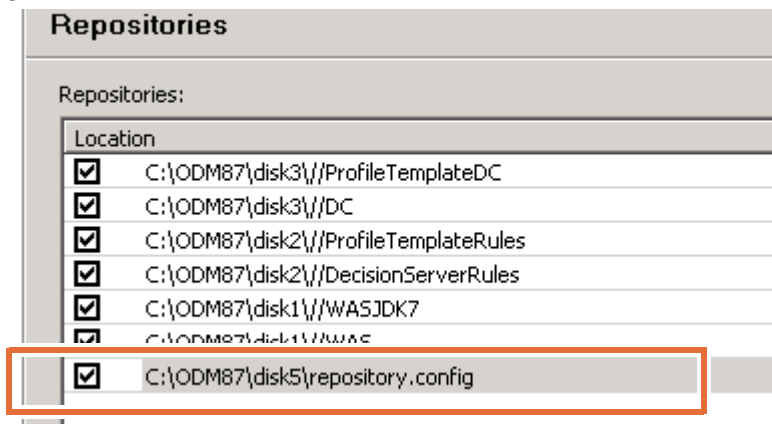
For this course, Decision Server Insights is already installed for you. You do not add the repository.

However, if you needed to add the repository, you would click **Browse** and go to the directory where you saved the Decision Server Insights V8.7 installation **disk5** folder, and select the `repository.config` file.

For this exercise, you explore your installation.

- ___ d. Click **Cancel**.

The installation path to the Decision Server Insights installation repository is listed in Repositories.



- ___ 3. Click **Cancel** to exit and close the Preferences window.

- ___ 4. Explore installation steps for Decision Server Insights.
- ___ a. In the IBM Installation Manager window, click **Modify**.



Note

Click **OK** to close the warning that the repositories are not connected.

You see a list of the Installed packages, including Decision Server Insights.

Package Group Name	Directory
<input checked="" type="checkbox"/> Decision Server Insights V8.7	C:\IBM\ODMInsights87
<input type="checkbox"/> IBM WebSphere Application Server Network Deployment V8.5.5	C:\Program Files\IBM\ODM87\WAS\AppServer
<input type="checkbox"/> Operational Decision Manager V8.7	C:\Program Files\IBM\ODM87\ODM

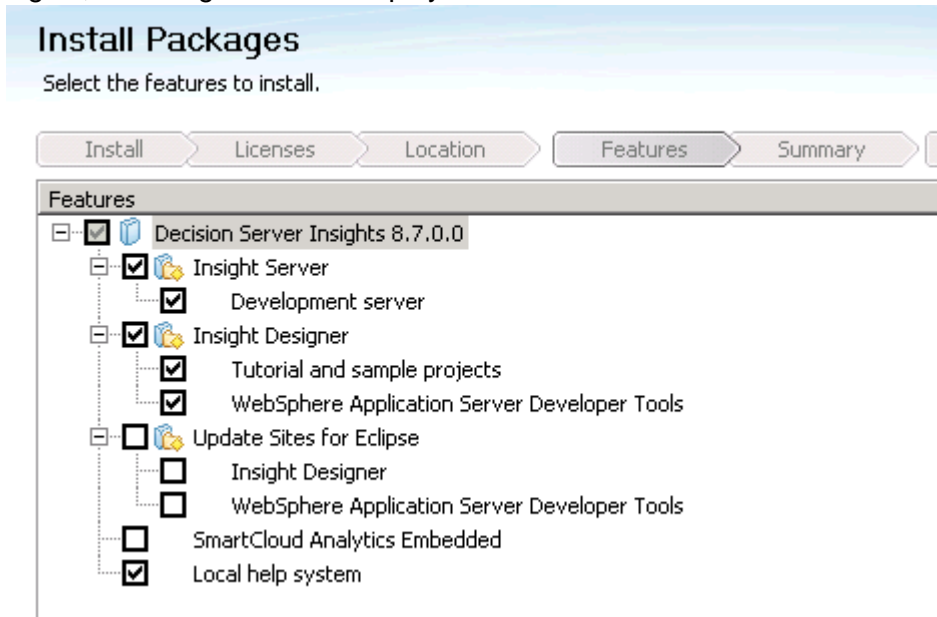
- ___ b. With **Decision Server Insights Version 8.7.0.0** selected, click **Next**.

You see a list of the translations that are available.

Translations			
<input checked="" type="checkbox"/> English	<input type="checkbox"/> Brazilian Portuguese	<input type="checkbox"/> Dutch	<input type="checkbox"/> French
<input type="checkbox"/> Italian	<input type="checkbox"/> Japanese	<input type="checkbox"/> Korean	<input type="checkbox"/> Polish
<input type="checkbox"/> Simplified Chinese	<input type="checkbox"/> Spanish	<input type="checkbox"/> Traditional Chinese	

- ___ c. Click **Next**.

In the Features section, you see the list of installed components for Decision Server Insights, including the Local help system.

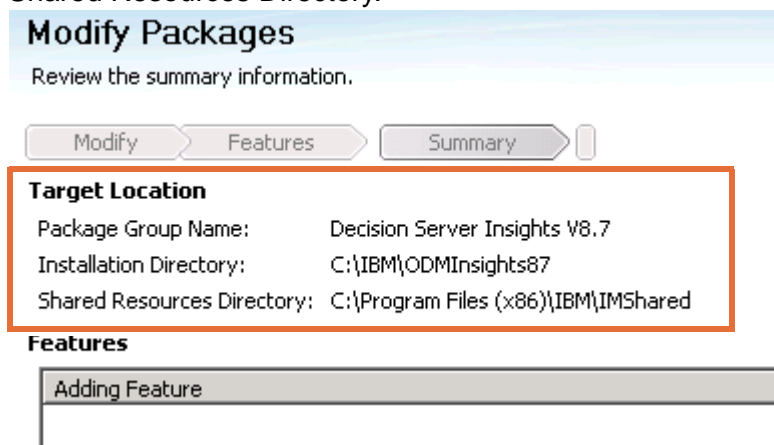


Note

You do not select **Update Sites for Eclipse** to avoid conflicts with your IBM Operational Decision Manager installation.

___ d. Click **Next**.

The summary page lists the Target Location paths, including the Installation Directory and the Shared Resources Directory.



For this course, the default installation path is:

C:\IBM\ODMInsights87

This course uses **InstallDir** to reference the installation path. The installation path for your environment might differ.

- __ e. Click **Cancel**.
- __ 5. Close Installation Manager.

**Warning**

When you install Decision Server Insights, you are also prompted to choose the installation path. To avoid potential conflicts with user permissions in Operational Decision Manager and Decision Server Insights, you must install Decision Server Insights in a separate directory from Operational Decision Manager.

By default, Operational Decision Manager is installed in the `C:\Program Files\IBM\ODM87` directory. Make sure that you install Decision Server Insights outside of Program Files and Program Files (x86) to avoid user privilege conflicts.

Section 2. Exploring your Decision Server Insights installation

- ___ 1. In Windows Explorer, open InstallDir (or C:\IBM\ODMInsights87 directory).



Note

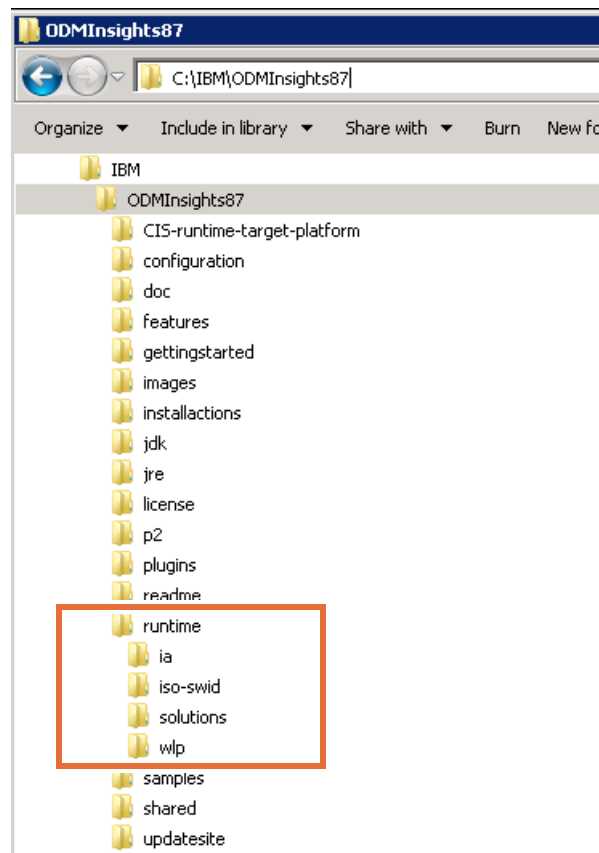
For this course, Decision Server Insights is installed in the C:\IBM\ODMInsights87 directory.

- ___ 2. Create a desktop shortcut to the installation directory.
- ___ a. Right-click **ODMInsights87** and click **Send to > Desktop (create shortcut)**.
- ___ b. You can rename the shortcut: Shortcut to DSI InstallDir



During the labs, you can use this shortcut for quick access to the installation directory.

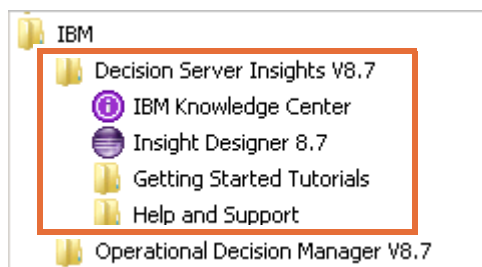
- ___ 3. Expand the ODMInsights87 folder to view the installed files and folders.



- ___ 4. Notice the **runtime** folder.

During the course, you refer often to this directory to access and manage solution files and WebSphere Liberty Profile.

- ___ 5. Close Windows Explorer.
- ___ 6. From the **Start** menu, click **All Programs > IBM > Decision Server Insights V8.7** and notice the Decision Server Insight menu options.



- ___ 7. Right-click **Insight Designer 8.7** and click **Send to > Desktop (create shortcut)**.
Throughout the labs, you work with **Insight Designer 8.7**. You can use this shortcut for quick access to the tool.

Section 3. Setting up the Decision Insight perspective

In this section, you see how to start the sample server from the Samples Console perspective in Insight Designer.

3.1. Opening Insight Designer

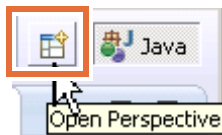
- ___ 1. Open Insight Designer by either clicking **Insight Designer 8.7** in the **Start** menu (which should still be open) or by double-clicking your new **Insight Designer 8.7** shortcut on the desktop.
- ___ 2. Create an empty workspace in the Workspace Launcher.
 - ___ a. In the **Workspace** field, type the following name to create an empty workspace:
`C:\labfiles\workspace-start`
 - ___ b. Click **OK**.
 - ___ c. When the workspace opens, close the **Welcome** view.
By default, Insight Designer opens in the Java perspective.

3.2. Switching to the Decision Insight perspective

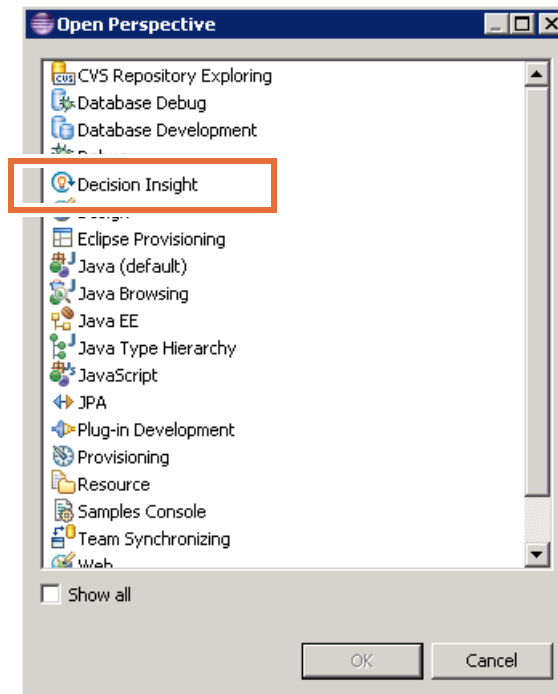
Throughout this course, you use these perspectives:

- Java
- Decision Insight
- Samples Console

- ___ 1. Switch to the Decision Insight perspective.
 - ___ a. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.



- ___ 2. In the Open Perspective dialog box, select **Decision Insight**.



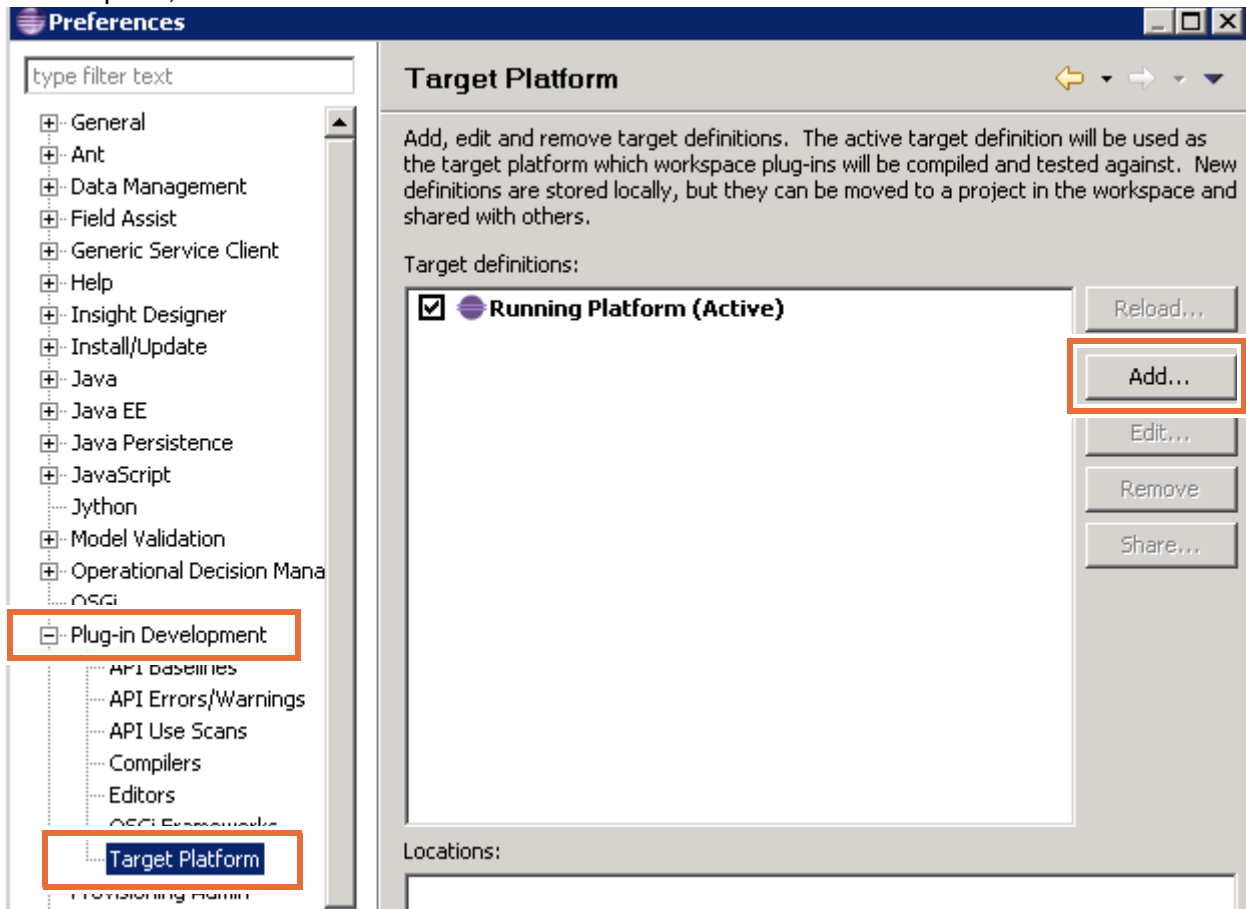
The Decision Insight perspective opens.

- ___ 3. If the **Help** pane is open, you can close it.

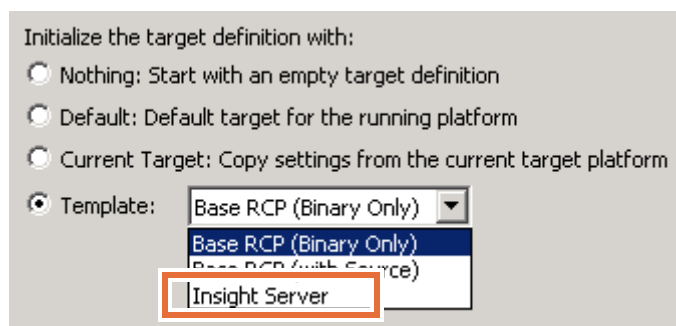
3.3. Setting the target platform

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

- ___ 2. Expand **Plug-in Development**, click **Target Platform**, and when the Target Platform page opens, click **Add**.



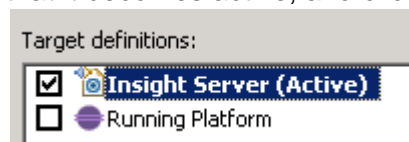
- ___ 3. On the Target Definition page, select **Template** and select **Insight Server** from the list, click **Next**.



- ___ 4. Click **Finish**.

On the Target Platform page, Insight Server is now in the list of Target definitions.

- ___ 5. Select **Insight Server** so that it becomes active, and click **OK**.

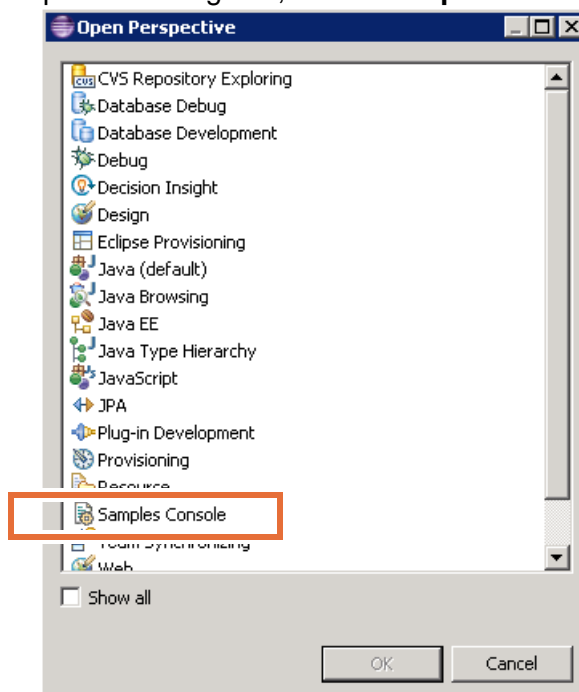


Section 4. Starting the server from Insight Designer

- ___ 1. Switch to the Samples Console perspective.
 - ___ a. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.



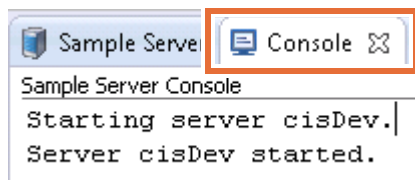
- ___ b. In the Open Perspective dialog box, select **Samples Console**.



- ___ 2. In the **Sample Server** view in the lower part of the workspace, click the **Start the sample server** icon to start the server.



The default server is called cisDev. The Console opens with the message that the server is started.



Section 5. Setting the debug port

The debug port is part of your installation configuration. After you set the debug port for your installation, the port value can be found in the `server.xml` file.

- __ 1. Use command prompt and the `propertyManager` to set the debug port to 6543.



Note

The server must be started before you can run the `propertyManager`.

- __ a. Open a command prompt window and go to the `C:\IBM\ODMInsights87\runtime\ia\bin` directory.
 - Type `cd ..\..` and press Enter.
 - Type `cd C:\IBM\ODMInsights87\runtime\ia\bin` and press Enter.
- __ b. Type the following command, and press Enter.

```
propertyManager set --username=tester --password=tester debugPort=6543
```

```
C:\IBM\ODMInsights87\runtime\ia\bin>propertyManager set --username=tester --password=tester debugPort=6543
Set property successful. Property name: debugPort, old value: null, new value: 6543
```

```
C:\IBM\ODMInsights87\runtime\ia\bin>
```

- __ 2. Wait for the “Set property successful” message and close the command prompt window.
- __ 3. Verify the port setting in the `server.xml` file.
 - __ a. In Windows Explorer, go to the `InstallDir\runtime\wlp\usr\servers\cisDev` directory.



Reminder

The default installation path is: `C:\IBM\ODMInsights87`

- __ b. Open the `server.xml` file in a text editor and scroll to the end of the file to see the newly added `ia_runtime debugPort` property value:


```
<ia_runtime debugPort="6543"/>
```
- __ c. Close the `server.xml` file and Windows Explorer.



Note

By default, the debug port is set to null and is not listed in the `server.xml` file. The `ia_runtime debugPort` property is listed in this file only after you set the value.

End of exercise

Exercise review and wrap-up

The first part of the exercise looked at the installation of Decision Server Insights. You also saw how to open Insight Designer and prepare your environment to start development.

Exercise 2. Creating a solution in Insight Designer

What this exercise is about

This exercise demonstrates how to create the solution project in Insight Designer.

What you should be able to do

After completing this exercise, you should be able to:

- Create a solution project

Introduction

This exercise includes these sections:

- Section 1, "Creating a solution project"

Requirements

This exercise requires that you use the workspace that you created in Exercise 1, "Getting started with Decision Server Insights".

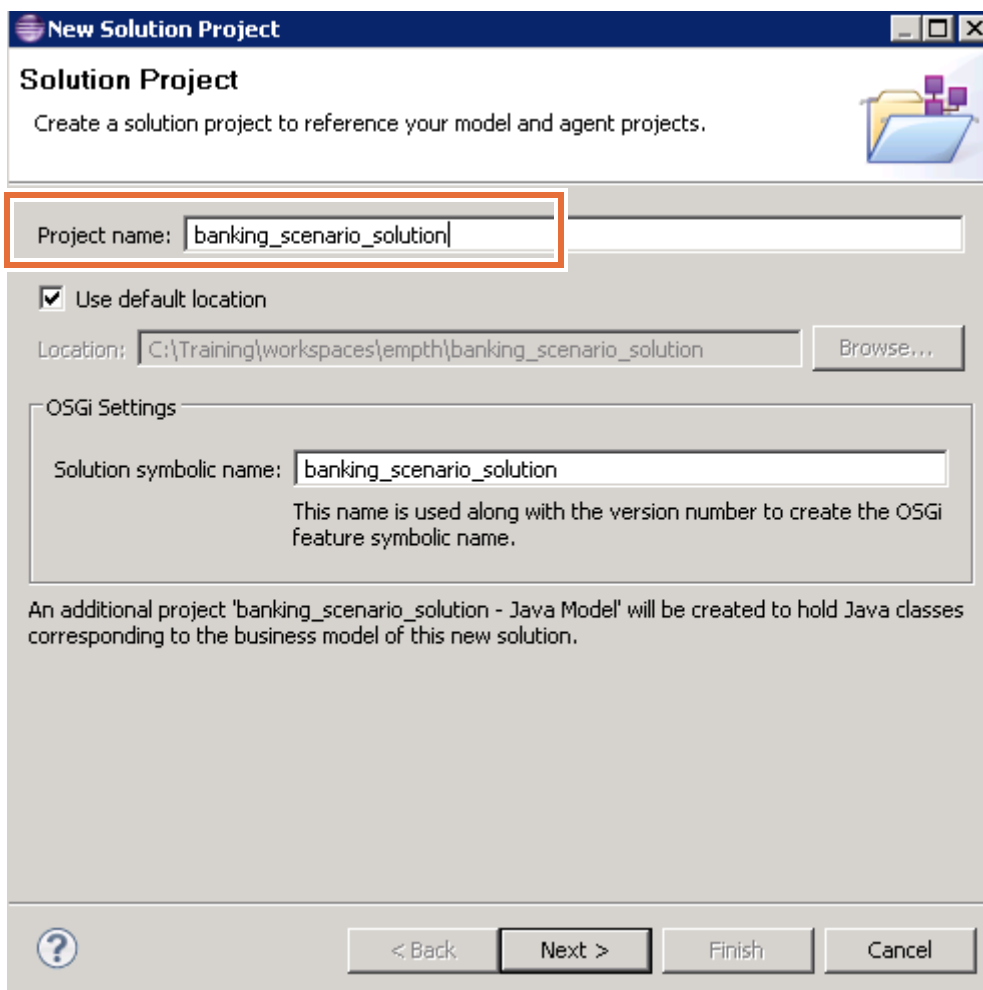
Section 1. Creating a solution project

All solution artifacts are managed through a solution project. In this section, you create a solution project and its referenced projects.

