Course Exercises Guide

Developing Solutions with IBM Decision Server Insights V8.8

Course code WB399 / ZB399 ERC 1.2

IBM Training

August 2016 edition

Notices

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

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

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

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

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

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

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

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

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

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

Trademarks

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

Contents

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®	DB2®	IBM SmartCloud®
ILOG®	Insight™	PartnerWorld®
Redbooks®	Redpaper™	SPSS®
WebSphere®	z/OS®	

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

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 objectives

After completing the exercises, you should be able to:

- Create a Decision Server Insights solution
- Create rule agents and Java agents
- Implement the business logic in rules that use time-based and location-based reasoning
- Create global aggregates and use them to identify outlier behavior
- Write rules that test for the absence of an event
- Deploy solutions to the Insight Server runtime and manage deployed artifacts
- Use the REST API to manage solution artifacts
- Use Insight Inspector and other test clients to test implementation behavior
- Model and define connectivity for a solution

Working on the exercises

Each exercise depends on successful completion of the previous exercises.

The exercises can be categorized into these groups:

- ***Starting from scratch***
  - 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"
- ***Writing and testing***
  - 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, "Testing solutions"
  - Exercise 11, "Using the Map Viewer"
  - Exercise 12, "Defining connectivity"
- ***Administration***
  - Exercise 13, "Installing Decision Server Insights"
  - Exercise 14, "Configuring Decision Server Insights"
  - Exercise 15, "Deploying solutions"
  - Exercise 16, "Administering Decision Server Insights"

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.

General exercise information

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

Important

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

- `C:\labfiles` for the Windows platform
- `/usr/labfiles` for the Linux platform

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

User IDs and passwords

Here is a list of user ID and password information for this course.

Entry point	User ID	Password
VMware image	`administrator`	`web1sphere`
Windows 2008 R2	`administrator`	`web1sphere`
Single-sign-on ID for ODM installation and user ID for WebSphere Application Server and Decision Server	`odm`	`odm`

How to follow the exercise instructions

Structure of exercise steps

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 done.
- The lettered substeps (a, b, c) provide detailed guidance on how to complete the action.

Information

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

Here is an example from this exercise.

Example

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
- Button names

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.

Optional exercise sections

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

File references

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

- ***<InstallDir>***: This directory refers to the IBM Decision Server Insights V8.8 installation directory. During the class, you install Decision Server Insights.
- ***<LabfilesDir>***: This directory refers to the directory that contains the files that are required during demonstrations, exercises, and the workshop steps, such as samples of code that you can copy and paste.By default, this directory is: `C:\labfiles`

Note

If you are not using the VMware images that are provided with this course, ask the installer of your environment, or your instructor, where to find the ***<LabfilesDir>*** directory.

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 V8.8.1 > Insight Designer**.

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

```
C:\labfiles\workspaces\myWorkspace
```

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

Using the product documentation

The product documentation is not installed locally on the VMware image that is provided with this course.

If you have Internet access, you can also view the online IBM Knowledge Center for Operational Decision Manager available at this URL:

`www.ibm.com/support/knowledgecenter/SSQP76_8.8.0/welcome/kc_welcome_odmV.html`

Classes that are delivered through the IBM Remote Lab Platform (IRLP)

To log on to the lab virtual machine image, use ID `administrator` and password `web1sphere`, and then follow the instructions.

Refer to the `readme.txt` file on the lab image desktop for possible additional information.

Important

Online course material updates might exist for this course. To check for updates, visit the IBM Cloud Education wiki at:

`http://ibm.biz/CloudEduCourses`

Exercise 1. Getting started with Decision Server Insights

Estimated time

01:30

Overview

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

Objectives

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:

Requirements

This exercise requires that you have the Decision Server Insights V8.8 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 ***<LabfilesDir>***)

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

Section 1. Checking for course corrections

Stop

Course updates and corrections

A Course Corrections document might be available for this course.

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

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

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

1. Go to the following URL: `http://www.ibm.biz/CloudEduCourses`
2. Find your course in the list and click the link.
3. Click the **Attachments** tab to see whether an errata document exists with updated instructions.
4. To save the file to your computer, click the document link and follow the prompts.

Section 2. Installing Decision Server Insights

You install Decision Server Insights by using IBM Installation Manager.