- ___ 1. If Insight Designer is closed, open it by double-clicking the Insight Designer shortcut on your desktop.
- ___ 2. In Insight Designer, switch to the Decision Insight perspective by clicking **Decision Insight** on the perspective toolbar.

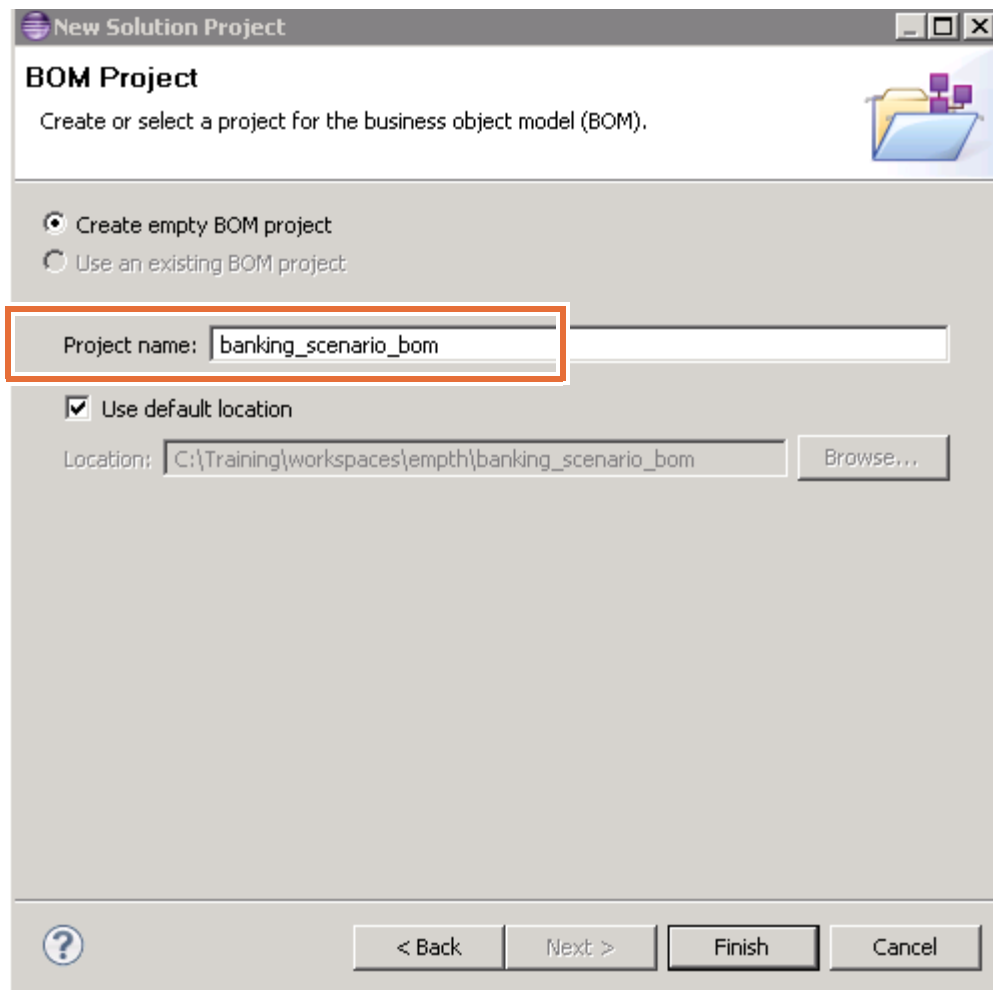


- ___ 3. From the **File** menu, click **New > Solution Project**.
- ___ 4. In the **Project name** field, type: `banking_scenario_solution`



- ___ 5. Click **Next**.

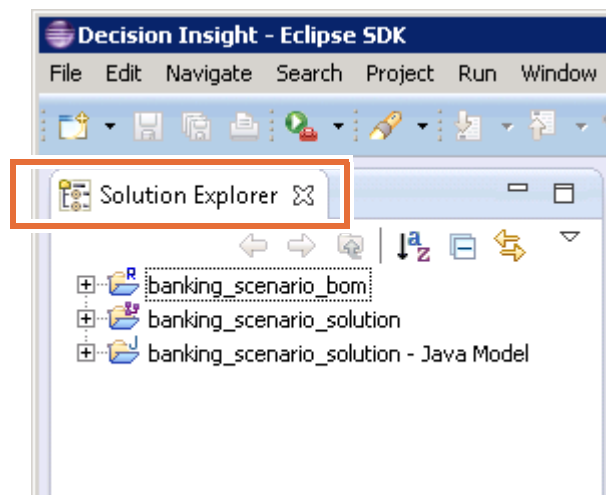
- ___ 6. On the BOM Project page, type a project name for the BOM: `banking_scenario_bom`



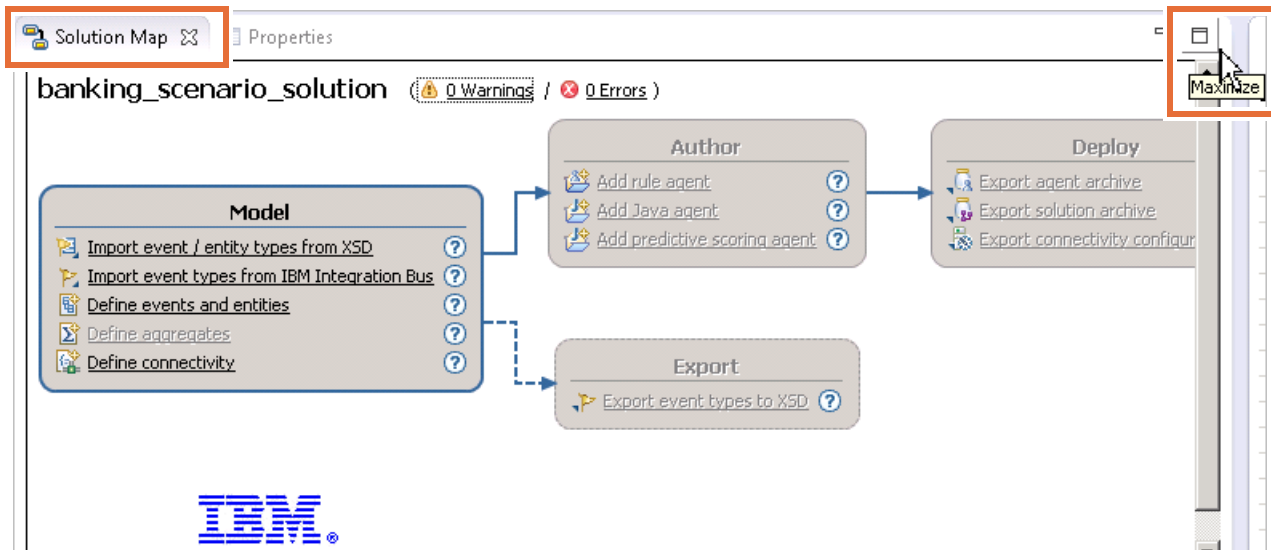
- ___ 7. Click **Finish**.

You now have three projects in Solution Explorer.

- `banking_scenario_bom`
- `banking_scenario_solution`
- `banking_scenario_solution - Java Model`



___ 8. In Solution Explorer, select **banking_scenario_solution** to open the **Solution Map** view.



Hint

You can use the maximize icon in the upper-right corner of the Solution Map view to open it to full view, and then restore it to the lower pane.

The Solution Map guides you through the steps of solution development. Notice that most of the links are not yet enabled because some tasks in the first goal, **Model**, must be complete before you can start on other goals.

End of exercise

Exercise review and wrap-up

This exercise showed you how you create a solution project and the associated business object model (BOM) project. The generated Java project is empty. During the next exercise, you see how you model the entities and events by populating the BOM project.

Exercise 3. Defining the business model

What this exercise is about

This exercise covers how to create a business model.

What you should be able to do

After completing this exercise, you should be able to:

- Create a business model definition file

Introduction

This exercise includes these sections:

- Section 1, "Modeling the domain"
- Section 2, "Creating the business model definition"

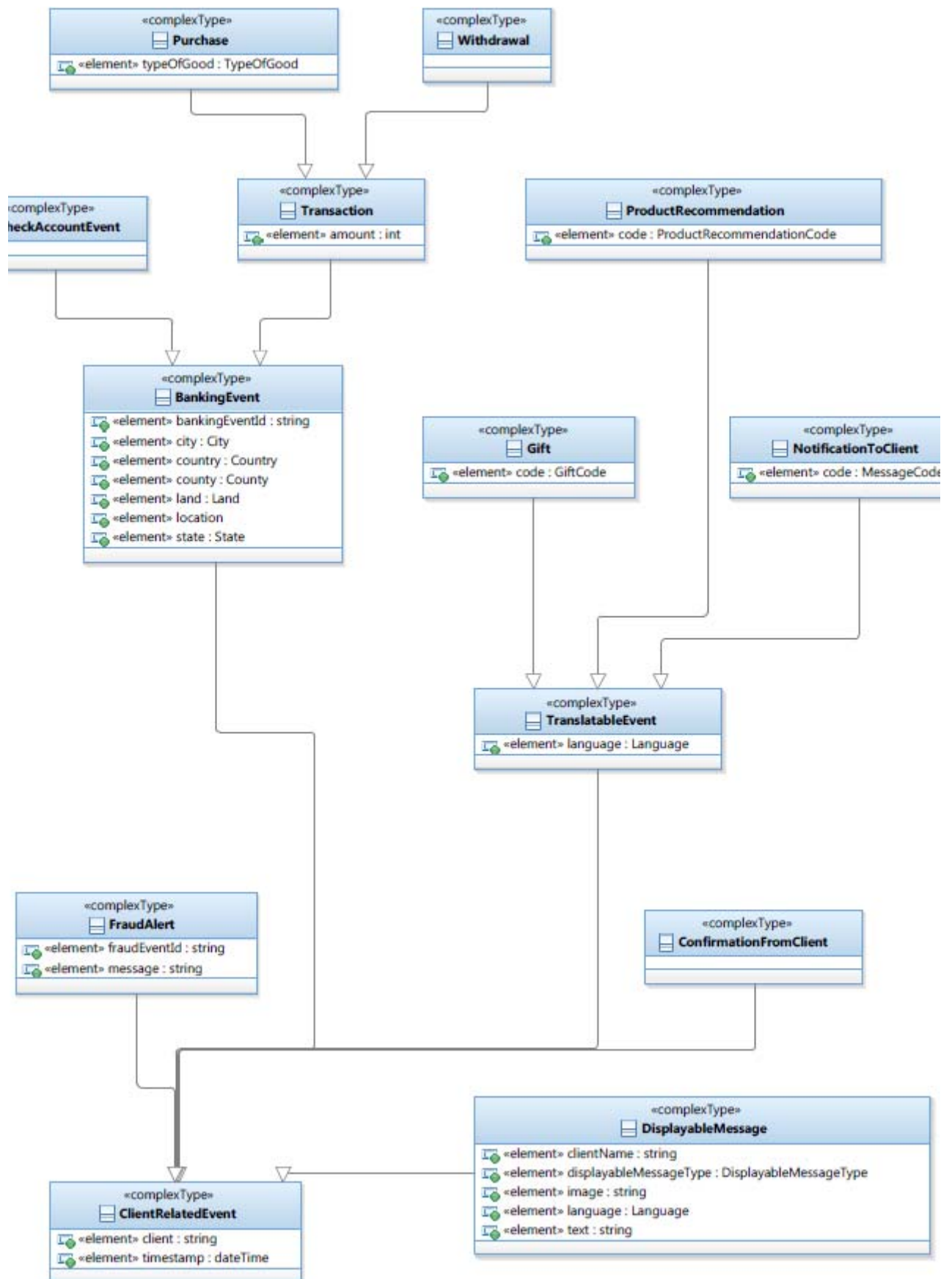
Requirements

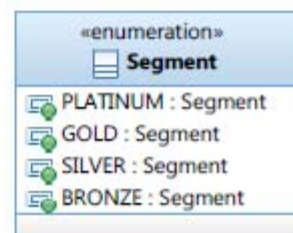
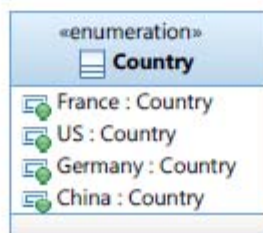
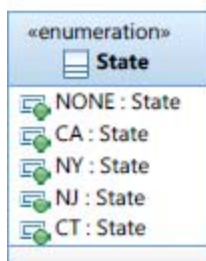
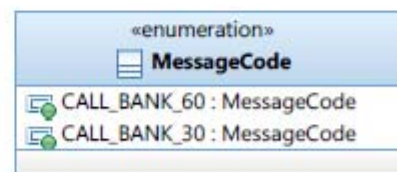
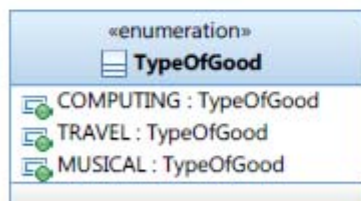
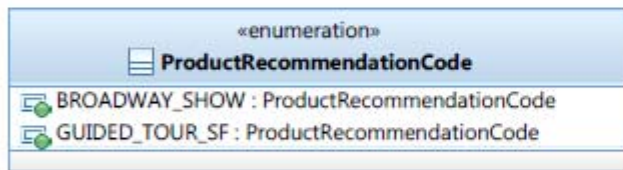
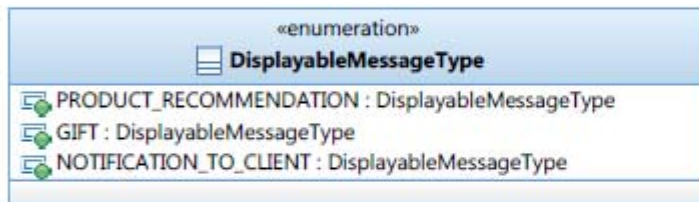
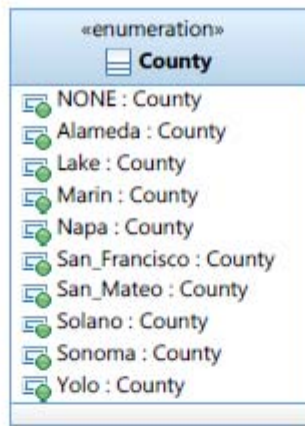
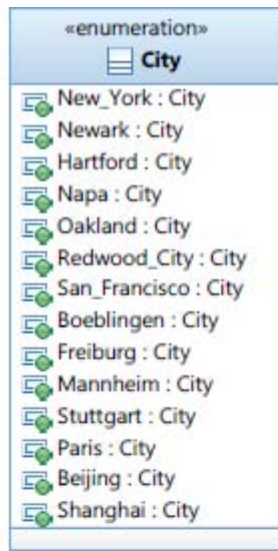
This exercise requires that you continue in the workspace that you created in Exercise 1, "Getting started with Decision Server Insights".

Section 1. Modeling the domain

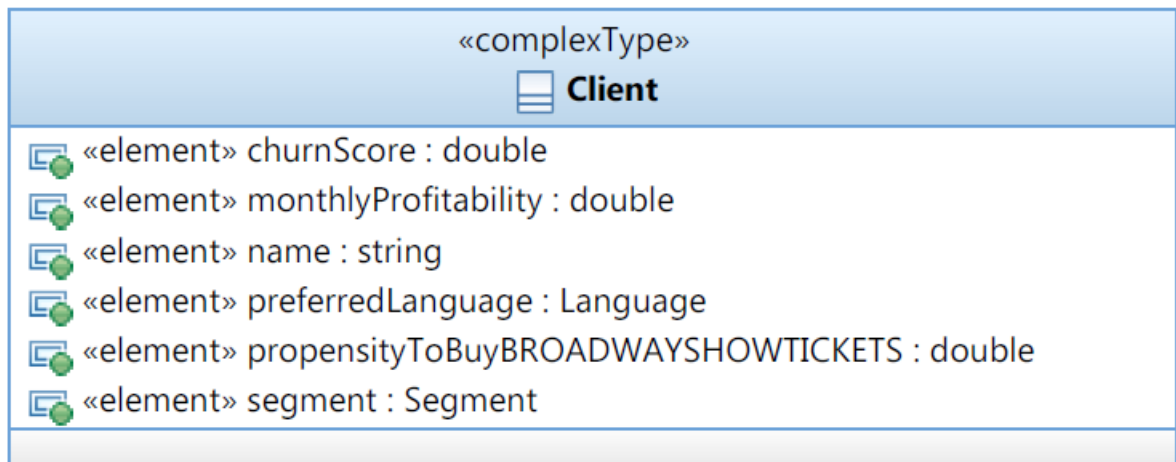
Before you can define how to process events, you must first define which events to monitor, which entities are involved, and the relationships between entities and events. Modeling the entities in your domain is key to creating a complete business model definition.

___ 1. Look at the business model diagrams that are shown here.

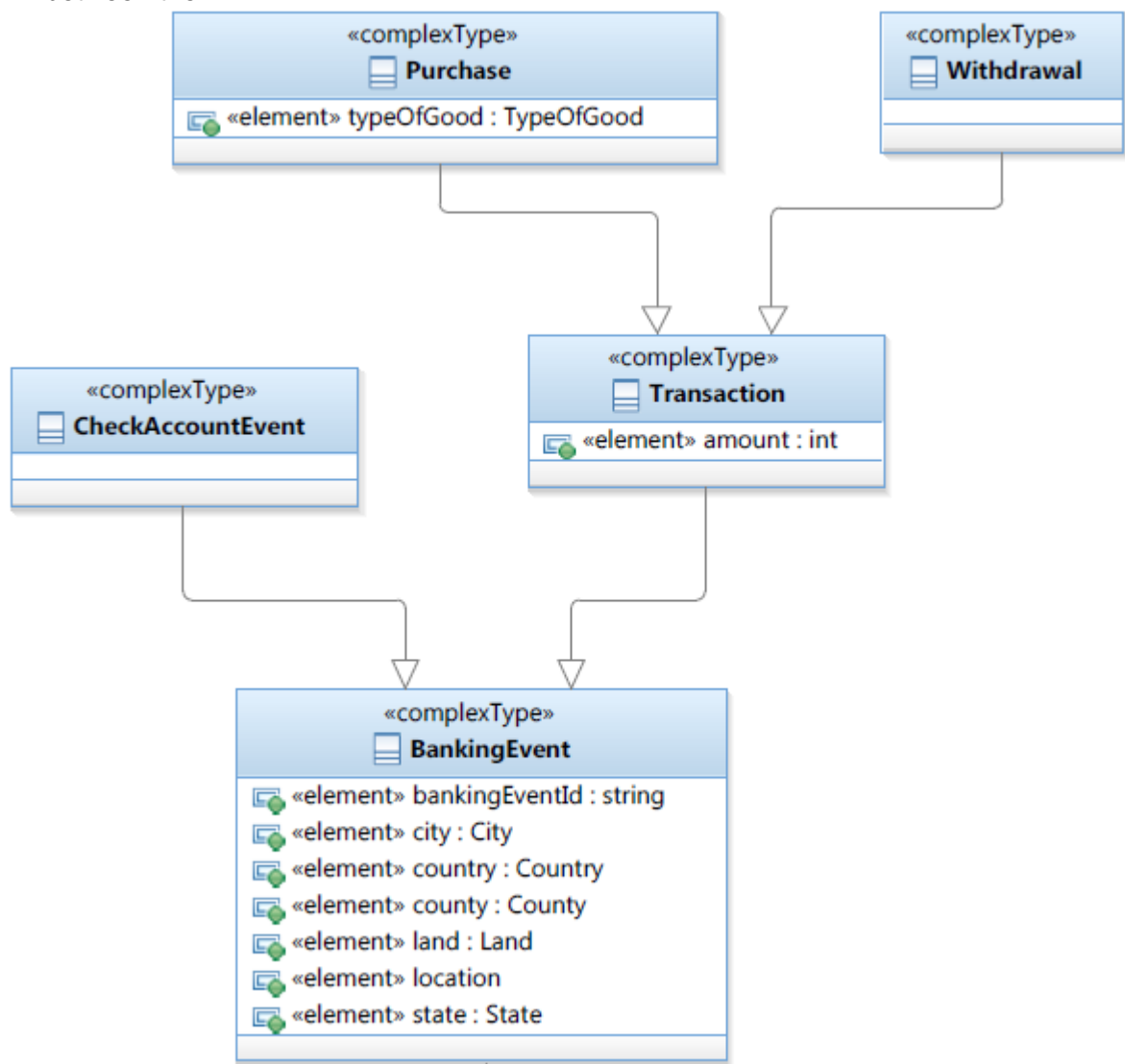




- ___ 2. Look the Client class diagram and note the attributes and types.



- ___ 3. Notice the events, such as BankingEvent and Transaction, and the inheritance relationships between them.

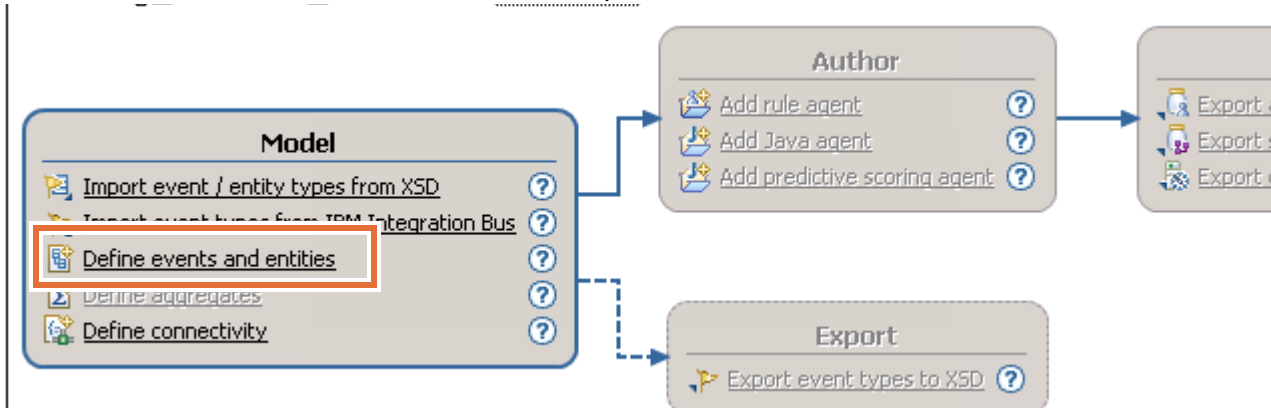


The entities and events in the domain model must be defined in the business model for your solution.

Section 2. Creating the business model definition

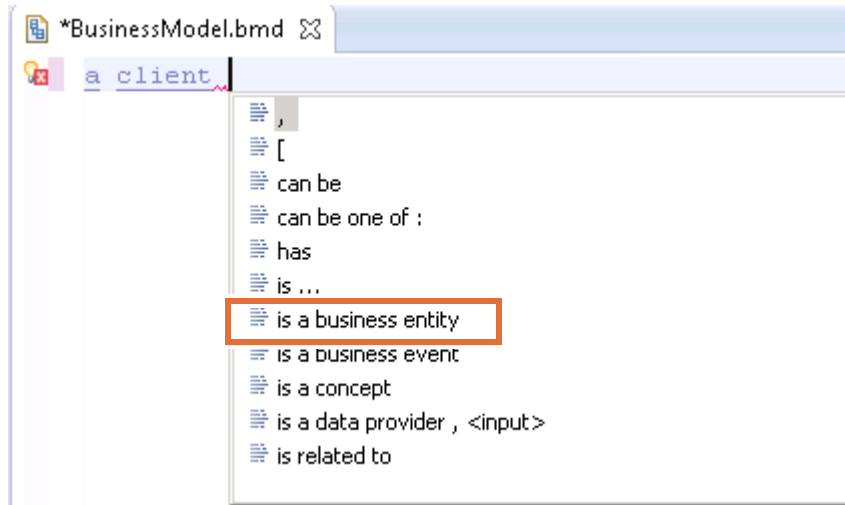
Next, you transfer the information from the domain models to a business model definition (BMD) file.

- ___ 1. In Insight Designer, make sure that you are in the same workspace that you used during Exercise 2, "Creating a solution in Insight Designer".
- ___ 2. Use the Solution Map to create a BMD file.
 - ___ a. In Solution Explorer, click **banking_scenario_solution** to open the **Solution Map** view.
 - ___ b. In the Model task of the Solution Map, click **Define events and entities**.

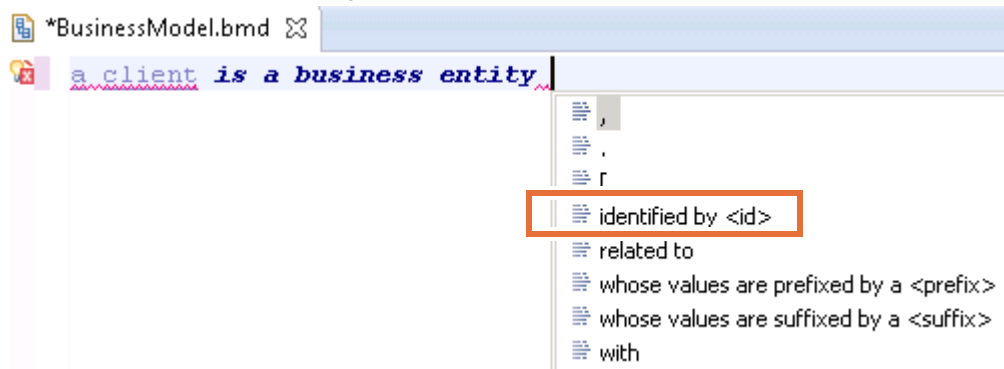


- ___ c. In the New Business Model Definition wizard, in the **Package** field, type: `banking_scenario`
- ___ d. Click **Finish**.
An empty `BusinessModel.bmd` file is created in the **banking_scenario_bom > bom > banking_scenario** folder and opens in the editor.
- ___ 3. Create a Client entity definition.
 - ___ a. Look again at the Client object in the UML diagram and its attributes:
 - name
 - segment
 - churn score
 - monthly profitability
 - propensity to buy BROADWAYSHOWTICKETS
 - preferred language
 - ___ b. In the BMD editor, start the definition by typing: `a client`

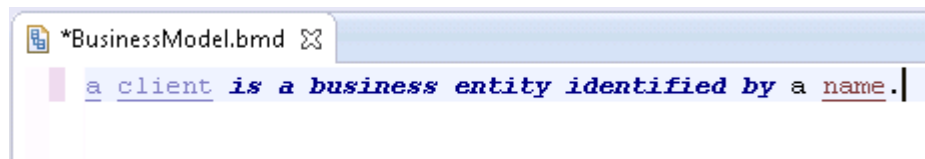
- ___ c. When you press Space and the editor prompts you for the next part of the statement, choose **is a business entity**.



- ___ d. Next, choose **identified by <id>** from the list.



- ___ e. Complete the statement by typing: a name.

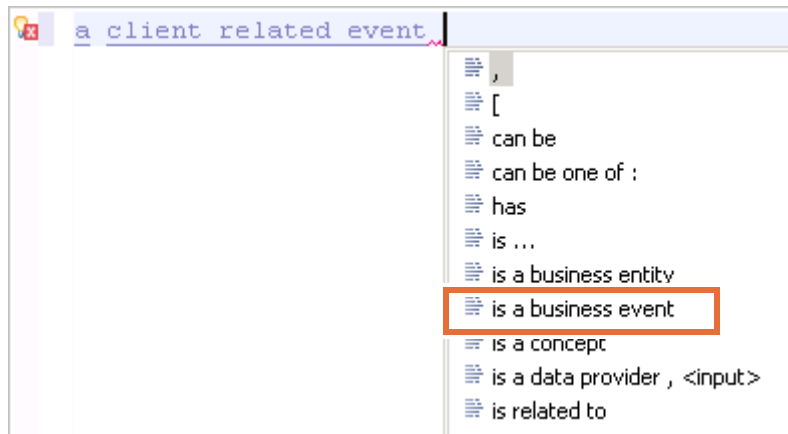


Hint

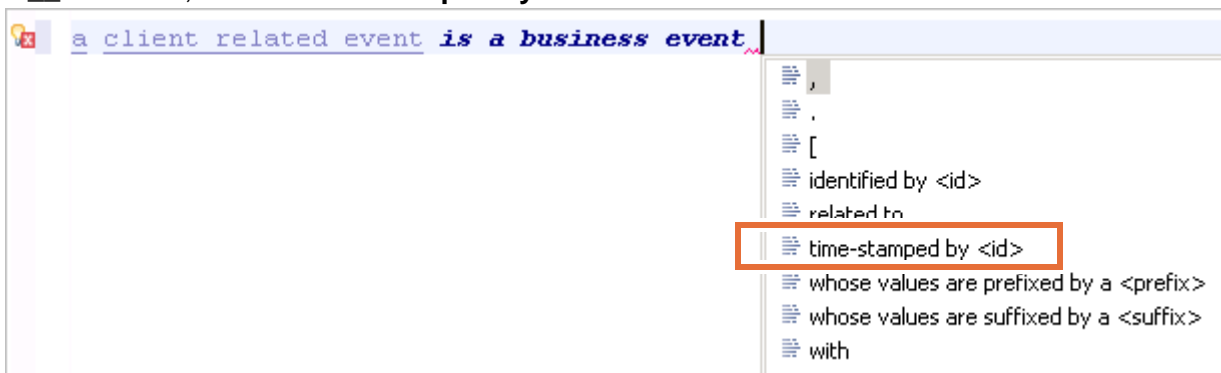
Make sure that you include a period (.) at the end of the definition phrase.

- ___ f. Add an attribute to the Client entity by typing this line:
a client has a segment.
By default, attributes are of type String.
- ___ g. You can complete the Client by adding the remaining attributes.

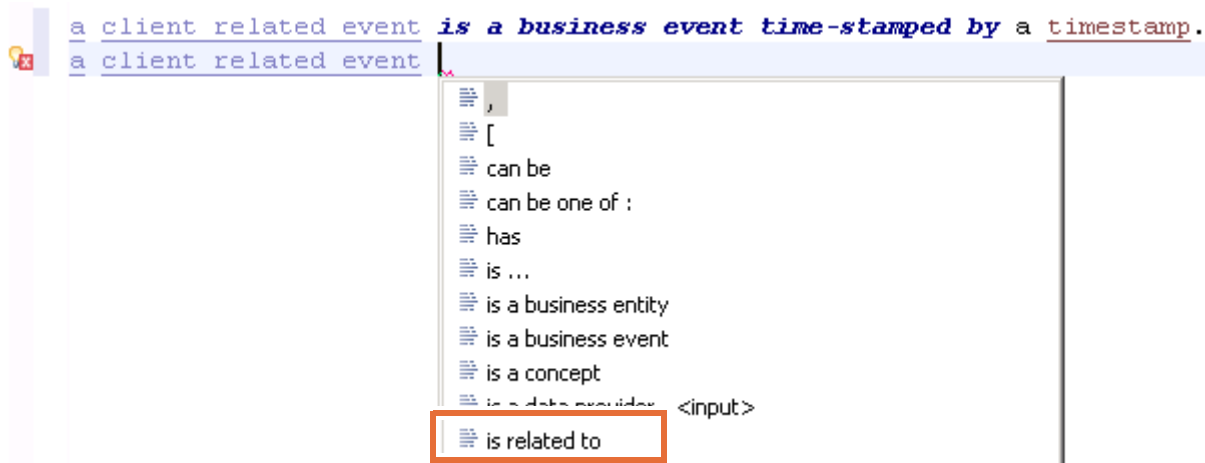
- ___ 4. Define an event.
- ___ a. Look again at the UML diagram, and note BankingEvent, which inherits from the ClientRelatedEvent, and the event attributes.
- ___ b. Define ClientRelatedEvent by typing: a client related event
- ___ c. When you press Space and the editor prompts you for the next part of the statement, choose **is a business event**.



- ___ d. Next, choose **time-stamped by <id>**.



- ___ e. Type: a timestamp .
- ___ f. Define the relationship between the client-related event and the client by using the “is related to” construct.



- __ g. Complete the statement by typing: a `client`.



Questions

How would you define the inheritance relationship between `BankingEvent` and `ClientRelatedEvent`?

- __ h. Define the `BankingEvent` by using the “is a” construct, and typing this line:
`a banking event is a client related event.`



Hint

The complete business model is defined for you in the `BMD.txt` file of **TrainingDir\code** folder.

- __ 5. To complete the business model, copy the content from the `BMD.txt` file in the **TrainingDir\code** directory, and paste it in the business model editor to overwrite your definitions.
- __ a. Open the `BMD.txt` file and press Ctrl+A, and then press Ctrl+C.
- __ b. In the business model editor, press Ctrl+A, and then press Ctrl+V.
- __ 6. Review the BMD definitions and compare these lines to the UML diagram.
- Is the relevant information from the UML diagram captured in this file?
 - How are “is-a” and “has-a” relationships defined?
 - How do the entity and the event models refer to each other? For example, a `client related event` is related to a `client`.



Information

The business model definition is expressive. You can use the **BOM** tab of the BMD editor to verify that your definitions are correct.

- __ 7. Save your work by pressing Ctrl+S and close the `BusinessModel.bmd` file editor window, and close the `BMD.txt` file.
- __ 8. Keep Insight Designer open for the next exercise.

End of exercise

Exercise review and wrap-up

The first part of the exercise showed you how to define the entity and the event model by using natural language constructs. You also saw how entity and event relationships are defined in the model.

Exercise 4. Creating a rule agent

What this exercise is about

This exercise covers how to create agents, how to write agent descriptors that bind the agent to an entity, and how to write a rule that emits an event.

What you should be able to do

After completing this exercise, you should be able to:

- Create a rule agent
- Write an agent descriptor
- Write a rule that emits an event
- Create a Java agent

Introduction

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Creating a rule agent"
- Section 3, "Writing the San Francisco rule"

Requirements

This exercise requires that you continue in the workspace that you used during the previous exercise.

Section 1. Setting up your workspace

Before you start this exercise, make sure that you have Insight Designer open to the correct workspace. Insight Designer should be open from the previous exercise. You continue to work in that workspace.

1.1. Opening the workspace

- ___ 1. Make sure that you are in the same workspace that you used during Exercise 3, "Defining the business model".

Section 2. Creating a rule agent

In this section, you learn how to create rule agents that process the business events that you defined in your business model.

Scenario: Banks want to encourage client satisfaction by recognizing purchasing trends and recommending banking and non-banking products that match client interests. Banks can provide personalized services by gathering and applying context to their operational business decisions.

The bank can look at account activity to determine whether a client is traveling, perhaps on holiday, and more likely to be interested in tourist activities, as opposed to regular activities, such as grocery shopping. The bank can also use predictive scoring and SPSS to predict the client's propensity to buy certain products. With that type of knowledge, the bank can make accurate recommendations or offer timely promotions, specific to the client's location, and at exactly the right time.



Requirements

The bank's mobile app features personalized recommendations. For example, for clients who are not residents of California, but use their mobile banking app while in the California area, the app suggests a guided tour of San Francisco.

The agent must:

- Be bound to a Client entity
- Subscribe to BankingTransaction events
- Update the entity after sending the recommendation

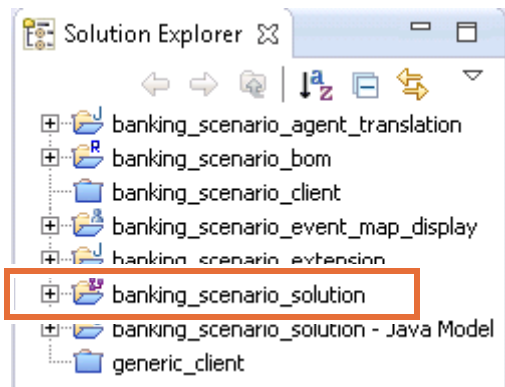
The rule must:

- Determine the location of the client
- Emit an event that carries a product code (GUIDED TOUR SF) and a language code (the client's preferred language)
- Cancel the recommendation if the client recently received a similar recommendation to avoid overwhelming the client with similar notifications

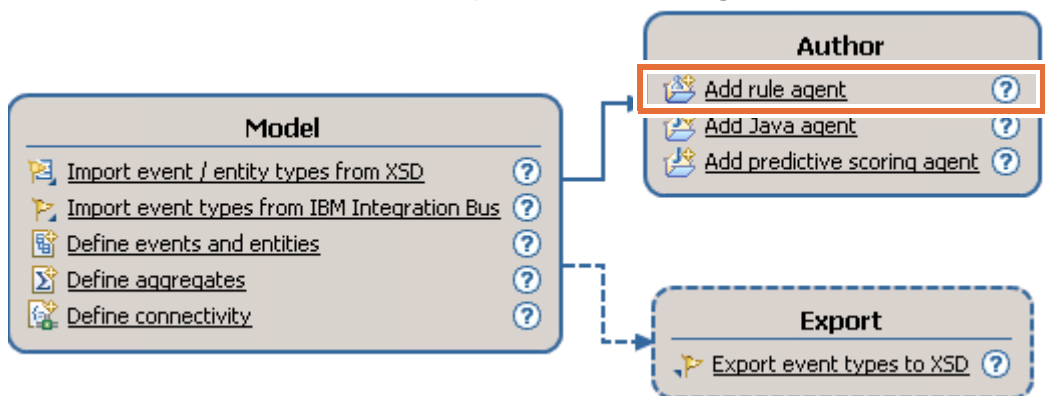
2.1. Creating the product recommendation rule agent

In this step, you create a rule agent that is bound to a client entity. The agent turns this client's past behavior, current activity, and location into *insight* and a product recommendation.

- ___ 1. In Solution Explorer, click **banking_scenario_solution** to make sure that the **Solution Map** view is open.



- ___ 2. In the **Author** task of the Solution Map, click **Add rule agent**.



- ___ 3. In the **Project name** field, type: `banking_scenario_agent_product_recommendation`
- ___ 4. Click **Finish**.

The `agent.adsc` file opens in the editor.

2.2. Writing the agent descriptor



Information

The `agent.adsc` file is the agent descriptor. It defines which entity the agent is bound to and which events the agent subscribes to.

For this rule agent, the descriptor tells the Decision Server Insights runtime environment that an instance of the product recommendation agent is bound to an instance of Client entity.

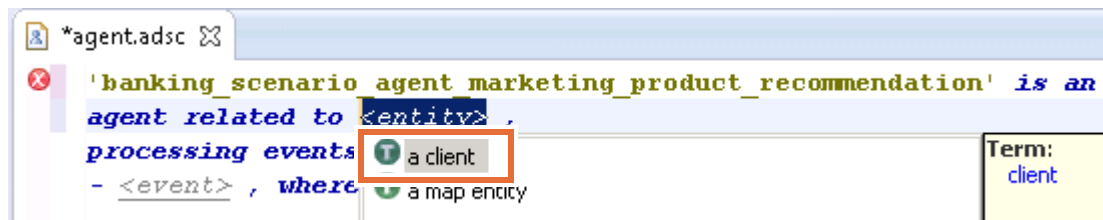
When the Decision Server Insights runtime environment receives any banking event for a client instance, a component of the environment, called the gateway, routes the event to the appropriate agents for that client. Each agent that subscribes to that event evaluates the rules against that

event, combined with the current state of the context (or entity). Because the product recommendation agent is bound to the client entity, the agent can access the client's *context* to verify which recommendations this client already received and avoid sending redundant recommendations.

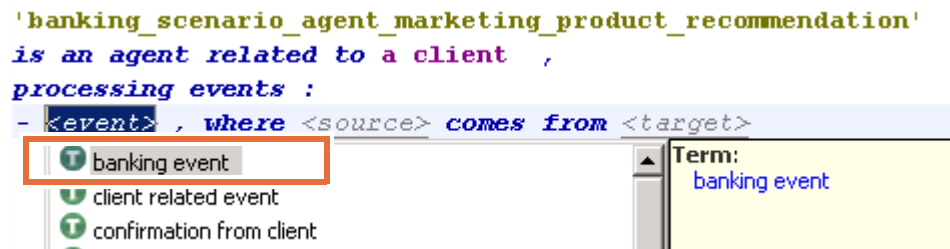
___ 1. Complete the descriptor to match this text:

'banking_scenario_agent_product_recommendation' is an agent related to a client ,
processing events :
- banking event , where this client comes from the client of this banking event
- product recommendation , where this client comes from the client of this product recommendation

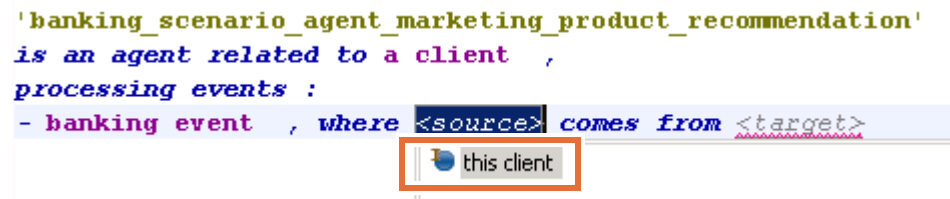
___ a. Click **entity** and select **a client**.



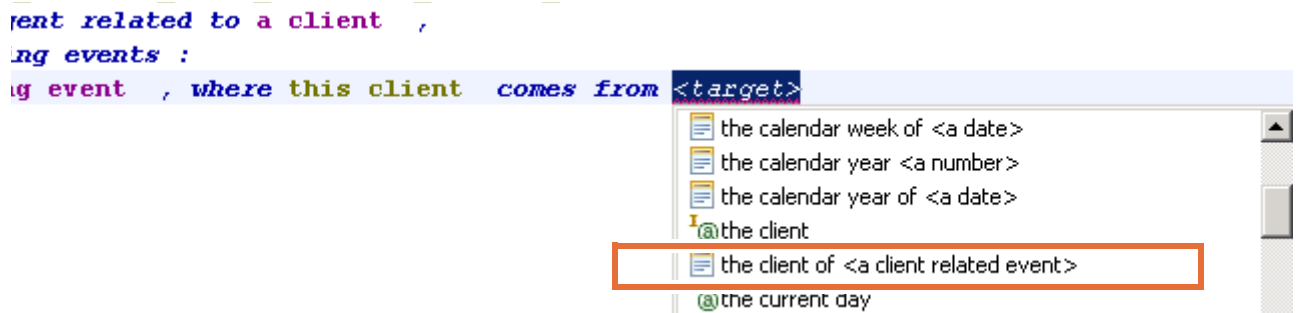
___ b. Click **event**, and select **banking event**.



___ c. Click **source**, and select **this client**.



___ d. Click **target**, and select **the client of <a client related event>**.



- ___ e. Click **a client related event**, and select **this banking event**.

comes from the client of *<a client related event>*

this banking event



Hint

You can format text in the editor by selecting all your text and pressing Ctrl+Shift+F.

- ___ 2. Save the `agent.adsc` file (Ctrl+S) and close it.

Section 3. Writing the San Francisco rule

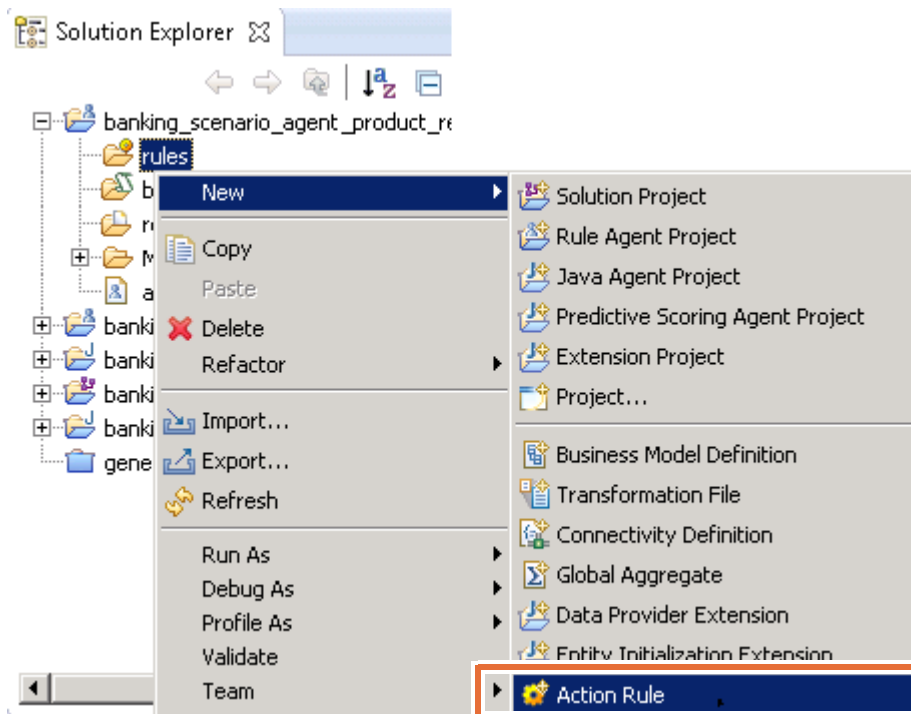


Requirements

The rule is triggered by an incoming event when clients access their accounts while they are in California. Your rule conditions also test that the client did not already receive this same recommendation recently.

The action statement emits an event that carries a product code (GUIDED TOUR SF) and a language code (the client's preferred language).

- ___ 1. Add the **Recommend guided tour of San Francisco** rule to your rule agent.
 - ___ a. Expand the **banking_scenario_agent_product_recommendation** project and right-click the **rules** folder.
 - ___ b. Click **New > Action Rule**.



- ___ c. In the Name field of the New Action Rule wizard, type: 1 Recommend guided tour of San Francisco
- ___ d. Click **Finish**.

The new rule opens in the rule editor.

__ 2. Define the rule to match the following text.

```
when a banking event occurs
  where the state of this banking event is CA
if
  there is no product recommendation
  where the code is GUIDED TOUR SF
  and this product recommendation is within 30 seconds before now ,
then
  emit a new product recommendation where
    the client is 'the client' ,
    the language is the preferred language of 'the client' ,
    the code is GUIDED TOUR SF ;
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder. To format the text in the rule editor, select all your text and press Ctrl+Shift+F.

__ 3. Take a moment to review this rule in comparison to the requirements to see how the rule implements the required tests and actions.

__ 4. Save the rule (Ctrl+S) and close the rule editor.



Note

For this lab, this rule checks for recommendations within the past 30 seconds so that you can experiment with rule behavior. In real life, a more plausible duration might be 30 days.

Section 4. Creating a Java agent



Requirements

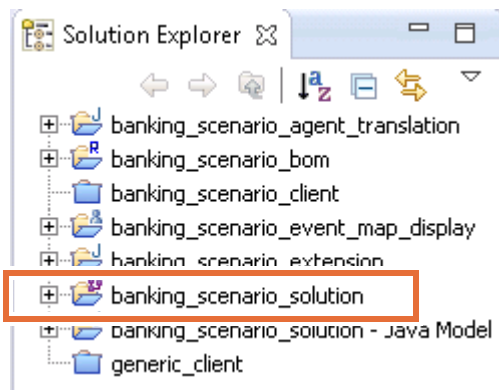
The product recommendation agent emits an event to send a message to the client's mobile device. However, Client entities have a preferred language attribute. To personalize the recommendations to use in the client's language, some post-processing is required before message can be sent.

Open the `BusinessModel.bmd` file to see that product recommendation event is a translatable event. You must create a Java agent to intercept product recommendation events and provide the translated message the client's language.

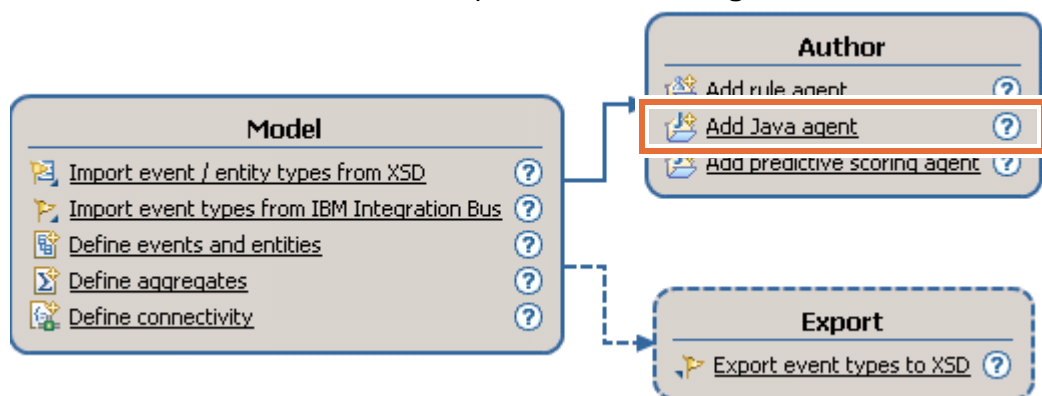
This multi-agent architecture enables separation of logic:

- Rule logic (what to show)
- Procedural logic (how to show it)

1. In Solution Explorer, click **banking_scenario_solution** to make sure that the **Solution Map** view is open.



2. In the **Author** task of the Solution Map, click **Add Java agent**.



3. Define the Java agent name.
 - a. In the **Project name** field, type: `banking_scenario_agent_translate_message`
 - b. In the **Agent Name** field, type: `TranslationAgent`

- __ c. Click **Finish**.

The `agent.adsc` file opens in the editor.

- __ 4. Define the agent descriptor to match the following text.