2.1. Verify your lab environment

Your lab environment includes four hosts. For this exercise, you install Decision Server Insights on the main host. The default name for the main host is dsiHost1. However, your host might have a unique name.

__ 1. Make a note the host names and IP addresses for the hosts that were assigned to you for your lab environment. See Appendix A, "Host names and IP addresses".

2.2. Installing IBM Installation Manager

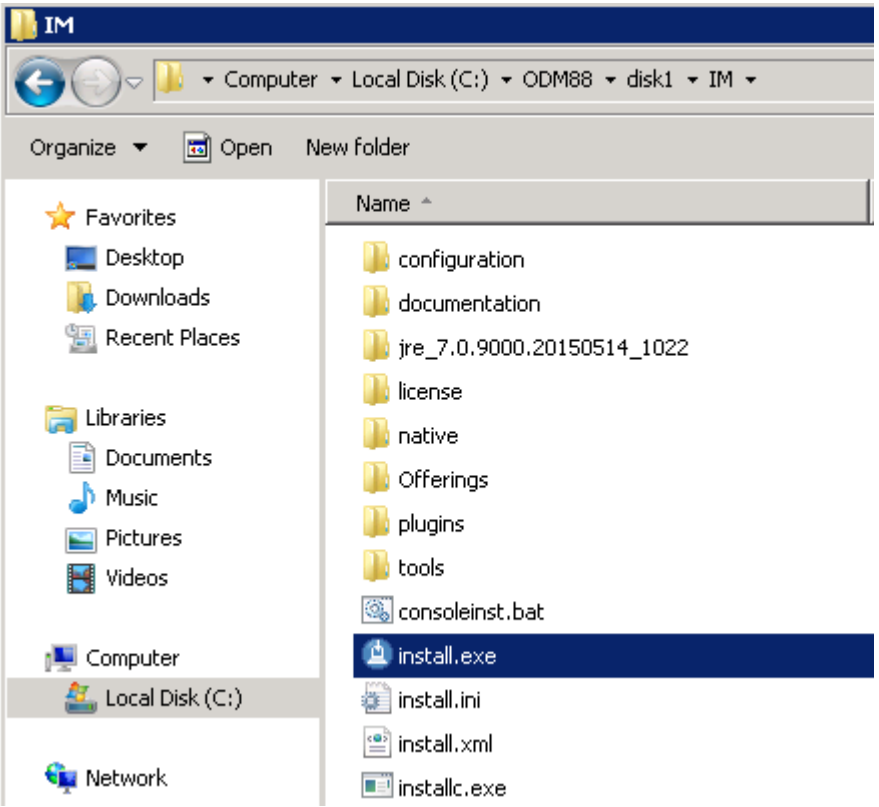
__ 1. Go to the *<LabfilesDir>*`\ODMinstallers` directory.

By default, *<LabfilesDir>* refers to `C:\labfiles` on the computer lab environment that is provided for this course.

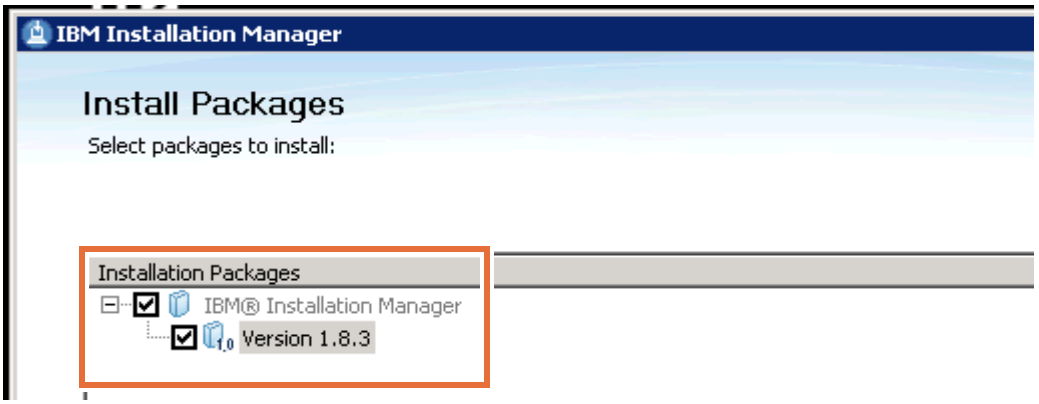
__ 2. Install IBM Installation Manager.