```
'banking_scenario_solution.banking_scenario_agent_translate_message.TranslationAgent' is an agent related to a client ,
processing events :
    - translatable event , where this client comes from the client of this
    translatable event
```



Hint

You can copy and paste this text from the `java-agent.xml` file in the **TrainingDir\code** folder. Use Notepad++ to open the file.

To format the text in the agent editor, select all your text and press Ctrl+Shift+F.

- __ 5. Save your work and close the `agent.adsc` file.
- __ 6. Complete the Java code for the agent.
- __ a. Expand the **banking_scenario_agent_translate_message > src > banking_scenario_solution > banking_scenario_agent_translate_message** folder to find the `TranslateAgent.java` file.
- __ b. Double-click the `TranslateAgent.java` file to open it in the editor.
- __ c. Make sure that the `java-agent.xml` file in the **TrainingDir\code** directory is open and copy the import statements from that file to the import section of the `TranslateAgent.java` file.
- __ d. Replace the “TODO” line with the code that is provided in the `java-agent.xml` file
- ```
public class TranslationAgent extends EntityAgent<Entity> {
 @Override
 public void process(Event event) throws AgentException {
 // TODO Add logic to handle the event
 }
}
```
- \_\_ e. Save your work to make sure that you do not have compilation errors.
- \_\_ f. Close the `java-agent.xml` file.
- \_\_ 7. Take some time to review the code and consider how this agent might be reused with other rules or agents for translation purposes.
- \_\_ 8. Close the Java editor.

**End of exercise**

## Exercise review and wrap-up

This exercise showed you how to create a rule agent and bind it to the client entity that you defined in the business model. You also wrote the business logic that detects client behaviors and emits a product recommendation event.



# Exercise 5. Writing and testing rules

## What this exercise is about

This exercise covers how to add a rule to an existing rule agent and deploy the solution for testing.

## What you should be able to do

After completing this exercise, you should be able to:

- Add a rule to a rule agent
- Deploy a solution
- Submit events through a test client to test rule behavior

## Introduction

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Adding the New York recommendation rule"
- Section 3, "Deploying the solution"
- Section 4, "Testing the solution"

## Requirements

This exercise requires that you use the **TrainingDir\workspace1-ny** workspace that is provided for this exercise.

## Section 1. Setting up your workspace

Before you start this exercise, make sure that you have Insight Designer open to the correct workspace and that you are in the Decision Insight perspective.



### Important

Whenever you switch workspaces, you must follow these steps:

- "Setting the target platform"
- "Importing the projects"
- "Verifying properties"
- "Verifying the Java build path"

### 1.1. Opening the workspace

- \_\_\_ 1. In Insight Designer, switch to the workspace that is provided for this exercise.
  - \_\_\_ a. From the **File** menu, click **Switch Workspace > Other**.
  - \_\_\_ b. When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspace1-ny** directory.

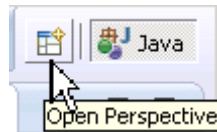


### Note

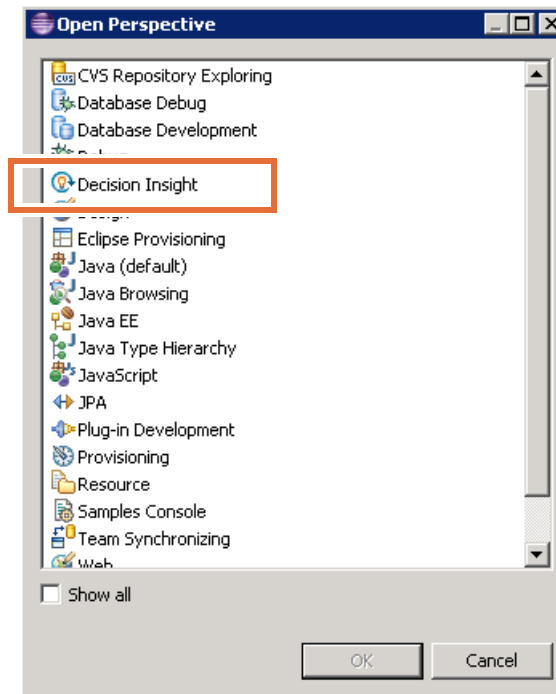
If you closed Insight Designer after the previous exercise, open it to the correct workspace.

- From the **Start** menu, click **All Programs > IBM > Decision Server Insights V8.7 > Insight Designer 8.7**.
- When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspace1-ny** folder.

- \_\_\_ 2. Close the **Welcome** view.
- \_\_\_ 3. Switch to the Decision Insight perspective.
  - \_\_\_ a. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.



- \_\_\_ 4. In the Open Perspective dialog box, select **Decision Insight**.



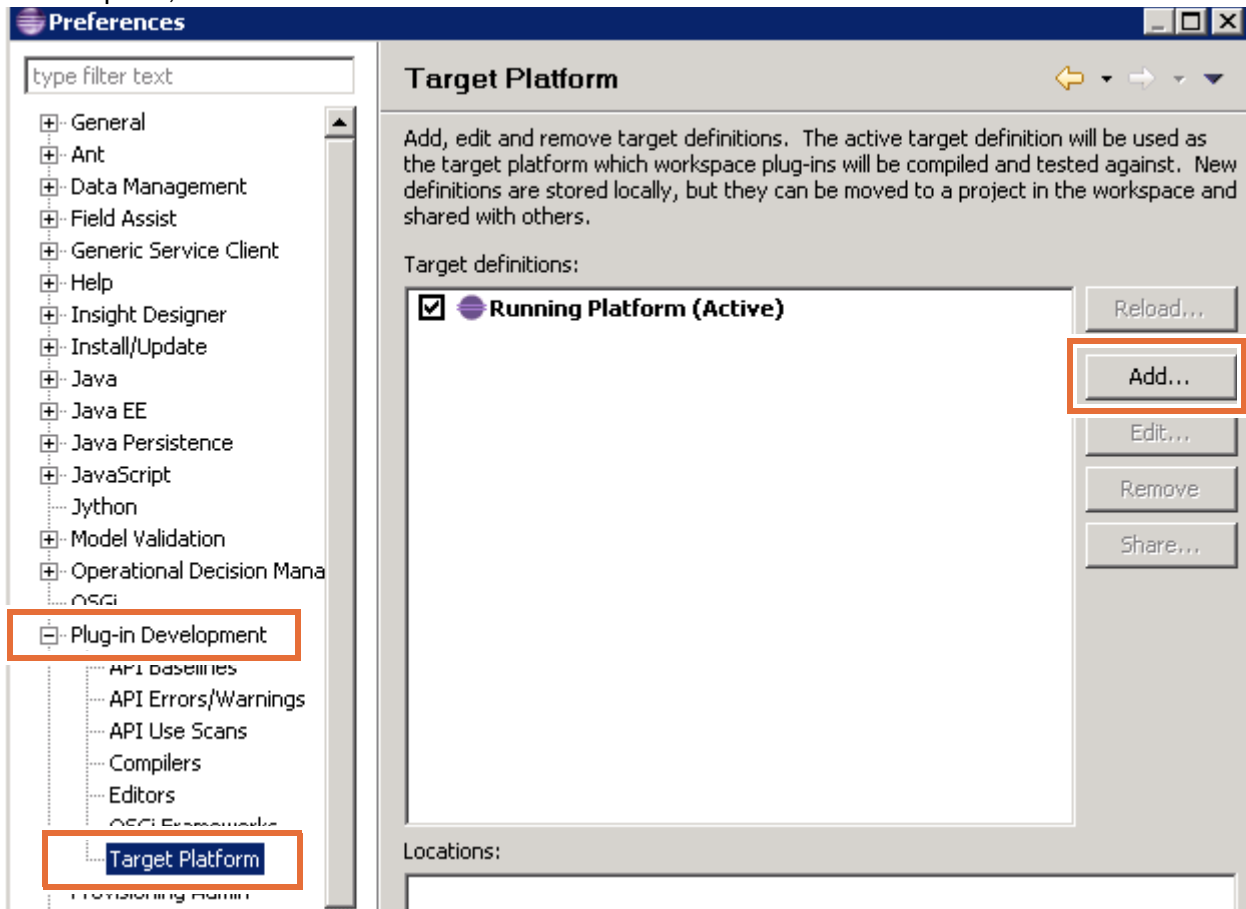
The Decision Insight perspective opens.

- \_\_\_ 5. Close the **Help** pane.

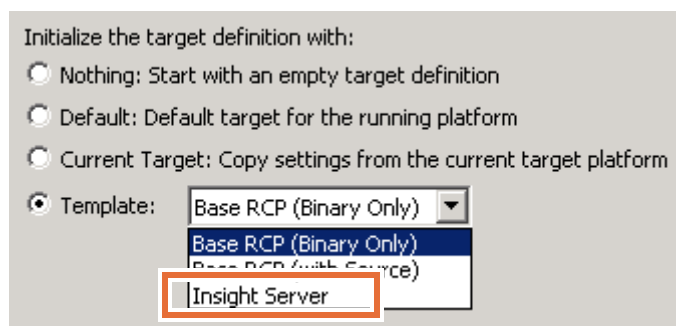
## 1.2. Setting the target platform

- \_\_\_ 1. From the **Window** menu, click **Preferences**.

- \_\_\_ 2. Expand **Plug-in Development**, click **Target Platform**, and when the Target Platform page opens, click **Add**.



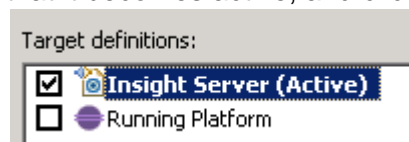
- \_\_\_ 3. On the Target Definition page, select **Template** and choose **Insight Server** from the list, click **Next**.



- \_\_\_ 4. Click **Finish**.

On the Target Platform page, Insight Server is now in the list of Target definitions.

- \_\_\_ 5. Select **Insight Server** so that it becomes active, and click **OK**.



The workspace is ready for you to start.

### 1.3. Importing the projects

- \_\_\_ 1. In the Decision Insight perspective, import the solution projects.
  - \_\_\_ a. From the **File** menu, click **Import**.
  - \_\_\_ b. In the Import wizard, click **General > Existing Projects into Workspace**, and click **Next**.
  - \_\_\_ c. Choose **Select root directory** and click **Browse**.
  - \_\_\_ d. Go to your workspace folder (for this exercise, **TrainingDir\workspace1-ny**) and click **OK**.
  - \_\_\_ e. Select all the projects.
  - \_\_\_ f. Clear the **Copy projects into workspace** check box.



- \_\_\_ g. Click **Finish**.

Your workspace now contains all the required projects. The projects might contain errors, but you resolve these errors during the next steps.



#### Troubleshooting

If you get a generation error window open, click **OK** to ignore it.

### 1.4. Verifying properties

Before you can run the Java client, you must first verify the project properties for the `testdriver.properties` file.

- \_\_\_ 1. In Solution Explorer, expand the **banking\_scenario\_client** project, and double-click **testdriver.properties** to open it in the editor.
- \_\_\_ 2. Verify the properties.
  - \_\_\_ a. If the value of the **trustStoreLocation** property does not match your product installation path, change the value to your product installation path and save your changes (Ctrl+S).
  - \_\_\_ b. Look for the `debugservers` property, and make sure that it is set to use port 6543.
 

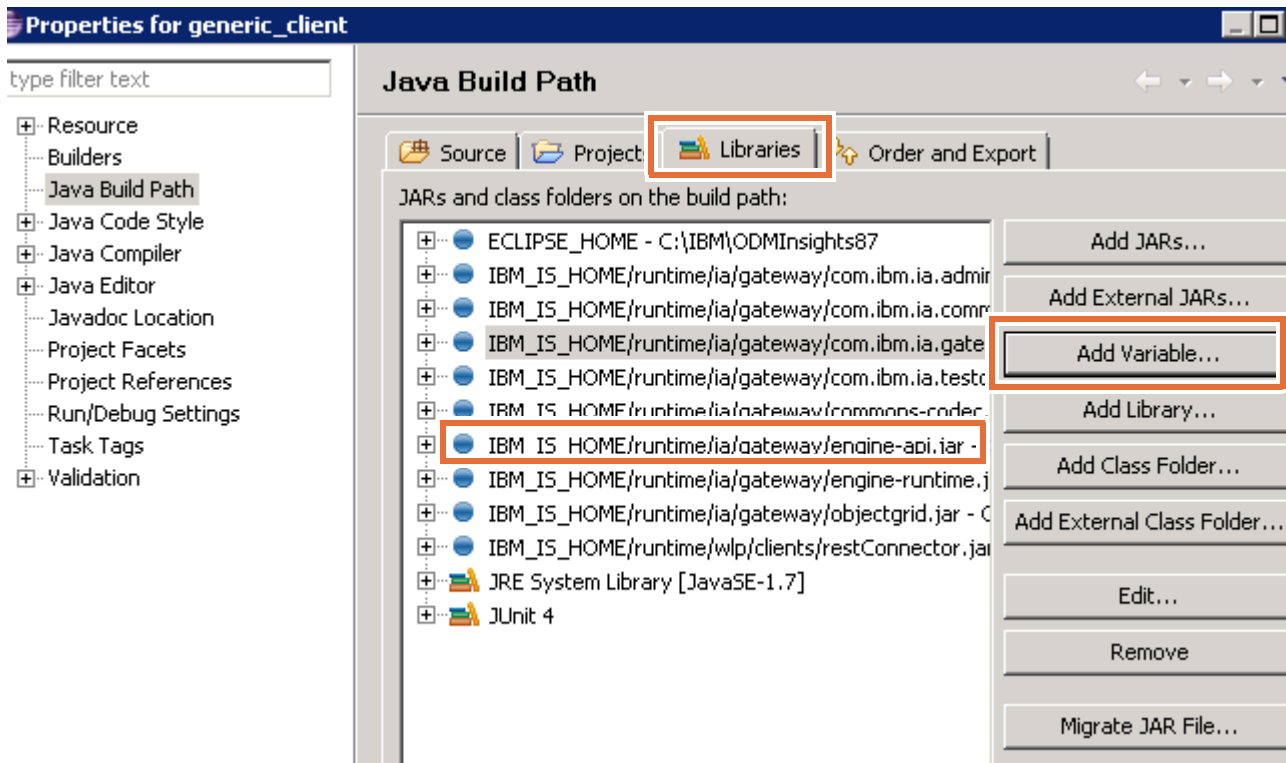
```
debugservers=localhost:6543
```
- \_\_\_ 3. Close the `testdriver.properties` file.

### 1.5. Verifying the Java build path

If you see errors on the Java client application projects, you might need to update the Java build path by following these steps.

- \_\_\_ 1. In Solution Explorer, right-click **generic\_client** and click **Properties**.

- \_\_\_ 2. In the Properties dialog box, click **Java Build Path** and open the **Libraries** tab.
- \_\_\_ 3. Select any one of the JAR files that starts with the **IBM\_IS\_HOME** variable and click **Add Variable**.



The New Variable Classpath Entry dialog box opens.

- \_\_\_ 4. Click **OK**.  
The **IBM\_IS\_HOME** variable is now added to point to your product installation path.
- \_\_\_ 5. Click **OK** to close the Properties dialog box and wait for the workspace to rebuild.  
You should not see any errors after the workspace is rebuilt.

## Section 2. Adding the New York recommendation rule

In this section, you learn how to create rule agents that process the business events that you defined in your business model.



### Requirements

You must add a new product recommendation for bank clients who are visiting the New York area and whose past banking activities indicate that they might have an interest in Broadway shows. The rule is triggered when clients access their accounts in a particular location and at a particular time.

The rule must:

- Test the location of the client entity
- Test the date to determine whether the show is scheduled
- Determine the client's interest in Broadway
- Emit an event that carries a product code (`BROADWAY SHOW`) and a language code (the client's preferred language)
- Cancel the recommendation if the client recently received a similar recommendation

A client's interest, or propensity, to buy Broadway tickets might be determined either through historical data analysis **or** predictive scoring. For this rule, this interest score is stored in the `propensity to buy BROADWAY SHOW TICKETS` property.

As with the San Francisco recommendation rule, this rule emits an event that is intercepted by the Java agent to send the message to the client's mobile device in the client's language.

### 2.1. Creating the New York rule

- \_\_\_ 1. In Solution Explorer, expand the **banking\_scenario\_agent\_product\_recommendation > rules** folder.

You should see the San Francisco rule that is listed in the project.

- \_\_\_ 2. Add the **Recommend Broadway show New York City** rule to the `banking_scenario_agent_product_recommendation` rule agent.
  - \_\_\_ a. Right-click the **rules** folder and click **New > Action rule**.
  - \_\_\_ b. In the **Name** field of the New Action Rule wizard, type: `Recommend Broadway show New York City`
  - \_\_\_ c. Click **Finish**.

The new rule opens in the rule editor.

\_\_ 3. Define the rule to match the following text.

```
when a banking event occurs
if
 all of the following conditions are true :
 - the propensity to buy BROADWAY_SHOW_TICKETS of 'the client' is at least
 0.8
 or the total amount of all purchases during the last period of 80 days ,
 where the type of good of each purchase is MUSICAL is at least 500

 - the city of this banking event is one of { New York , Newark }

 - today is after 12/1/2014

 - there is no product recommendation
 where the code is BROADWAY SHOW
 and this product recommendation is within 30 seconds before now ,
then
 emit a new product recommendation where
 the client is 'the client' ,
 the language is the preferred language of 'the client' ,
 the code is BROADWAY SHOW ;
```



#### Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and format the text by pressing Ctrl+Shift+F.

\_\_ 4. Take a moment to review this rule in comparison to the requirements to see how the rule implements the required tests and actions.



#### Note

For this lab, this rule checks for recommendations within the past 30 seconds so that you can experiment with rule behavior. In real life, a more plausible duration might be 30 days.

\_\_ 5. Save the rule (Ctrl+S) and close the rule editor window.



#### Important

After you write a rule, you should test to ensure that the rule works before you continue writing more rules. For this banking scenario, a test client is provided for you.

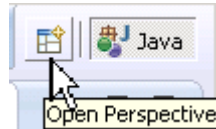


## Section 3. Deploying the solution

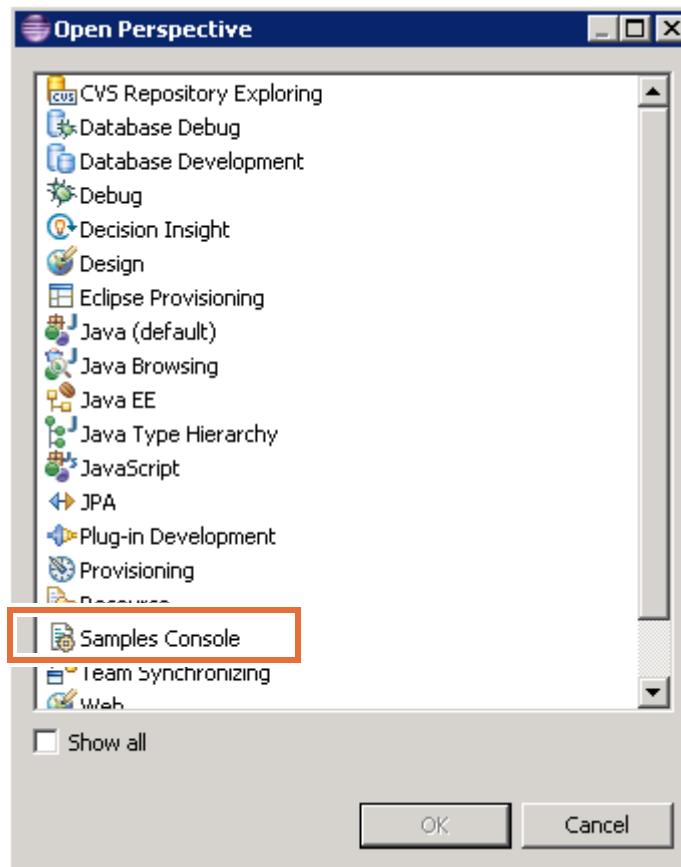
In this section, you deploy the solution and verify the deployment. If the sample server is not already started, you must start it before deployment.

### 3.1. Starting the server

- \_\_\_ 1. Make sure that the server is started.
  - \_\_\_ a. Switch to the Samples Console perspective.
  - \_\_\_ b. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.



- \_\_\_ c. In the Open Perspective dialog box, select **Samples Console**.

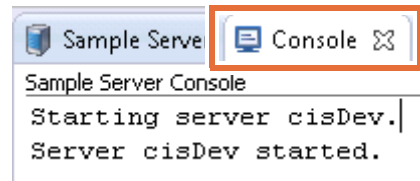


The Samples Console opens.

- \_\_\_ 2. In the **Sample Server** pane in the lower part of the workspace, click the **Start the sample server** icon to start the server.



The default server is called cisDev. The Console opens with the message that the server is started.

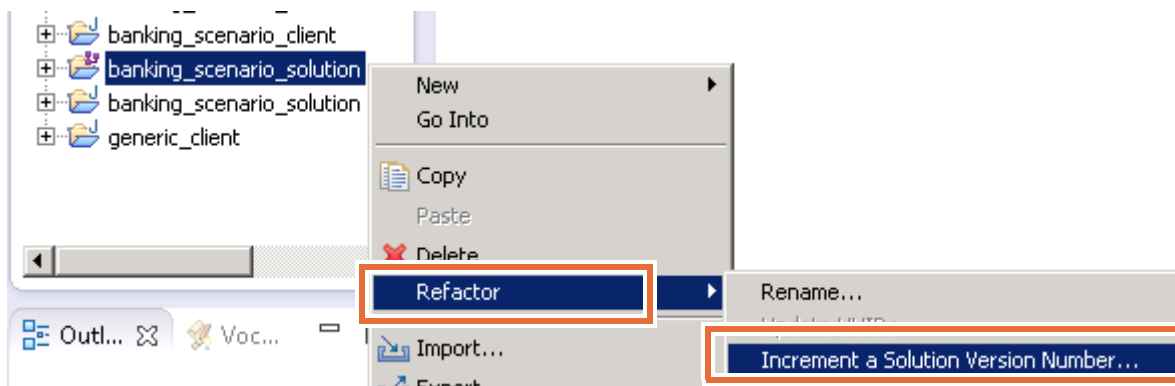


### Note

If the server is already started, the Console displays a message that the server is running.

## 3.2. Exporting the solution archive

- \_\_\_ 1. Switch back to the Decision Insight perspective (by clicking **Decision Insight** on the perspective toolbar).
- \_\_\_ 2. Increment the solution version number.
- \_\_\_ a. Right-click the **banking\_scenario\_solution** project and click **Refactor > Increment a Solution Version Number**.



- \_\_\_ b. Set **New solution version** to 1.0 and click **Finish**.

**New Solution Version Number**

### Increment a Solution Version Number

Increment the version number of a solution and all of its agents. The new version number is used when you export the solution.

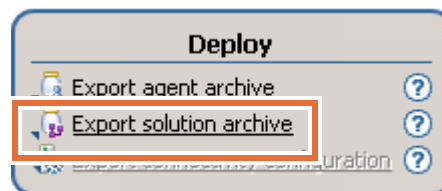
Solution project:

The current version in the solution manifest is 0.0.

New solution version:  .

- \_\_\_ 3. Export the solution archive.

- \_\_\_ a. In the Solution Map view, in the **Deploy** goal, click the **Export solution archive** link.

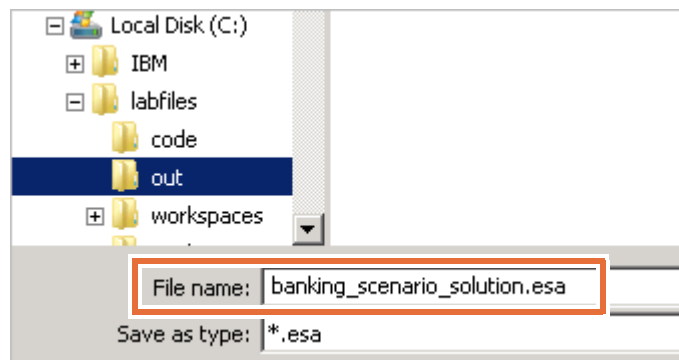


- \_\_\_ b. In the Solution Archive Export wizard, set the **Output file** field to:

TrainingDir\Solutions

where TrainingDir is the installation directory for the lab files. By default, TrainingDir is: C:\labfiles

- \_\_\_ c. Remove the version number from the file name and click **Save**.



**Note**

By default, when you select the output directory, the default archive name is provided with the version number the “.esa” file extension.

- \_\_ 4. Click **Save** and click **Finish**.
- \_\_ 5. Open the TraininDir\Solutions directory in Windows Explorer and you see your banking\_scenario\_solution.esa file.

### 3.3. Deploying automatically with Ant tasks

- \_\_ 1. Switch to the Java perspective.
- \_\_ 2. In the Package Explorer, expand the **banking\_scenario\_solution** folder to find the solution\_deploy.xml file.
- \_\_ 3. Double-click the solution\_deploy.xml file to open it and verify that the insights.home property is set correctly to the installation path for Decision Server Insights. The default path is: C:\IBM\ODMInsights87

```
<!-- #### EDIT THOSE PROPERTIES TO REFLECT YOUR ENVIRONMENT #### -->
<property name="insights.home" value="C:/IBM/ODMInsights87" />
<property name="solution.project" value="banking_scenario_solution" />
```

- \_\_ 4. If you modified the value, save your changes and close the editor.
- \_\_ 5. In Solution Explorer, right-click solution\_deploy.xml and click **Run As > Ant Build**.

**Warning**

There are two options for **Ant Build**. Choose the one without the ellipsis (...).

Make sure that you are in the Java perspective for this step so that the Ant Build task runs properly.

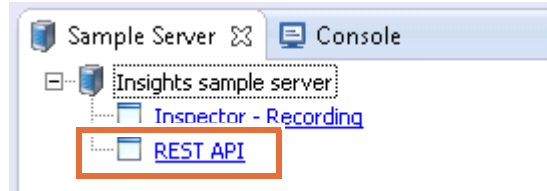
Deployment takes a few moments. After deployment is complete, you see a “Build successful” message in the console.

```
<terminated> banking_scenario_solution solution_deploy.xml [Ant Build] C:\IBM\ODMInsights87\jdk\bin\javaw.exe (Jan 10, 2015)
deploy.solution:
 [echo] Deploying ESA solution...
 [exec] Server configuration file successfully updated for server: cisDev
BUILD SUCCESSFUL
Total time: 22 seconds
```

### 3.4. Verifying deployment

To view the deployed solution, you can switch to the Samples Console perspective and use the REST API tool to verify that the solution was deployed to the application server.

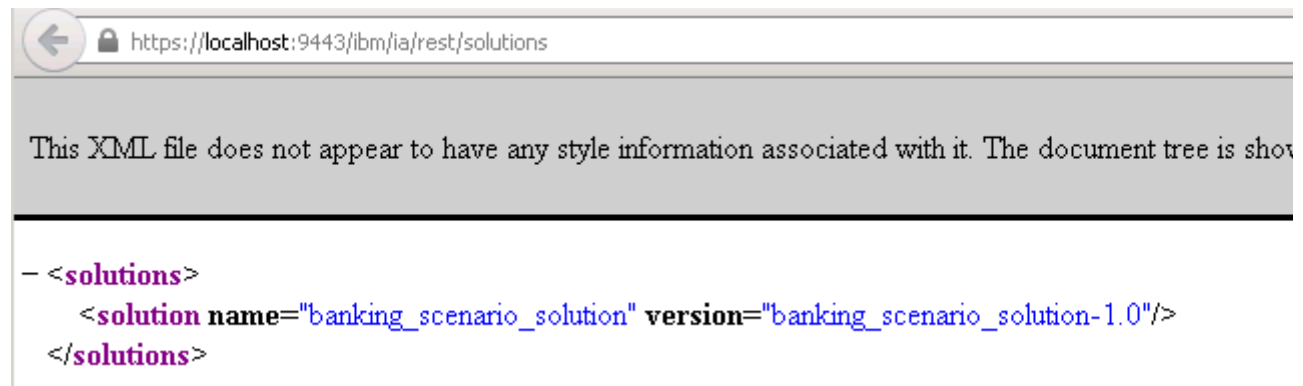
- \_\_\_ 1. Switch to the Samples Console perspective and open the **Sample Server** view.
- \_\_\_ 2. Click **REST API**.



- \_\_\_ 3. When the browser window opens, accept any security warnings and continue.
  - \_\_\_ a. In Mozilla, click **I understand the Risks** and click **Add Exception**.
  - \_\_\_ b. Click **Confirm Security Exception**.

The browser opens at the following URL and lists your solution:

`http://localhost:9080/ibm/ia/rest/solutions`



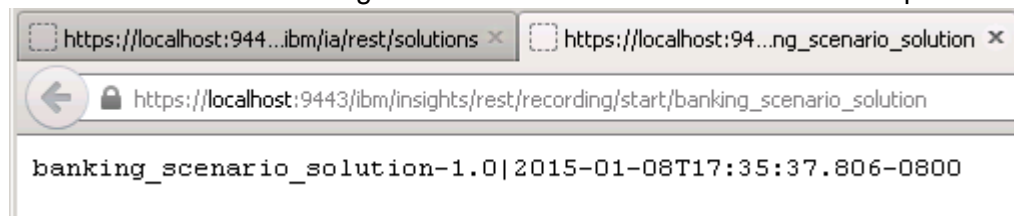
### 3.5. Preparing the recording of events in Insight Inspector

Before you submit test events, you can start the recording of event processing for your solution.

- \_\_\_ 1. Open a new tab or browser window and type this URL:

`https://localhost:9443/ibm/insights/rest/recording/start/banking_scenario_solution`

The browser returns a message with the solution name and a time stamp.



- \_\_\_ 2. Close or minimize the browser while you run tests in the next steps.

After you finish testing, you return to Insight Inspector to view the results.

## Section 4. Testing the solution

In this section, you test the solution.

- \_\_\_ 1. In Insight Designer, switch to the Java perspective.
- \_\_\_ 2. In the Package explorer, expand **banking\_scenario\_client > src > banking\_scenario\_client**, right-click **BankingScenarioClient.java**, and click **Run As > Java Application**.

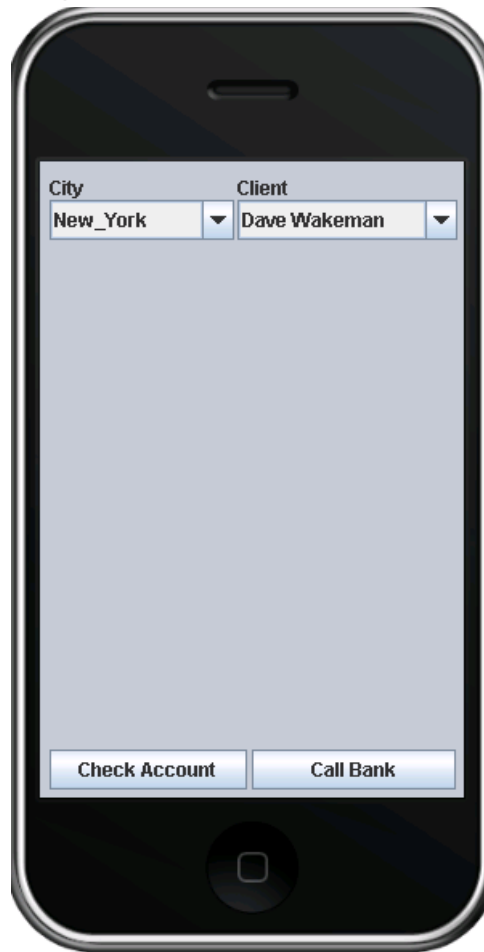
The Control Panel opens.



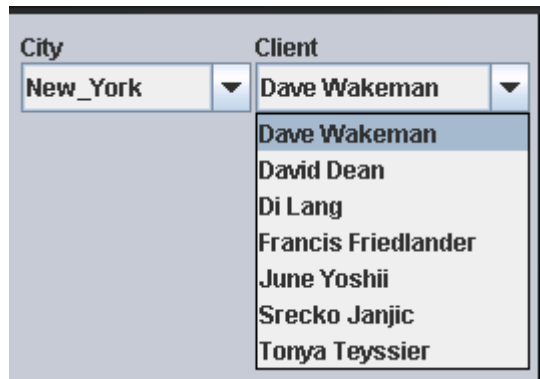
The Control Panel application is designed specifically for this lab. You can use it to launch pseudo-mobile devices and ATMs to interact with the solution.

- \_\_\_ 3. Click **LAUNCH MOBILE**.

A mobile device interface opens.



- \_\_\_ 4. Expand the **Client** list to see the list of clients that are available for this exercise.



Each client corresponds to an instance of the entity type Client as described in the `BusinessModel.bmd` file. These entities are created behind the scenes when you start the Control Panel.

- \_\_\_ 5. Choose **Tonya Teyssier**, make sure that the city is **New\_York** and click **Check Account**.

You should see the account balance and a message to recommend a Broadway show.

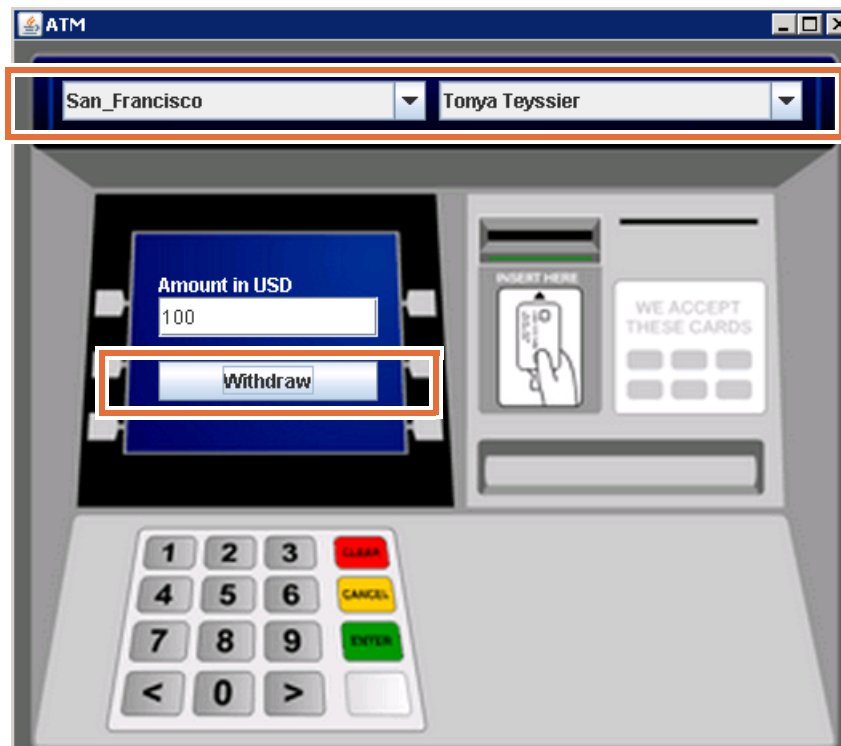


When Tonya checks her account, a `check_account` event is sent to the Insight Server run time.

The `check_account` event is a banking event that includes Tonya's ID. Thus, according to the statement in the `.adsc` file, it is sent to an instance of the `banking_scenario_agent_product_recommendation` rule agent that is bound to the Tonya entity. The event triggers the New York rule, which sends a product recommendation with the code `BROADWAY SHOW`.

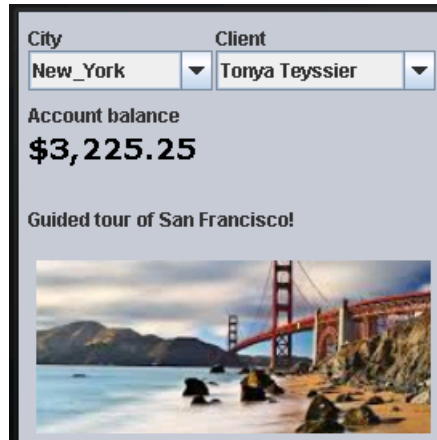
As explained earlier, a Java agent intercepts that product recommendation to check the language, then, emits a message that is displayed on the mobile device.

- \_\_\_ 6. Keep the mobile interface open and submit a withdrawal event from a different location.
  - \_\_\_ a. Go to the Control Panel and click **LAUNCH ATM**.
  - \_\_\_ b. Select **San Francisco** and **Tonya Teyssier**, and click **Withdraw**.





Now, Tonya sees a recommendation (in her language, English) on her mobile phone for a guided tour of San Francisco.



This recommendation is the result of the rule `Recommend guided tour of San Francisco`. By using the withdrawal operation, Tonya's bank detects that she is in San Francisco.

- \_\_\_ 7. Test the business logic for not resending notifications to clients who already received a particular product recommendation.
  - \_\_\_ a. Switch the ATM location to **New\_York** and submit a withdraw event.  
You should see the Broadway message on the mobile interface because 30 seconds elapsed since the Broadway show was initially recommended.
  - \_\_\_ b. Switch back to **San\_Francisco** and submit another withdraw event.  
You should see the San Francisco message on the mobile interface.
  - \_\_\_ c. Switch back to New York and submit another withdraw event.  
If you run the sequence New York-San Francisco-New York quickly enough, you should not see the Broadway recommendation again.

This logic is intended to avoid overwhelming the clients with the same recommendations and corresponds to the following condition in the New York rule:

  - there is no product recommendation  
  where the code is BROADWAY SHOW  
  and this product recommendation is within 30 seconds before now ,
- \_\_\_ 8. Test the preferred language by switching clients.
  - \_\_\_ a. In both the mobile device and the ATM, change **Client** to **Francis Friedlander** and redo the same sequence.
    - In the ATM interface, the city set to **San\_Francisco**, click **Withdraw**.
  - \_\_\_ b. In both the mobile device and the ATM, change **Client** to **Di Lang**, and redo the same sequence.
    - In the mobile interface, with the city set to **New\_York**, click **Check Account**.
    - In the ATM interface, with the city set to **San\_Francisco**, click **Withdraw**.

You should now see messages in Chinese.



Rules in the rule agent `banking_scenario_agent_product_recommendation` emit a language-neutral product recommendation event with a product code. This event is captured by the Java agent `banking_scenario_agent_translation`, and is transformed into `DisplayableMessage` with text in the client's preferred language and a corresponding graphic for the recommended product.

- \_\_\_ 9. Take some time to review the agent descriptor of the `banking_scenario_agent_translation` agent. You can also check the procedural logic of the agent in the `TranslationAgent.java` file.
- \_\_\_ 10. Next, review what is happening by starting the Generic Client that is provided with this lab.

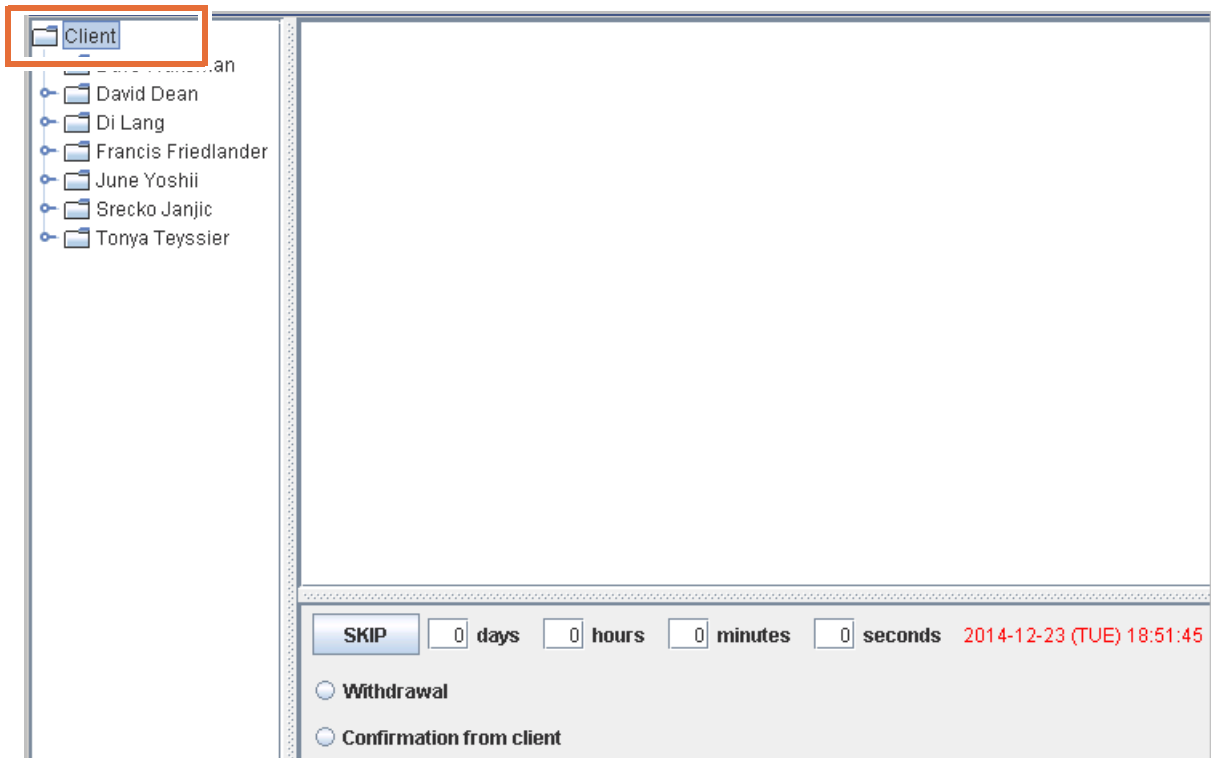


### Information

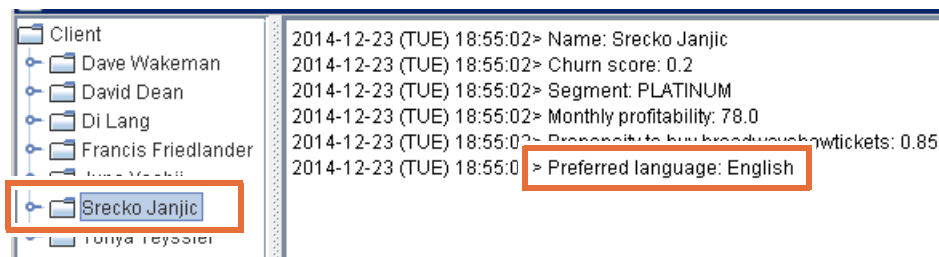
With the Generic Client, you can inspect the entities, generate events, and display the events that are generated by the Decision Server Insights run time. The Generic Client does not display the events that you emit from the ATM and from the mobile device.

- \_\_\_ a. In the Control Panel, select **LAUNCH GENERIC CLIENT**.

- \_\_\_ b. When the window opens, double-click **Client** to expand the entity graph.

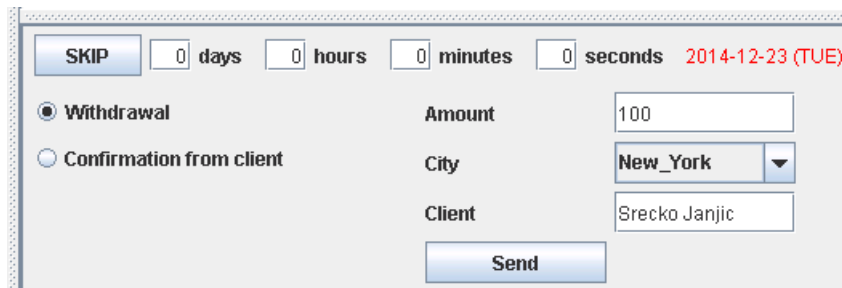


- \_\_\_ c. In the entity graph, click **Srecko Janjic** to see the attributes of this entity.



For example, notice that Srecko's preferred language value is English, so messages for him are displayed in English.

You can use the bottom pane to generate events as an alternative to using the mobile device and ATM interfaces.



**Note**

To change the client, you select **Withdrawal** and manually type in the **Client** field the name as it is spelled in the entity graph.

You can use the upper pane to visualize the entity attributes, the inbound events that you submit, and the outbound events that are emitted by the rules.

- Attribute values of the entity you are selecting on the left pane, for example:

```
2014-12-22 (MON) 17:06:20> Name: Srecko Janjic
2014-12-22 (MON) 17:06:20> Churn score: 0.9
2014-12-22 (MON) 17:06:20> Segment: PLATINUM
2014-12-22 (MON) 17:06:20> Monthly profitability: 180.0
2014-12-22 (MON) 17:06:20> Propensity to buy broadwayshowtickets: 0.85
2014-12-22 (MON) 17:06:20> Preferred language: English
```

- Events that you are sending by using the lower-right pane, for example:

```
2014-12-22 (MON) 17:06:42> SND #1: Withdrawal - Srecko Janjic - Location:
New_York POINT [-74.00590000000000,40.71270000000000] - Amount=100
```

- Events that the rules are emitting and the Generic Client is receiving. You can see in particular the generated instances of `DisplayableMessage`:

```
2014-12-22 (MON) 17:06:43> REC: Displayable message - Srecko Janjic - See
Broadway show tonight! - English
```

\_\_\_ 11. After you finish testing, click **EXIT** on the Control Panel to close all the interface windows.

## 4.1. Visualizing the event processing in Insight Inspector

Now that you submitted events to the run time, you can visualize that activity by using Insight Inspector.

\_\_\_ 1. To stop recording, open a browser and type this URL:

```
http://localhost:9080/ibm/insights/rest/recording/stop/banking_scenario_solu
tion
```

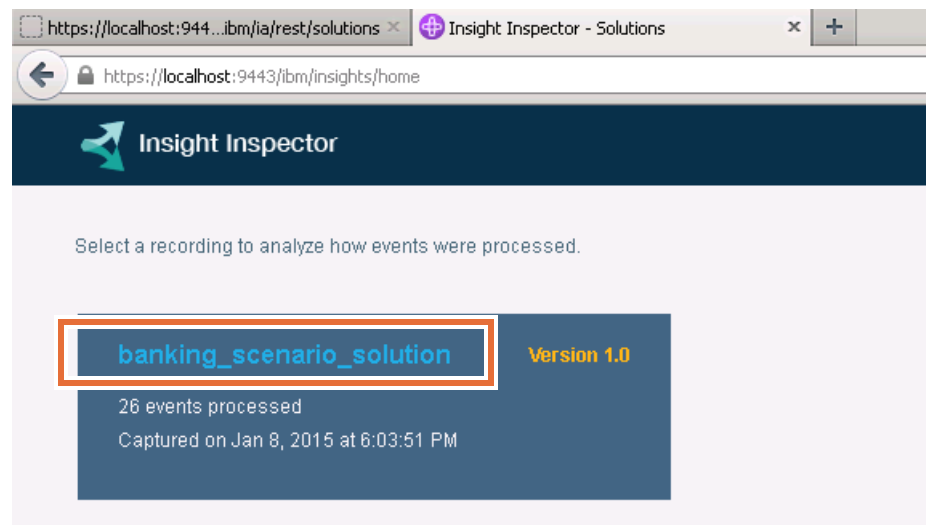
The browser returns a message that the recording stopped. You can now open Insight Inspector so see recorded events for the banking solution.

\_\_\_ 2. In your browser, open Insight Inspector by typing the following URL:

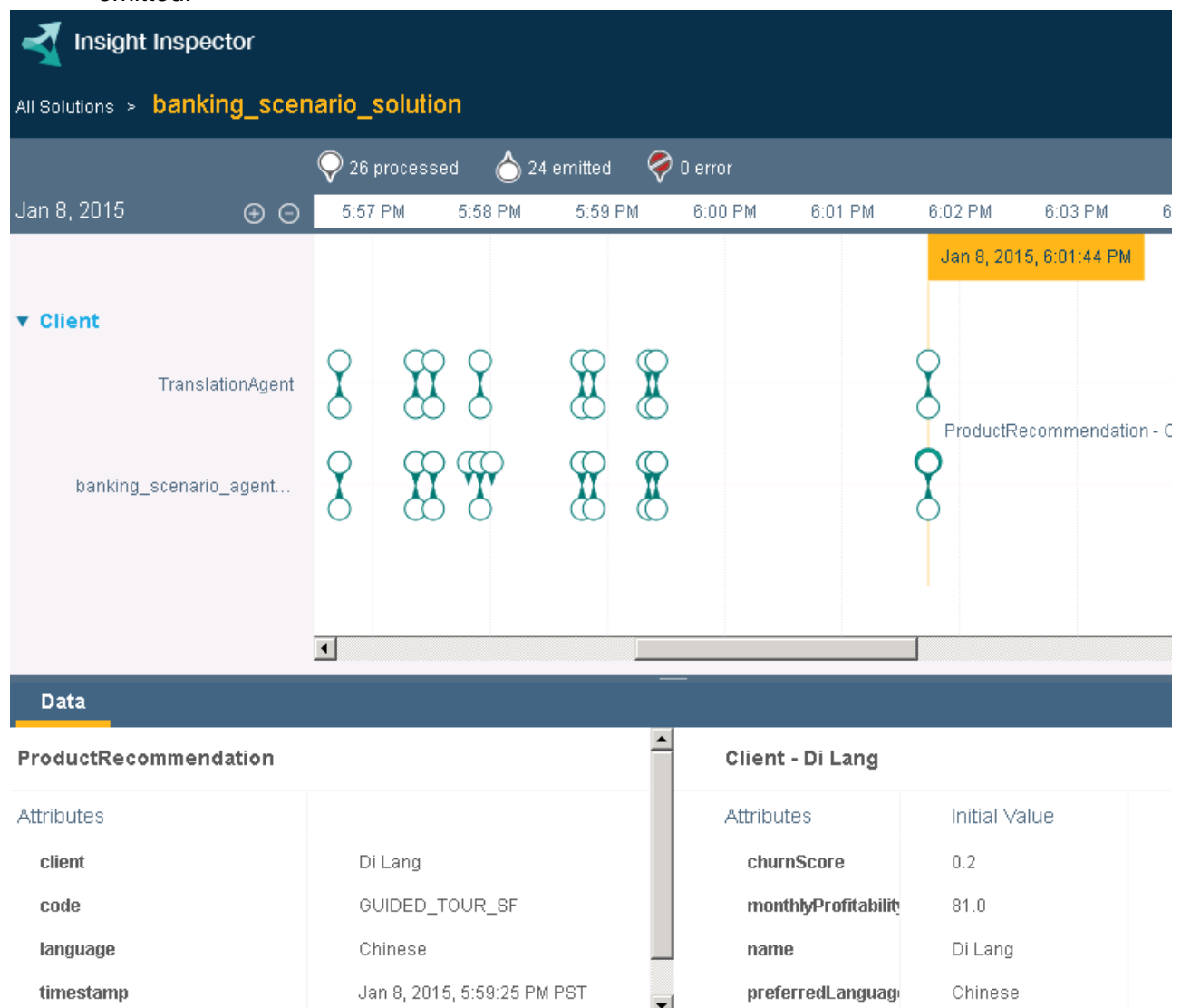
```
http://localhost:9080/ibm/insights
```

Your banking solution is listed on the home page.

\_\_\_ 3. View the events by clicking **banking\_scenario\_solution**.



On the top pane of Insight Inspector, you see the number of events that were processed and emitted.



Insight Inspector shows a timeline with events marked. When you click an event marker, you can review details about that event in the Data view of Insight Inspector.



#### Hint

To zoom in and see events that occurred before the end of the recording, click the plus (+) icon. To zoom out, click the minus (-) icon. You can also click and drag the timeline to the left and to the right to move to the beginning of the recording, or back to the final event.

\_\_\_ 4. Close the browser with Insight Inspector.

### End of exercise

## Exercise review and wrap-up

This exercise showed you how to create a rule agent and bind it to the client entity that you defined in the business model. You also wrote the business logic that detects client behaviors and emits a product recommendation event.





# Exercise 6. Using global aggregates in rules

## What this exercise is about

This exercise shows you how to create global aggregates and use them in your rules to identify and respond to outliers.

## What you should be able to do

After completing this exercise, you should be able to:

- Create a global aggregate
- Use global aggregates in rules
- Use the REST API to view aggregates in your solution

## Introduction

In this exercise, you create a churn management rule agent that turns insight into action by encouraging loyalty when it detects client propensity to churn (or defect).

The decision to allocate a gift to the client is based on the comparison of the client's predictive churn score against a global entity aggregate that measures the average churn score of a population of clients.

You first create the **global aggregates**, then you create the rule agent and rule that tests the client against the aggregates. To respond proactively to churn patterns, the rule emits a loyalty gift event. This event is handled by a translation agent, which translates that event into a message in the client's preferred language. Finally, the client receives that message through a mobile app that runs on the client's mobile device.

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Creating the global aggregates"
- Section 3, "Creating a churn prevention rule agent"
- Section 4, "Deploying the solution"
- Section 5, "Testing the solution"

## Requirements

This exercise requires that you complete the previous exercise before continuing or that you use the **TrainingDir\workspace2-agg** workspace.

## Section 1. Setting up your workspace

In Insight Designer, you can continue working in your current workspace or switch to the workspace that is provided for this exercise and includes the solution to the previous exercise.



### Important

If you choose to switch workspaces, you must follow these steps before you continue:

- "Setting the target platform"
- "Importing the projects"
- "Setting the target platform"
- "Verifying the Java build path"

- \_\_\_ 1. To switch workspaces:
  - \_\_\_ a. From the **File** menu, click **Switch Workspace > Other**.
  - \_\_\_ b. When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspace2-agg** folder.
- \_\_\_ 2. Follow the steps in Section 1, "Setting up your workspace" from Exercise 5, "Writing and testing rules" to complete the workspace setup.

## Section 2. Creating the global aggregates

In this section, you first create the global aggregates, which are stored with the business model.



### Information

A global aggregate measures the overall trend across a population of entities. Understanding the trend can help you identify entity behaviors that fall outside the expected pattern.

Increased awareness of individual client behavior helps the bank recognize certain patterns, such as when a client might be considering switching to another bank. Before a client leaves, the bank wants to respond proactively by offering incentives that show personal interest to retain the client.

For this scenario, you need to measure potential churn patterns across a subset of clients. You create two global aggregates that consolidate attribute values across all GOLD and PLATINUM clients to measure these factors:

- Average churn scores
- Monthly profitability

By finding the average for the population, you can then identify outliers. In this case, you want to identify your most profitable clients with the highest risk of churning.

### 2.1. Creating the Average churn for PLATINUM and GOLD aggregate

- \_\_\_ 1. If you are not in the Decision Insight perspective of Insight Designer, switch to it now.
- \_\_\_ 2. Create the aggregate definition file.
  - \_\_\_ a. Click **File > New > Global Aggregate**, and then click **Next**.
  - \_\_\_ b. In the **BOM project** field, click **Browse**, select **banking\_scenario\_bom**, and click **OK**.
  - \_\_\_ c. In the **Name** field, type: Average churn for PLATINUM and GOLD
  - \_\_\_ d. Click **Finish**.

The Average churn for PLATINUM and GOLD.agg file opens in the editor.



This global entity aggregate must store the average churn score for all clients whose segment attribute is set to Gold or Platinum. In the aggregate definition, you must specify:

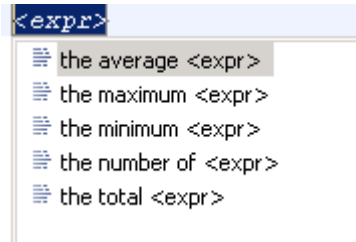
Aggregate name	Average churn for PLATINUM and GOLD
Aggregation operator	Average churn score
Attribute that is used to identify a subset of entities from the whole population	the segment is one of {PLATINUM , GOLD}
Evaluation schedule	intervals of 30 seconds

- \_\_\_ 3. Define the aggregate.
- \_\_\_ a. In the editor, click **<expr>**.



## Questions

Which of the expressions in the list do you choose?



How do you write the definition to test the whole population of clients to find the Gold and Platinum segment?

How do you set the evaluation interval?

- \_\_\_ b. Complete the definition to match the following text:
- ```
define 'Average churn for PLATINUM and GOLD' as the average churn score of
all clients ,
    where the segment is one of { PLATINUM , GOLD } ,
    evaluated at intervals of 30 seconds
```
- ___ c. Save your work and close the Average churn for PLATINUM and GOLD.agg file.

2.2. Creating the Average monthly profitability for PLATINUM and GOLD aggregate

- ___ 1. Create the aggregate definition file.
- ___ a. Click **File > New > Global Aggregate**, and then click **Next**.
- ___ b. In the **BOM project** field, click **Browse**, select **banking_scenario_bom**, and click **OK**.
- ___ c. In the **Name** field, type: Average monthly profitability for PLATINUM and GOLD
- ___ d. Click **Finish**.

This global entity aggregate must store the average monthly profitability for all clients whose segment attribute is set to Gold or Platinum. In the aggregate definition, you must specify:

| | |
|---|---|
| Aggregate name | Average monthly profitability for PLATINUM and GOLD |
| Aggregation operator | average monthly profitability |
| Attribute that is used to identify a subset of entities from the whole population | the segment is one of {PLATINUM , GOLD} |
| Evaluation schedule | intervals of 30 seconds |

- __ 2. Define the aggregate.
- __ a. In the editor, click **<expr>**.



Questions

Can you define the aggregate with the information that is provided in the table?

- __ b. Try to define the aggregate by using the prompts in the editor.
- __ c. After you complete your definition, check your work with the definition that is provided here.

```
define 'Average monthly profitability for PLATINUM and GOLD' as the
average monthly profitability of all clients ,
    where the segment is one of { PLATINUM , GOLD } ,
    evaluated at intervals of 30 seconds
```

- __ d. Save your work and close the `Average monthly profitability for PLATINUM and GOLD.agg` file.



Important

You can view a summary of the global aggregates in the `globalQueries.var` variable set file, which is in the `aggregates` folder of the BOM project. Do not change the content of the file because your changes to it are not saved.



Note

For this lab, the interval between aggregate computations is set to 30 seconds. In real life, global aggregates would be typically recalculated every day.

Section 3. Creating a churn prevention rule agent

In this section, you create a churn management rule agent. This agent turns into action the insight about the client's propensity to churn, and how this propensity compares to the average propensity in the GOLD + PLATINUM population to which the client belongs. You can use this information to allocate limited resources (such as a gift) to the most important cases.

3.1. Creating the rule agent

In this step, you create a rule agent that is bound to a client entity and subscribes to banking events that indicate whether the client might defect.

- __ 1. Create a rule agent named `banking_scenario_agent_churn_prevention`.
 - __ a. In the **Author** task of the Solution Map, click **Add rule agent**.
 - __ b. In the **Project name** field, type: `banking_scenario_agent_churn_prevention`
 - __ c. Click **Finish**.

The `agent.adsc` file opens in the editor.

- __ 2. Complete the agent to match this text:

```
'banking_scenario_agent_churn_prevention' is an agent related to a client ,  
processing events :  
    - product recommendation , where this client comes from the client of  
    this product recommendation
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and press Ctrl+Shift+F to format the text.

- __ 3. Save the `agent.adsc` file and close it.

3.2. Creating a rule that offers a client reward

This rule is triggered when a Broadway show is recommended to a client that is identified by testing against the global aggregates to meet these criteria:

- The client is among the most profitable of all GOLD and PLATINUM clients
- The client is among that GOLD and PLATINUM clients most likely to defect

- __ 1. Add the **Reward churn client New York** rule to your rule agent.
 - __ a. Expand the `banking_scenario_agent_churn_prevention` project, right-click **rules**, and click **New > Action rule**.
 - __ b. In the **Name** field of the New Action Rule wizard, type `Reward churn client New York` and click **Finish**.
- __ 2. Define the rule.

```
when a product recommendation occurs
  where the code of this product recommendation is BROADWAY SHOW
definitions
  set AVG_CHURN to 'Average churn for PLATINUM and GOLD' ;
  set AVG_PROFI to 'Average monthly profitability for PLATINUM and GOLD' ;
if
  the segment of 'the client' is one of { PLATINUM , GOLD }
  and the churn score of 'the client' is at least 1.5 * AVG_CHURN
  and the monthly profitability of 'the client' is at least 1.3 * AVG_PROFI
then
  emit a new gift where
    the client is 'the client' ,
    the language is the preferred language of 'the client' ,
    the code is FREE DINNER ON BROADWAY ;
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and press Ctrl+Shift+F to format the text.

- __ 3. Save the rule (Ctrl+S).



Important

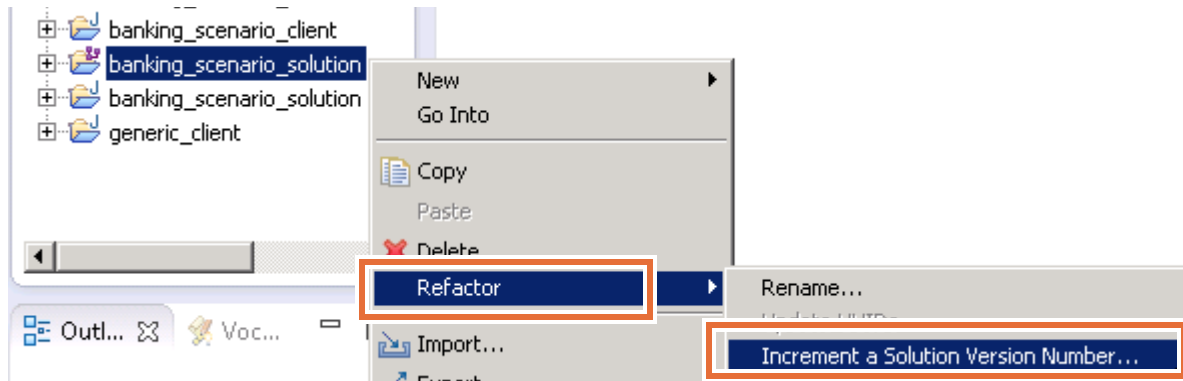
After you create a rule, you should test to ensure that the rule works before you continue writing more rules. For this banking scenario, a test client is provided for you.

Section 4. Deploying the solution

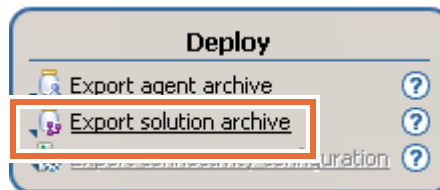
In this section, you deploy the solution and verify the deployment.

4.1. Exporting the solution

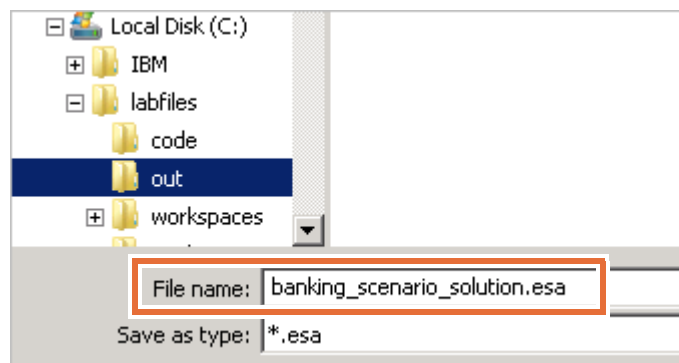
- ___ 1. Before you export the updated solution, increment the solution version number.
 - ___ a. Right-click the **banking_scenario_solution** project and click **Refactor > Increment a Solution Version Number**.



- ___ b. Set **New solution version** to 2.0 and click **Finish**.
- ___ 2. Export the solution archive.
 - ___ a. In the Solution Map view, in the **Deploy** goal, click the **Export solution archive** link.



- ___ b. In the Solution Archive Export wizard, set the **Output file** field to:
 TrainingDir\Solutions
 where TrainingDir is the installation directory for the lab files. By default,
 TrainingDir is: C:\labfiles
- ___ c. Remove the version number from the file name and click **Save**.



**Note**

By default, when you select the output directory, the default archive name is provided with the version number and the “.esa” file extension.

- __ 3. Click **Save** and click **Finish**.
- __ 4. When prompted to overwrite the existing file, click **Yes**.

4.2. Deploying automatically with Ant tasks

- __ 1. Make sure that you are in the Java perspective.
- __ 2. In the Package Explorer, expand the **banking_scenario_solution** folder to find the **solution_deploy.xml** file.
- __ 3. Deploy the solution by right-clicking **solution_deploy.xml** and click **Run As > Ant Build**.

**Note**

There are two options for **Ant Build**. Choose the one without the ellipsis (...).

Deployment takes a few moments. After deployment is complete, you see a “Build successful” message in the console.

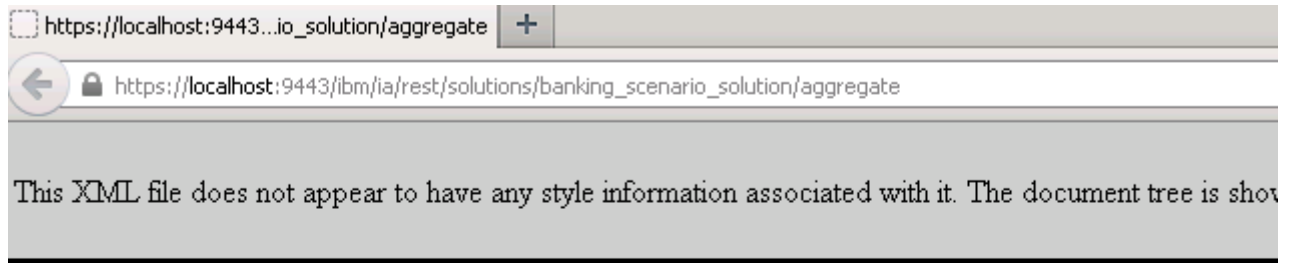
4.3. Accessing your aggregates through REST

To view the deployed aggregates, you can use REST to access them in a browser.

- __ 1. Open a browser and type this URL:
`http://localhost:9080/ibm/ia/rest/solutions/banking_scenario_solution/aggregate`
- __ 2. When the browser window opens, accept any security warnings and continue.

The browser opens at the following URL and lists the aggregates that are defined for your solution:

`http://localhost:9080/ibm/ia/rest/solutions/banking_scenario_solution/aggregate`



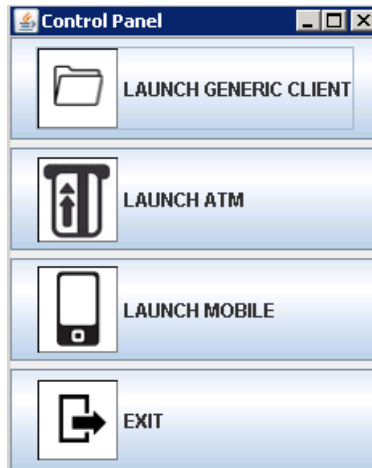
```
- <object type="Collection[aggregate]">
  - <aggregates>
    - <collection>
      - <object type="aggregate">
        - <attribute name="id">
          - <string>
            defvarAveragex95churnx95forx95PLATINUMx95andx95GOLD
          </string>
        </attribute>
        - <attribute name="value">
          <double>0.5</double>
        </attribute>
      </object>
      - <object type="aggregate">
        - <attribute name="id">
          - <string>
            defvarAveragex95monthlyx95profitabilityx95forx95PLATINUMx95andx95GOLD
          </string>
        </attribute>
        - <attribute name="value">
          <double>123.85714285714286</double>
        </attribute>
      </object>
    </collection>
  </aggregates>
```

Section 5. Testing the solution

In this section, you test the solution.

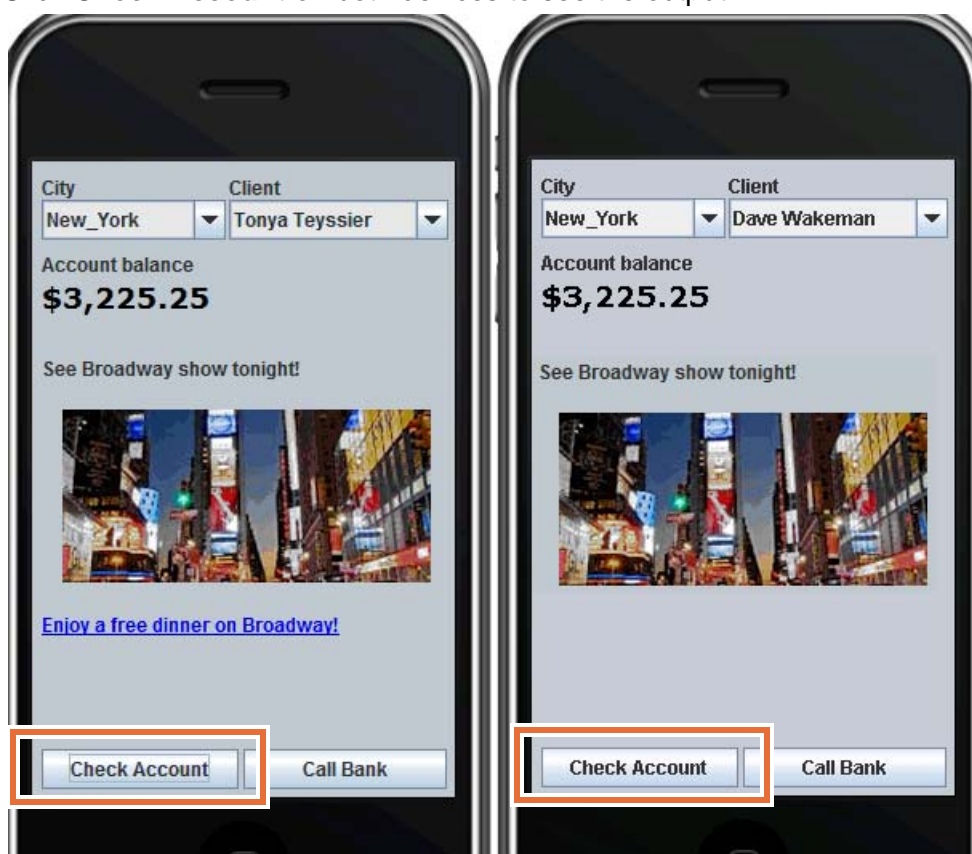
- ___ 1. Make sure that you are in the Java perspective.
- ___ 2. In the Package explorer, expand **banking_scenario_client > src > banking_scenario_client**, right-click **BankingScenarioClient.java**, and click **Run As > Java Application**.

The Control Panel opens.



- ___ 3. Click **LAUNCH MOBILE** twice to test with two mobile device interfaces.
 - ___ a. Place them side-by-side on your screen.
 - ___ b. In the **Client** list, choose **Tonya Teyssier** for the first mobile, and **Dave Wakeman** for the second.

- ___ c. Click **Check Account** on both devices to see the output.



- ___ 4. From the Control Panel, click **LAUNCH GENERIC CLIENT**.
- ___ 5. Click each of the client entities in the side pane to check the attribute values for the GOLD and PLATINUM population of clients.

For example, you should see these values:

| Client | Churn score | Profitability |
|--|-------------|---------------|
| Dave Wakeman | 0.20 | 80.00 |
| David Dean | 0.20 | 90.00 |
| Di Lang | 0.20 | 81.00 |
| Francis Friedlander | 0.20 | 80.00 |
| June Yoshii | 0.90 | 180.00 |
| Srecko Janjic | 0.90 | 180.00 |
| Tonya Teyssier | 0.90 | 179.00 |
| Average / Global Aggregate | 0.50 | 124.29 |
| Factor | 1.50 | 1.30 |
| Rule threshold = Factor x Average | 0.75 | 161.57 |

Notice the values for June, Srecko, and Tonya, which explains why they receive a gift. Note also that the averages that are calculated match the values that are displayed when you inspect the global aggregates with the REST API.

- ___ 6. Open the Reward churn client New York rule to understand the rule behavior in relation to the entity values.

___ 7. After you finish testing, close the rule, and click **EXIT** on the Control Panel.

End of exercise

Exercise review and wrap-up

The exercise demonstrated how you can use global aggregates to identify outliers and take specific actions when behavior patterns are detected across a population of entities in your domain.

Exercise 7. Using event aggregates in rules

What this exercise is about

This exercise shows you how to use event aggregates to analyze a current transaction in comparison to historical transactions.

What you should be able to do

After completing this exercise, you should be able to:

- Use event aggregates and aggregate functions in rules

Introduction

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Creating the fraud detection rule agent"
- Section 3, "Deploying the solution"
- Section 4, "Testing the solution"

Requirements

This exercise requires that you use the **TrainingDir\workspace3-fraud** workspace.

Section 1. Setting up your workspace

In Insight Designer, you switch to the workspace that is provided for this exercise, which includes the solution to the previous exercise.

- ___ 1. To switch workspaces:
 - ___ a. From the **File** menu, click **Switch Workspace > Other**.
 - ___ b. When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspace3-fraud** folder.
- ___ 2. Follow the steps in Section 1, "Setting up your workspace" from Exercise 5, "Writing and testing rules" to complete these workspace setup steps:
 - "Setting the target platform"
 - "Importing the projects"
 - "Setting the target platform"
 - "Verifying the Java build path"



Warning

Make sure that the workspace is set up correctly and that you do not have errors on the project before starting the next steps.

Section 2. Creating the fraud detection rule agent

In this section, you learn how to create rule agents that detect event patterns that indicate fraud.



Information

In this step, you write the event processing rule that detects suspicious withdrawal patterns and emits a warning to the bank.

The rule is triggered by withdrawal transactions, which are a particular type of banking event that is described in the BMD. When a withdrawal occurs, the amount of transaction is compared to the average amount for transactions over a specific period, in this case, 50 days. If the comparative value seems abnormal, an alert event is emitted.

2.1. Creating the rule agent

- __ 1. If you are not in the Decision Insight perspective, switch to it now.
- __ 2. Create a rule agent named `banking_scenario_agent_fraud_detection`.
 - __ a. In the **Author** task of the Solution Map, click **Add rule agent**.
 - __ b. In the **Project name** field, type: `banking_scenario_agent_fraud_detection`
 - __ c. Click **Finish**.

The `agent.adsc` file opens in the editor.

- __ 3. Complete the agent to match this text:


```
'banking_scenario_agent_fraud_detection' is an agent related to a client ,
processing events :
    - banking event , where this client comes from the client of this banking
event
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and press Ctrl+Shift+F to format the text.

- __ 4. Save the `agent.adsc` file and close it.

2.2. Creating the rule: Check amount versus historical average

- __ 1. Add the **Check amount versus historical average** rule to your rule agent.
 - __ a. Expand the `banking_scenario_agent_fraud_detection` project, right-click **rules**, and click **New > Action rule**.

- ___ b. In the **Name** field of the New Action Rule wizard, type Check amount versus historical average and click **Finish**.

___ 2. Define the rule.

when a withdrawal occurs , called 'NEW TRANSACTION'
definitions

set 'RECENT TRANSACTIONS' to all withdrawals during the last period of 50
days ;

set AVERAGE to the average amount of all withdrawals in 'RECENT
TRANSACTIONS' ;

if

there are at least 5 withdrawals in 'RECENT TRANSACTIONS'

and the amount of 'NEW TRANSACTION' is more than 3 * AVERAGE

then

emit a new fraud alert where

the message is "ATTEMPTING TO WITHDRAW ABNORMALLY HIGH AMOUNT" ,

the fraud event id is the banking event id of 'NEW TRANSACTION' ,

the client is 'the client' ;



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and format the text by pressing Ctrl+Shift+F.

___ 3. Save the rule (Ctrl+S).



Important

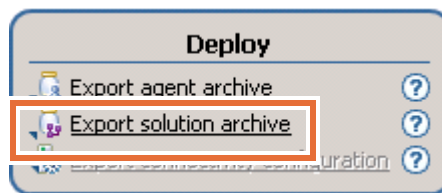
After you create a rule, you should test to ensure that the rule works before you continue writing more rules. For this banking scenario, use the test client that is provided for you.

Section 3. Deploying the solution

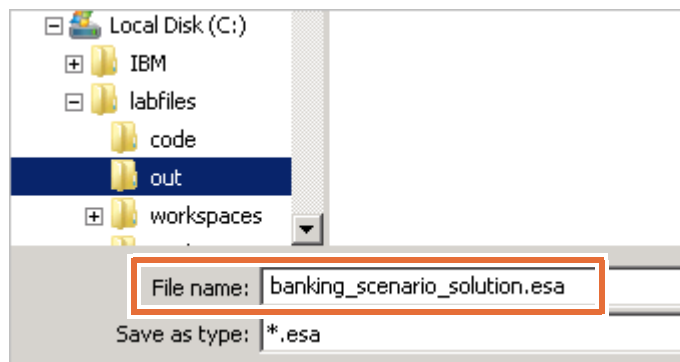
In this section, you export and deploy the solution.

3.1. Exporting the solution

- ___ 1. Before you export the updated solution, increment the solution version number.
 - ___ a. Right-click the **banking_scenario_solution** project and click **Refactor > Increment a Solution Version Number**.
 - ___ b. Set **New solution version** to 3.0 and click **Finish**.
- ___ 2. Export the solution archive.
 - ___ a. In the Solution Map view, in the **Deploy** goal, click the **Export solution archive** link.



- ___ b. In the Solution Archive Export wizard, set the **Output file** field to:
 TrainingDir\Solutions
 where TrainingDir is the installation directory for the lab files. By default,
 TrainingDir is: C:\labfiles
- ___ c. Remove the version number from the file name and click **Save**.



Note

By default, when you select the output directory, the default archive name is provided with the version number and the “.esa” file extension.

- ___ 3. Click **Save** and click **Finish**.
- ___ 4. When prompted to overwrite the existing file, click **Yes**.

3.2. Deploying automatically with Ant tasks

- ___ 1. Switch to the Java perspective.
- ___ 2. In the Package Explorer, expand the **banking_scenario_solution** folder to find the **solution_deploy.xml** file.
- ___ 3. Deploy the solution by right-clicking **solution_deploy.xml** and click **Run As > Ant Build**.



Note

There are two options for **Ant Build**. Choose the one without the ellipsis (...).

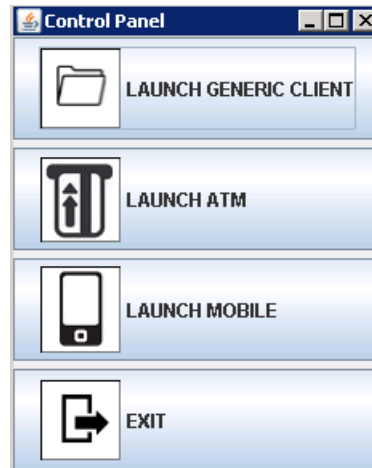
Deployment takes a few moments. After deployment is complete, you see a “Build successful” message in the console.

Section 4. Testing the solution

In this section, you test the solution.

- ___ 1. Make sure that you are in the Java perspective.
- ___ 2. In the Package explorer, expand **banking_scenario_client > src > banking_scenario_client**, right-click **BankingScenarioClient.java**, and click **Run As > Java Application**.

The Control Panel opens.



- ___ 3. Click **LAUNCH GENERIC CLIENT** so that you can see the fraud alerts the Insight Server run time generates.
You use the Generic Client to visualize fraud alerts.
- ___ 4. Click **LAUNCH ATM** and send Withdraw events for **Dave Wakeman**.
 - ___ a. In the **Amount in USD** field, overwrite the amount by typing: 80

- ___ b. Click **Withdraw** five times.



- ___ c. Change the value in the **Amount in USD** field to 240 and click **Withdraw** one time.
- ___ d. Check that the Generic Client shows no fraud alert.
- ___ 5. In the client list, switch to **Di Lang** to send more withdraw events.
- ___ a. In the **Amount in USD** field, type 80 and click **Withdraw** five times,
- ___ b. For the sixth withdrawal, set the **Amount in USD** value to 250 (instead of 240).
Immediately after you withdraw 250, the Generic Client displays a fraud alert.

14-12-22 (MON) 18:14:00> REC: Fraud alert - Di Lang - ATTEMPTING TO WITHDRAW ABNORMALLY HIGH AMOUNT

- ___ 6. Review the Check amount versus historical average rule to understand the rule behavior in relation to this alert.
- ___ 7. After you finish testing, click **Exit** on the Control Panel.

End of exercise

Exercise review and wrap-up

This exercise demonstrated the power of local event aggregates. The aggregates correlated the average across the amount attribute for past withdrawal transaction events. The rule was able to access this average for the particular client through the aggregate.

Exercise 8. Using time-based and location-based reasoning in rules

What this exercise is about

This exercise covers how to correlate time-stamped and geo-localized events.

What you should be able to do

After completing this exercise, you should be able to:

- Use time facets to implement time-based reasoning in rules
- Use location facets to implement spatial reasoning in rules

Introduction

This exercise includes these sections:

- Section 1, "Creating a rule to check time and location compatibility"
- Section 2, "Deploying the solution"
- Section 3, "Testing the solution"

Requirements

This exercise requires that you continue in the same workspace that you used during Exercise 7, "Using event aggregates in rules".

Section 1. Creating a rule to check time and location compatibility

In this section, you add a rule to an existing rule agent. The rule tests against the same banking events that your agent is already subscribed to.



Information

The rule is triggered by any banking event, and immediately verifies the location where the transaction occurred in comparison to the location and timing of previous transactions. If the distance between locations makes the timing of these transactions impossible, a fraud alert is sent to the bank.

1.1. Creating the rule: Check distance to recent events

- ___ 1. Make sure that you are in the Decision Insight perspective.
- ___ 2. Add the **Check distance to recent events** rule to your rule agent.
 - ___ a. Expand the `banking_scenario_agent_fraud_detection` project, right-click **rules**, and click **New > Action rule**.
 - ___ b. In the **Name** field of the New Action Rule wizard, type: `Check distance to recent events`
 - ___ c. Click **Finish**.
- ___ 3. Define the rule.



Information

Facets

This rule uses constructs that are called *facets*. A facet is used in time-based or space-driven logic and is defined in the business model.

In the business model for this scenario, a time facet is defined for client-related events:

`a client related event is a business event time-stamped by a timestamp.`

A location facet is defined for banking events.

`a banking event has a location (a point) used as the default geometry.`

In this rule, you use the time facet of client-related events to compare durations between banking events. You use the location facet to compare distance between the location of event occurrence. You see these comparisons in the definitions part of the rule:

- The distance between NEW and OLD, which technically means the distance between the location of NEW and the location of OLD
- The duration between NEW and OLD, which technically means the duration between the time stamp of NEW and the time stamp of OLD

These shortcuts are made possible through facets.

```
when a banking event occurs , called NEW
definitions
```

```
  set OLD to a banking event ;
```

```
  set DISTANCE to the distance between NEW and OLD in terrestrial miles ;
  set DURATION to the duration between NEW and OLD in minutes ;
```

```
  set 'MINUTES PER HOUR' to 60 ;
```

```
  set 'DRIVE SPEED LIMIT' to 70.0 / 'MINUTES PER HOUR' ;
  set 'DRIVE MAX DISTANCE' to DURATION * 'DRIVE SPEED LIMIT' ;
```

```
  set 'DURATION CHECK IN AND LANDING' to 120 ;
  set 'FLY DURATION' to DURATION - 'DURATION CHECK IN AND LANDING' ;
  set 'FLY SPEED LIMIT' to 500.0 / 'MINUTES PER HOUR' ;
  set 'FLY MAX DISTANCE' to 'FLY DURATION' * 'FLY SPEED LIMIT' ;
```

```
  set 'CLOSE ENOUGH TO DRIVE' to ( DISTANCE is less than 800 ) ;
  set 'FAR ENOUGH TO FLY'      to ( DISTANCE is more than 500 ) ;
```

```
  set 'CAN DRIVE' to 'CLOSE ENOUGH TO DRIVE' and DISTANCE is at most 'DRIVE
MAX DISTANCE' ;
```

```
  set 'CAN FLY'   to 'FAR ENOUGH TO FLY'      and DISTANCE is at most 'FLY MAX
DISTANCE' ;
```

```
if
```

```
  none of the following conditions are true :
```

- 'CAN DRIVE'
- 'CAN FLY'

```
then
```

```
  emit a new fraud alert where
```

```
    the message is "ABNORMAL COMBINATION OF BANKING EVENTS" ,
    the fraud event id is the banking event id of NEW ,
    the client is 'the client' ;
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and press **Ctrl+Shift+F** to format the text.

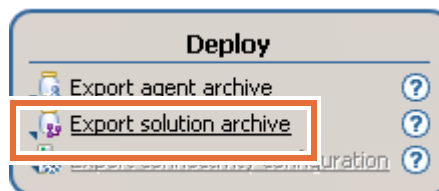
___ 4. Save the rule (Ctrl+S).

Section 2. Deploying the solution

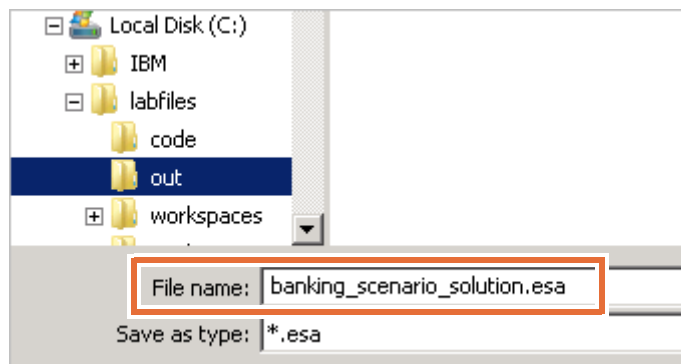
In this section, you export and deploy the solution.

2.1. Exporting the solution

- ___ 1. Before you export the updated solution, increment the solution version number.
 - ___ a. Right-click the **banking_scenario_solution** project and click **Refactor > Increment a Solution Version Number**.
 - ___ b. Set **New solution version** to 4.0 and click **Finish**.
- ___ 2. Export the solution archive.
 - ___ a. In the Solution Map view, in the **Deploy** goal, click the **Export solution archive** link.



- ___ b. In the Solution Archive Export wizard, set the **Output file** field to:
 TrainingDir\Solutions
 where TrainingDir is the installation directory for the lab files. By default,
 TrainingDir is: C:\labfiles
- ___ c. Remove the version number from the file name and click **Save**.



Note

By default, when you select the output directory, the default archive name is provided with the version number and the “.esa” file extension.

- ___ 3. Click **Save** and click **Finish**.
- ___ 4. When prompted to overwrite the existing file, click **Yes**.

2.2. Deploying automatically with Ant tasks

- __ 1. Switch to the Java perspective.
- __ 2. In the Package Explorer, expand the **banking_scenario_solution** folder to find the `solution_deploy.xml` file.
- __ 3. Deploy the solution by right-clicking `solution_deploy.xml` and click **Run As > Ant Build**.



Note

There are two options for **Ant Build**. Choose the one without the ellipsis (...).

Deployment takes a few moments. After deployment is complete, you see a “Build successful” message in the console.

2.3. Deploying the solution to the map tool

- __ 1. Submit the solution.
 - __ a. Open a browser to this URL:
`http://localhost:9080/IAMap/solution.html`
 - __ b. Complete the submission form:
 - **Solution name:** Select **banking_scenario_solution**
 - **Solution title:** Type: `Banking Scenario`
 - **Upload the file:** Click **Browse** and go to the `model.bom` file in the **banking_scenario_bom\bom** folder of your workspace (`TrainingDir\workspace3-fraud`) folder.
 - __ c. Click **Deploy**.

Solution Submission

Solution name

Solution title

Upload the file

The browser returns a successful deployment message.



Success! Your solution 'banking_scenario_solution' was deployed successfully.

You can access your visualization [here](#)

You can configure your visualization [here](#)

__ 2. Open a browser to this URL:

`http://localhost:9080/IAMap/config_raw.html?s=banking_scenario_solution`

__ 3. Replace the contents of the text box with the text in the `config-map.txt` in the **TrainingDir\code** folder, and click **Submit**.



Note

You can copy and paste this text from the `TrainingDir\code\config-map.txt` file.

Note that the last line overwrites the title that you selected in Step 1 unless you update it to match.

Section 3. Testing the solution

In this section, you test the solution. You also use a map application to view your events on a world map. As you submit events during testing, those events appear on the map in a different color for each client.

- ___ 1. In the Package explorer of the Java perspective, expand **banking_scenario_client > src > banking_scenario_client**, right-click **BankingScenarioClient.java**, and click **Run As > Java Application**.

The Control Panel opens.



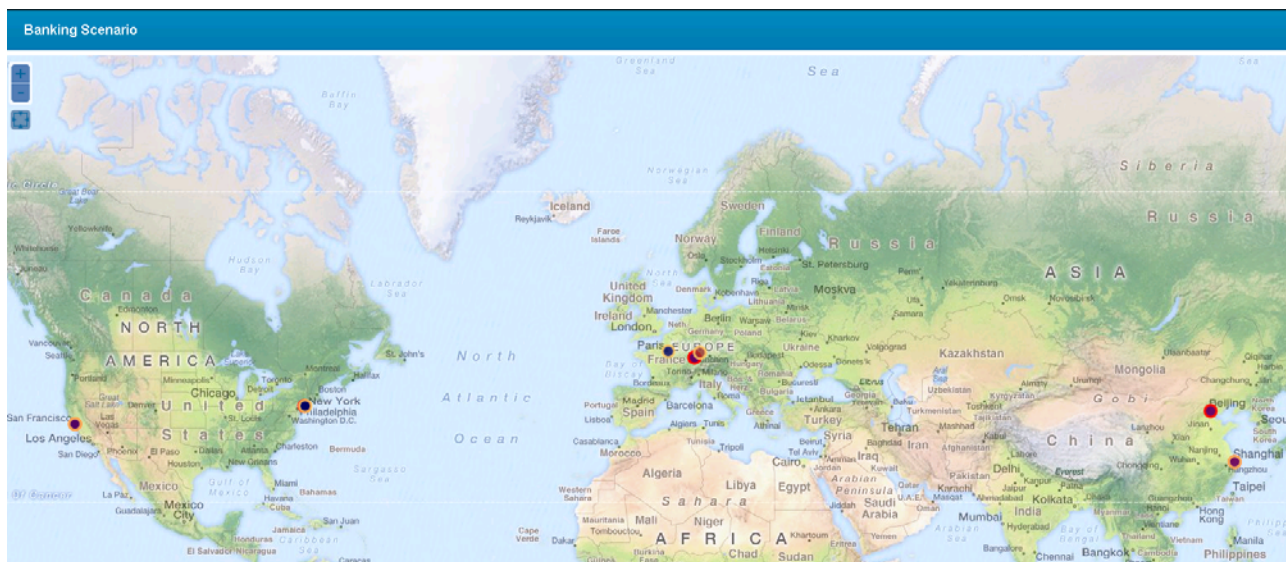
- ___ 2. Click **LAUNCH GENERIC CLIENT** so that you can see the fraud alerts generated by the Insight Server run time.
- ___ 3. Click **LAUNCH ATM** twice and place them side-by-side on your screen.



For this test, the same client withdraws from two different locations within seconds.

___ 4. To view the events on the map, open a browser to this URL:

http://localhost:9080/IAMap/index.html?s=banking_scenario_solution



___ 5. Set the ATM options:

- ATM 1: **New_York** and **David Dean**
- ATM 2: **San_Francisco** and **David Dean**

___ 6. Click **Withdraw** first on ATM 1, and then on ATM 2.

Make sure that the time lapse between the two clicks is less than the time David needs to drive from New York to Newark. (Flying from New York to Newark does not make sense.)

Immediately after David withdraws cash from Newark, the Generic Client shows a fraud alert:

2014-12-22 (MON) 18:45:44> REC: Fraud alert - David Dean - ABNORMAL COMBINATION OF BANKING EVENTS

___ 7. Review the **Check distance to recent events** rule to understand the behavior.



Note

You can reproduce this scenario with the mobile device interface and the **Check Account** operation. Make sure that you select the same client for both devices but with different cities.

___ 8. After you finish testing, click **EXIT** in the Control Panel.

At the end of this exercise, the map should have the following features. The trajectory of each client is represented in a specific color. Normal events are marked with an orange line. Fraud events are marked with a red line.

End of exercise

Exercise review and wrap-up

The exercise showed you how you can correlate time-stamped and geo-localized events.

Exercise 9. Testing for the absence of events

What this exercise is about

This exercise covers how to recognize when an event did not occur and respond in a timely manner.

What you should be able to do

After completing this exercise, you should be able to:

- Test for the absence of events

Introduction

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Creating a fraud management rule agent"
- Section 3, "Deploying the solution"
- Section 4, "Testing the solution"

Requirements

This exercise requires that you complete the previous exercise before continuing or that you use the **TrainingDir\workspace5-absent** workspace.

Section 1. Setting up your workspace

In Insight Designer, you can continue working in your current workspace or switch to the workspace that is provided for this exercise and includes the solution to the previous exercise.



Important

If you choose to switch workspaces, you must follow these steps before you continue:

- "Setting the target platform"
- "Importing the projects"
- "Setting the target platform"
- "Verifying the Java build path"

- ___ 1. To switch workspaces:
 - ___ a. From the **File** menu, click **Switch Workspace > Other**.
 - ___ b. When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspace5-absent** folder.
- ___ 2. Follow the steps in Section 1, "Setting up your workspace" from Exercise 5, "Writing and testing rules" to complete the workspace setup.

Section 2. Creating a fraud management rule agent

In this section, you learn how to create a rule agent that tests for the absence of an event. You write a rule to ensure that fraud alerts are handled appropriately and in a timely manner. In particular, if a fraud alert is sent to a client, but no response is received from the client within a specific time lapse, a reminder is sent.



Requirements

When the bank receives a fraud detection alert, clients are asked to verify within 60 minutes whether they are aware of the suspicious transaction.

If the bank does not receive a response from a client about a suspicious transaction, the client is sent a reminder to contact the bank. The reminder should be sent 30 minutes after the initial notification is sent to the client.

2.1. Creating the rule agent

In this step, you create a rule agent that detects suspicious transaction patterns and emits a warning to the client.

- ___ 1. If you are not in the Decision Insight perspective of Insight Designer, switch to it now.
- ___ 2. Create a rule agent named `banking_scenario_agent_fraud_management`.
 - ___ a. In the **Author** task of the Solution Map, click **Add rule agent**.
 - ___ b. In the **Project name** field, type: `banking_scenario_agent_fraud_management`
 - ___ c. Click **Finish**.

The `agent.adsc` file opens in the editor.

- ___ 3. Complete the agent to match this text:


```
'banking_scenario_agent_fraud_management' is an agent related to a client ,
processing events :
- fraud alert , where this client comes from the client of this fraud alert
- confirmation from client , where this client comes from the client of
this confirmation from client
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder and press **Ctrl+Shift+F** to format the text.

- ___ 4. Save the `agent.adsc` file and close it.

2.2. Creating a rule: Ask client to confirm within 60 minutes

- ___ 1. Add the Ask client to confirm within 60 minutes rule to your rule agent.
 - ___ a. Expand the `banking_scenario_agent_fraud_management` project, right-click **rules**, and click **New > Action rule**.
 - ___ b. In the **Name** field of the New Action Rule wizard, type: Ask client to confirm within 60 minutes
 - ___ c. Click **Finish**.
- ___ 2. Define the rule to match this text.

The rule is triggered by fraud activity. However, you already wrote fraud detection rules so you do not need to retest for fraudulent conditions. You can write these rules to emit an outbound event that is based on the knowledge that fraudulent activity occurred.

```
when a fraud alert occurs
then
    emit a new notification to client where
        the client is the client of this fraud alert ,
        the code is CALL_BANK_60 ;
```



Hint

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder.

- ___ 3. Save the rule (Ctrl+S).

2.3. Creating a rule: Remind after 30 min if no action taken

The rule tests for the **absence of an event**. Note for testing purposes, the rule uses 30 seconds instead of minutes.

- ___ 1. Add the **Remind after 30 min if no action taken** rule to your rule agent.
 - ___ a. Expand the `banking_scenario_agent_fraud_management` project, right-click **rules**, and click **New > Action rule**.
 - ___ b. In the **Name** field of the New Action Rule wizard, type: Remind after 30 min if no action taken
 - ___ c. Click **Finish**.

__ 2. Define the rule.

```
when a fraud alert has occurred 30 seconds ago
if
    there is no confirmation from client
        where this confirmation from client is after this fraud alert ,
then
    emit a new notification to client where
        the client is the client of this fraud alert ,
        the code is CALL_BANK_30 ;
```

**Hint**

You can copy and paste this text from the `agents.txt` file in the **TrainingDir\code** folder.

__ 3. Save the rule (Ctrl+S).

**Important**

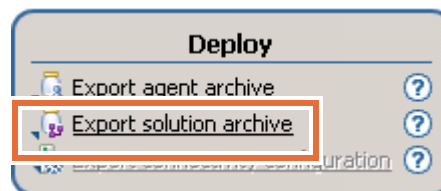
After you create a rule, you should test to ensure that the rule works before you continue writing more rules. For this banking scenario, a test client is provided for you.

Section 3. Deploying the solution

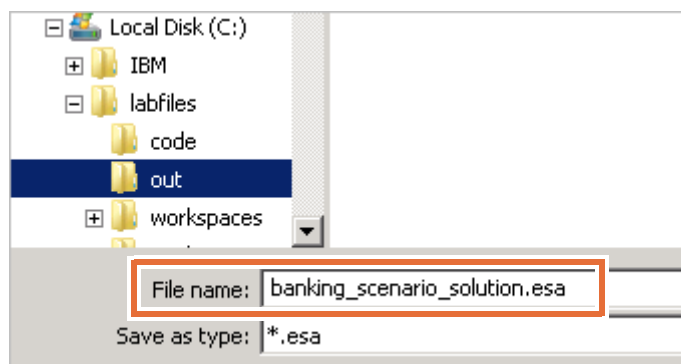
In this section, you deploy the solution.

3.1. Exporting the solution

- ___ 1. Before you export the updated solution, increment the solution version number.
 - ___ a. Right-click the **banking_scenario_solution** project and click **Refactor > Increment a Solution Version Number**.
 - ___ b. Set **New solution version** to 4.0 and click **Finish**.
- ___ 2. Export the solution archive.
 - ___ a. In the Solution Map view, in the **Deploy** goal, click the **Export solution archive** link.



- ___ b. In the Solution Archive Export wizard, set the **Output file** field to:
 TrainingDir\Solutions
 where TrainingDir is the installation directory for the lab files. By default,
 TrainingDir is: C:\labfiles
- ___ c. Remove the version number from the file name and click **Save**.



Note

By default, when you select the output directory, the default archive name is provided with the version number and the “.esa” file extension.

- ___ 3. Click **Save** and click **Finish**.
- ___ 4. When prompted to overwrite the existing file, click **Yes**.

3.2. Deploying automatically with Ant tasks

- ___ 1. Switch to the Java perspective.
- ___ 2. In the Package Explorer, expand the **banking_scenario_solution** folder to find the **solution_deploy.xml** file.
- ___ 3. Deploy the solution by right-clicking **solution_deploy.xml** and click **Run As > Ant Build**.



Note

There are two options for **Ant Build**. Choose the one without the ellipsis (...).

Deployment takes a few moments. After deployment is complete, you see a “Build successful” message in the console.

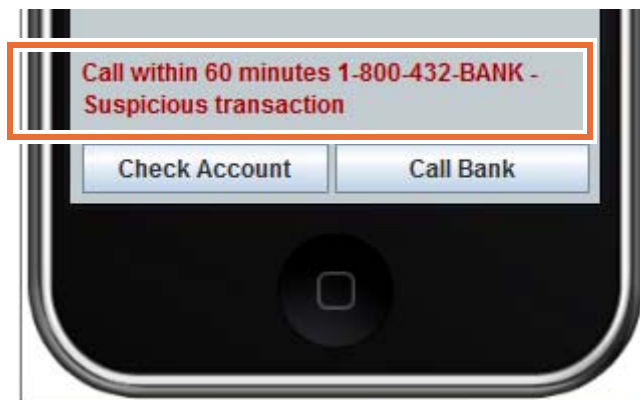
Section 4. Testing the solution

In this section, you test the solution.

- ___ 1. Make sure that you are in the Java perspective.
- ___ 2. In the Package explorer, expand **banking_scenario_client > src > banking_scenario_client**, right-click **BankingScenarioClient.java**, and click **Run As > Java Application**.
- ___ 3. Click **LAUNCH GENERIC CLIENT** so that you can see the fraud alerts generated by the Insight Server run time.
- ___ 4. Click **LAUNCH MOBILE** and select **June Yoshii**.
- ___ 5. Click **LAUNCH ATM** twice and redo the steps from the previous exercise with **June Yoshii** as the client.
 - ___ a. Place the ATM interfaces side-by-side on your screen.
 - ___ b. Set the ATM options:
 - ATM 1: **New_York** and **June Yoshii**
 - ATM 2: **Newark** and **June Yoshii**
 - ___ c. Click **Withdraw** first on ATM 1, and then on ATM 2.

Make sure that the time lapse between the two clicks is less than the time June needs to drive from New York to Newark.

Immediately after June makes the second withdrawal, June's mobile device displays a message.



- ___ d. Wait for 30 seconds to see the reminder.



- ___ 6. Review the `Ask client to confirm within 60 minutes` rule to understand the rule behavior in relation to the alert.
- ___ 7. Review the `Remind after 30 min if no action taken` rule to understand the rule behavior in relation to the reminder message.
- ___ 8. Redo Step 5, and then click **Call Bank** on the mobile interface within 30 seconds.
- Because June is calling her bank, her bank can report the fraud case to her. Therefore, she does not get a reminder notification.
- ___ 9. After you finish testing, click **Exit** on the Control Panel.

End of exercise

Exercise review and wrap-up

The exercise demonstrated how to test for the absence of an event and respond in a timely manner.

Exercise 10. Deploying solutions

What this exercise is about

This exercise covers how to manage solution deployment with the solutionManager script.

What you should be able to do

After completing this exercise, you should be able to:

- Export a solution as a deployable archive
- Use solutionManager to deploy and undeploy solutions

Introduction

This exercise includes these sections:

- Section 1, "Deploying and removing a solution manually"

Requirements

This exercise requires that you continue working in the previous workspace.

Section 1. Deploying and removing a solution manually

In this section, you see how to deploy a solution to the sample server, and view the deployed files. Then, you see how to remove a solution and verify that the solution files are cleaned out of the server and the solution file directories.

1.1. Managing deployed solutions

- ___ 1. Open a command prompt window and go to the `<InstallDir>/runtime/ia/bin` directory.
- ___ 2. View the list of deployed solutions by running the `solutionManager list` command.

```
solutionManager list local
```

The result is a list of all deployed solutions.

```
C:\IBM\ODMInsights87\runtime\ia\bin>solutionManager list local
banking_scenario_solution-1.0
banking_scenario_solution-2.0
banking_scenario_solution-3.0
banking_scenario_solution-4.0
banking_scenario_solution-5.0 (current)
```

Notice that beside the listed solutions, you also see “current” in parentheses.

When you have several versions of a solution that are deployed, you can find the current version by:

- Running the `list` command to see which versions of all your deployed solutions are active
- Running the `getcurrentversion` command for a specific solution

- ___ 3. Run the `getcurrentversion` command for `banking_scenario_solution`.

```
solutionManager getcurrentversion banking_scenario_solution
```

The result is the current version of your solution.

```
C:\IBM\ODMInsights87\runtime\ia\bin>solutionManager getcurrentversion banking_sc
enario_solution
Current version of solution banking_scenario_solution on server localhost is ban
king_scenario_solution-5.0
```

- ___ 4. View the deployed files by opening Windows Explorer and going to the solutions directories:

- `InstallDir\runtime\solutions\lib`

The list of deployed files includes JAR files for the global aggregates and the rule and Java agents. Notice the version 1.0 files.

- `InstallDir\runtime\solutions\lib\features`

The solution MF files, which include the solution version number, are stored in the features folder.

**Note**

Later, you return to these directories to make sure that remaining files are deleted after undeploying a solution.

1.2. Undeploying and deleting solution files

- __ 1. Stop the solution by running the stop command.

```
solutionManager stop banking_scenario_solution
```

The result is that the current solution is no longer running.

```
C:\IBM\ODMInsights87\runtime\ia\bin>solutionManager stop banking_scenario_solution
Solution successfully stopped: banking_scenario_solution
```

- __ 2. Run the undeploy command.

```
solutionManager undeploy local banking_scenario_solution-1.0
```

Before you can delete the solution files, you must stop the server.

- __ 3. Stop the server so that you can delete the version 1.0 solution files.

- __ a. In the command prompt window, change to the *InstallDir/runtime/wlp/bin* directory

```
cd ..\..
cd wlp\bin
```

- __ b. Run the server management stop script.

```
server stop cisDev
```

```
C:\IBM\ODMInsights87\runtime\wlp\bin>server stop cisDev
Stopping server cisDev.
Server cisDev stopped.
```

- __ 4. Go back to the *InstallDir/runtime/ia/bin* directory and run the delete command to delete the solution files.

```
cd ..\..
cd ia\bin
solutionManager delete local banking_scenario_solution-1.0
```

As a result, you see a list of all deleted files and you are prompted to restart the server with the `--clean` option.

- __ 5. Go back to the *InstallDir/runtime/wlp/bin* directory and restart the server by typing this command.

```
cd ..\..
cd wlp\bin
server start cisDev --clean
```

___ 6. Reopen Windows Explorer to check for leftover solution files in these directories:

- InstallDir\runtime\solutions\lib\features
- InstallDir\runtime\solutions\lib

Notice the version 1.0 files are deleted.

End of exercise

Exercise review and wrap-up

This exercise covered how to manage solution deployment with the solutionManager script.

Exercise 11. Defining connectivity for a solution

What this exercise is about

This exercise demonstrates how to define connectivity.

What you should be able to do

After completing this exercise, you should be able to:

- Define inbound and outbound connectivity for a solution
- Deploy connectivity

Introduction

This exercise includes these sections:

- Section 1, "Setting up your workspace"
- Section 2, "Defining connectivity"
- Section 3, "Exporting the solution and connectivity"

Requirements

This exercise requires that you continue in the workspace that you used during Exercise 9, "Testing for the absence of events" or that you switch to the **TrainingDir\workspaces-final** workspace.

Section 1. Setting up your workspace

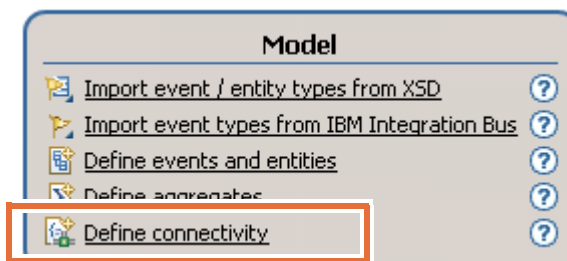
In Insight Designer, you can continue working in your current workspace or switch to the workspace that is provided for this exercise and includes the solution to the previous exercise.

- ___ 1. To switch workspaces:
 - ___ a. From the **File** menu, click **Switch Workspace > Other**.
 - ___ b. When prompted for a workspace, click **Browse**, and go to the **TrainingDir\workspaces-final** folder.
- ___ 2. If you are not in the Decision Insight perspective, switch to it now.

Section 2. Defining connectivity

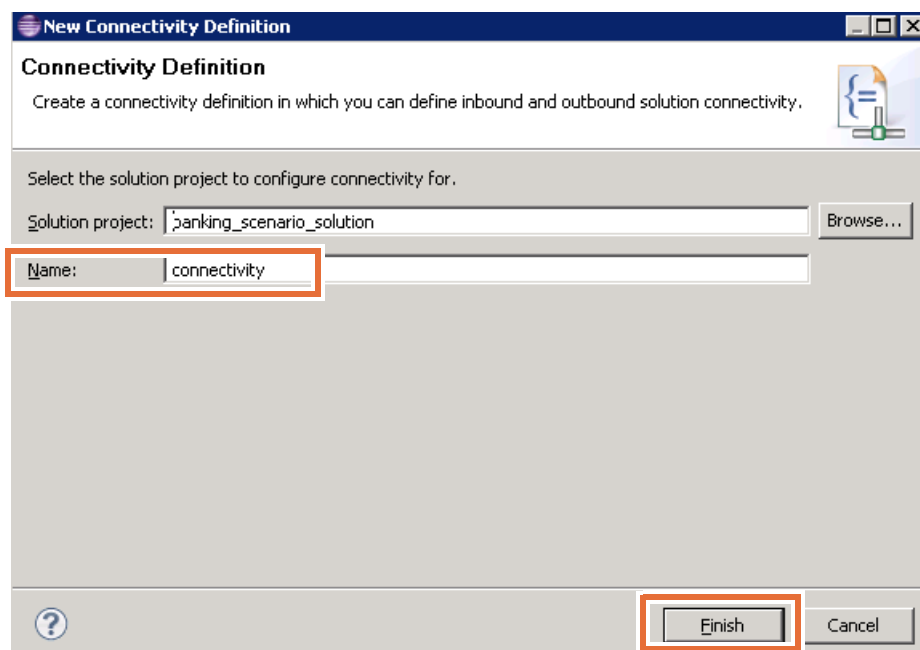
In Insight Designer, you can continue working in your current workspace or switch to the workspace that is provided for this exercise and includes the solution to the previous exercise.

- ___ 1. In the Decision Insight perspective, click **banking_scenario_solution** to open the **Solution Map** view.
- ___ 2. In the Model goal, click the **Define connectivity** link.

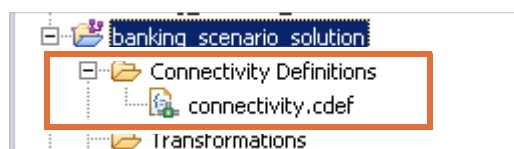


The New Connectivity Definition wizard opens.

- ___ 3. Leave the **Name** field set to connectivity and click **Finish**.



The `connectivity.cdef` file opens in the editor and is listed in the Connectivity Definitions folder of the solution.



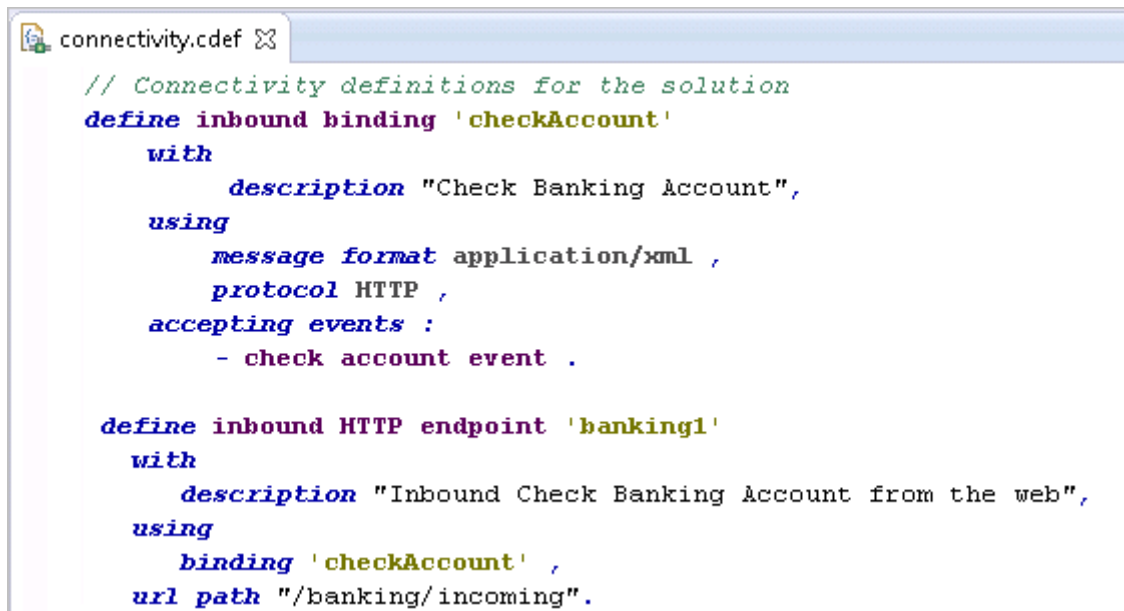
___ 4. Define the inbound binding and endpoint.

- ___ a. In the editor, after the introductory comment line, type the following text to define the inbound binding.

```
define inbound binding 'checkAccount'
  with
    description "Check Banking Account",
  using
    message format application/xml ,
    protocol HTTP ,
  accepting events :
    - check account event .
```

- ___ a. Next, type the following text to define the inbound endpoint.

```
define inbound HTTP endpoint 'banking1'
  with
    description "Inbound Check Banking Account from the web",
  using
    binding 'checkAccount' ,
  url path "/banking/incoming".
```



```
connectivity.cdef
// Connectivity definitions for the solution
define inbound binding 'checkAccount'
  with
    description "Check Banking Account",
  using
    message format application/xml ,
    protocol HTTP ,
  accepting events :
    - check account event .

define inbound HTTP endpoint 'banking1'
  with
    description "Inbound Check Banking Account from the web",
  using
    binding 'checkAccount' ,
  url path "/banking/incoming".
```



Note

You use the completion menu of the Connectivity Definition editor to specify the required elements in the .cdef file. You can also copy and paste this text from the connectivity.txt file in the **TrainingDir\code** folder.

- ___ b. Save your work and close the file.

Section 3. Exporting the solution and connectivity

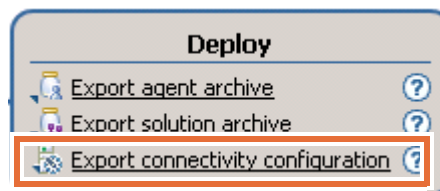
In this section, you export and deploy the solution and the connectivity configuration.

3.1. Exporting the solution

- ___ 1. Increment the solution version number, for example, to 5.1.
 - ___ a. In Solution Explorer, right-click the **banking_scenario_solution** project and select **Refactor > Increment a Solution Version Number**.
 - ___ b. Set **New solution version** to 5.1 and click **Finish**.
- ___ 2. Export the solution archive to the TrainingDir\Solutions directory.
 - ___ a. From the Solution Map, click **Export solution archive**.
 - ___ b. Set the path to the TrainingDir\Solutions directory and save the ESA file without the version number.

3.2. Exporting the connectivity configuration

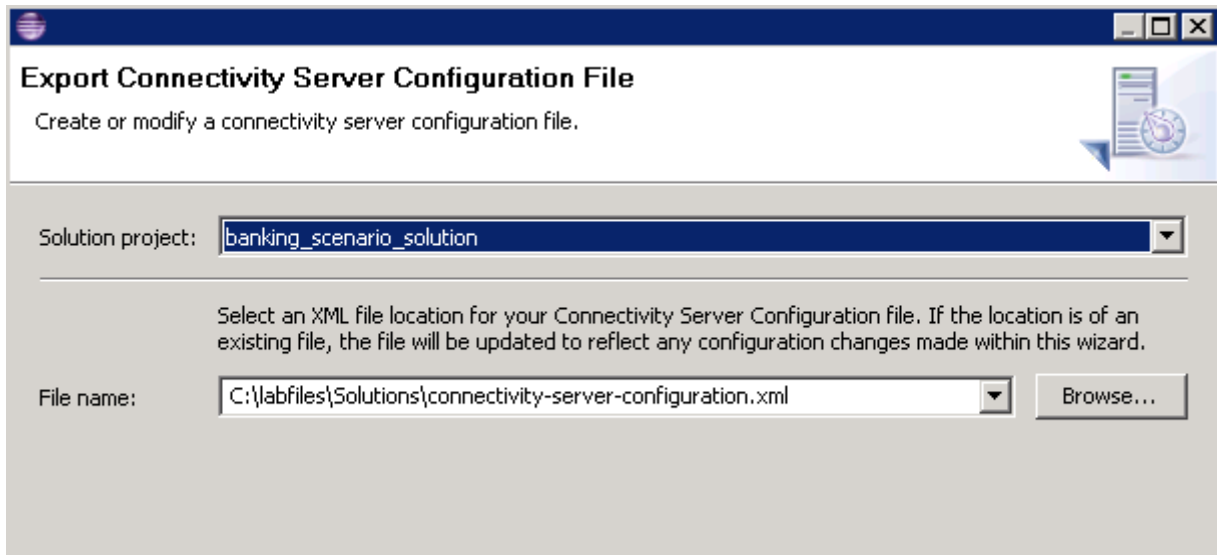
- ___ 1. In the Solution Map view, in the Deploy goal, click the **Export connectivity configuration** link.



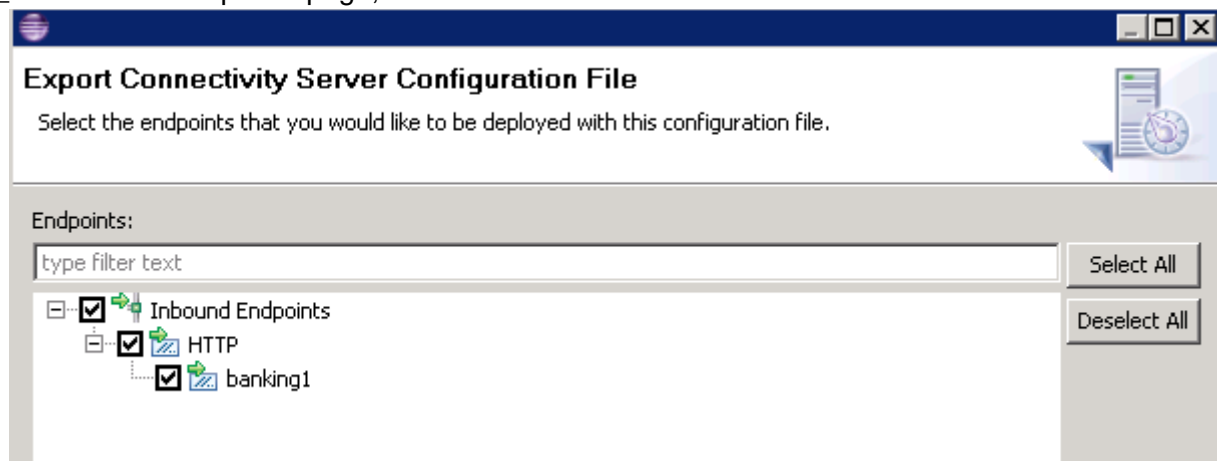
- ___ 2. The Export Connectivity Server Configuration File wizard opens.
- ___ 3. Click **Browse** and set the path to the TrainingDir\Solutions directory. When you save the path, the **File name** field is set to the path plus the default file name:

TrainingDir\Solutions\connectivity-server-configuration.xml

where TrainingDir is the installation directory for the lab files. By default, TrainingDir is set to: C:\labfiles



- ___ 4. Click **Next**.
- ___ 5. On the Endpoints page, click **Select All**.



- ___ 6. Click **Finish**.
- ___ 7. Open the TraininDir\Solutions directory in Windows Explorer and you see your connectivity-server-configuration.xml file.

3.3. Using solutionManager to manually deploy the solution

- ___ 1. Open a command prompt window to manually deploy the banking_scenario_solution solution.
 - ___ a. Go to the InstallDir\runtime\ia\bin folder.
 - ___ b. Type this command:


```
solutionManager deploy local
c:\labfiles\Solutions\banking_scenario_solution.esa
```

The command prompt window shows that the solution was successfully deployed.

```
C:\IBM\ODMInsights87\runtime\ia\bin>solutionManager deploy local c:\labfiles\Solutions\banking_scenario_solution.esa
Server configuration file successfully updated for server: cisDev
```

3.4. Using connectivityManager to deploy the connectivity

1. In the command prompt window, still in the InstallDir\runtime\ia\bin directory, deploy the connectivity by typing this command:

```
connectivityManager deploy local
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\connectivity-server-configuration.xml
```

You see a message that the connectivity was successfully deployed and the inbound endpoint is active.

```
C:\IBM\ODMInsights87\runtime\ia\bin>connectivityManager deploy local C:\labfiles\Solutions\banking_scenario_solution.esa C:\labfiles\Solutions\connectivity-server-configuration.xml
CWMBE1146I: Reading the input file: C:\labfiles\Solutions\banking_scenario_solution.esa
CWMBE1475I: The connectivity server configuration file for the solution "banking_scenario_solution" contains the configuration required for the specified endpoints.
CWMBE1148I: Writing to the output file: C:\Users\ADMINI~1\AppData\Local\Temp\2\banking_scenario_solution-inbound.ear1057868634149119709.tmp
CWMBE1144I: Successfully copied the file from "C:\Users\Administrator\AppData\Local\Temp\2\banking_scenario_solution-inbound.ear1057868634149119709.tmp" to "C:\IBM\ODMInsights87\runtime\wlp\usr\servers\cisDev\apps\banking_scenario_solution-inbound.ear".
CWMBE1144I: Successfully copied the file from "C:\labfiles\Solutions\connectivity-server-configuration.xml" to "C:\IBM\ODMInsights87\runtime\wlp\usr\servers\cisDev\banking_scenario_solution-config.xml".
CWMBE1452I: Successfully deployed connectivity for the solution "banking_scenario_solution".
CWMBE1471I: Connectivity is already active for the solution "banking_scenario_solution".
CWMBE1498I: Number of active inbound endpoints: 1
CWMBE1499I: Number of active outbound endpoints: 0
```

End of exercise

Exercise review and wrap-up

This exercise demonstrated how to define connectivity.