IBM Installation Manager is an installation program that uses remote or local software repositories to install, modify, or update certain IBM products.

__ a. In the *<LabfilesDir>*`\ODMinstallers` directory, open the **disk1 > IM64** folder, and double-click the `install.exe` file.



__ b. In the Install Packages window, make sure that the **IBM Installation Manager** is selected and click **Next**.

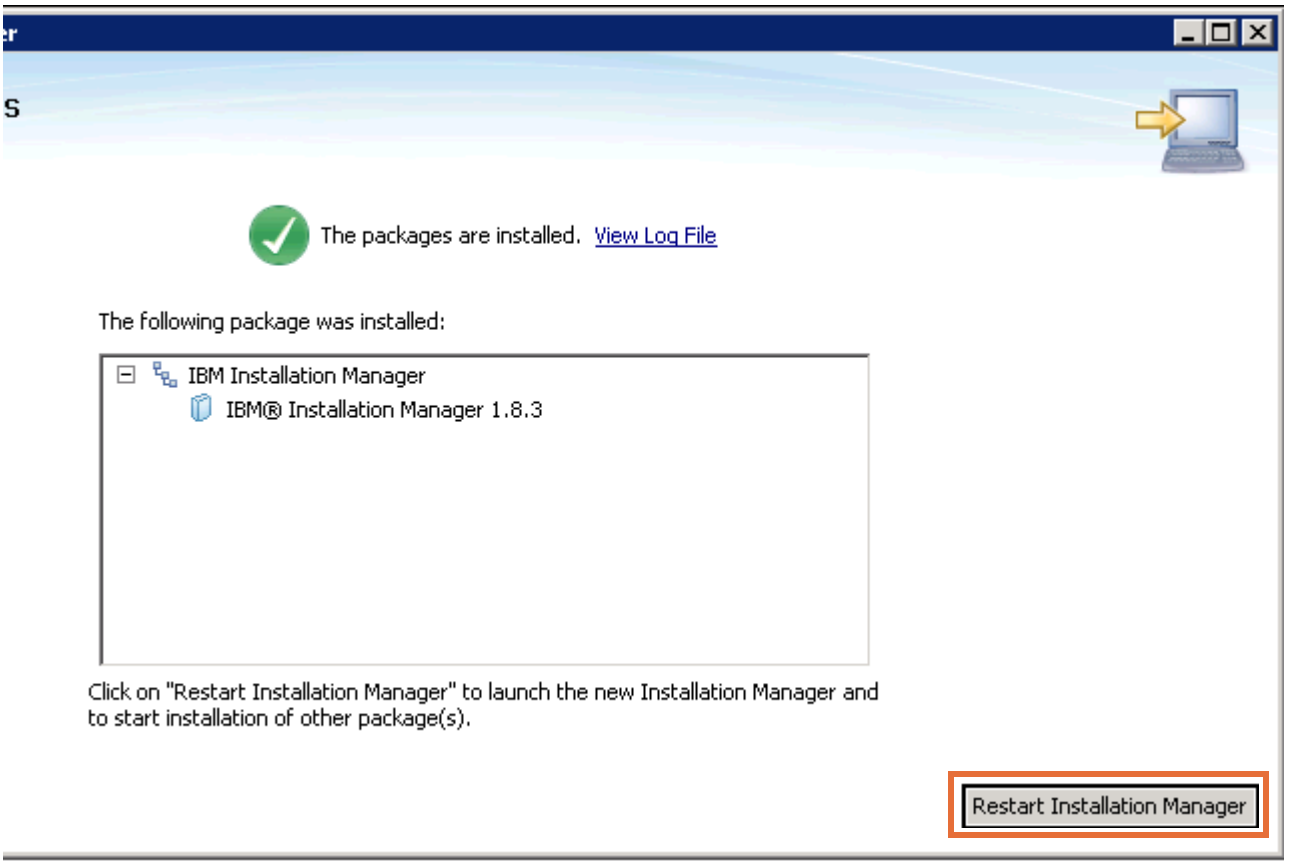


__ c. Select **I accept the terms in the license agreement** and click **Next**.

__ d. Keep the default installation path and click **Next**.

__ e. Click **Install**.

__ 3. When the installation of Installation Manager is complete, click **Restart Installation Manager**.

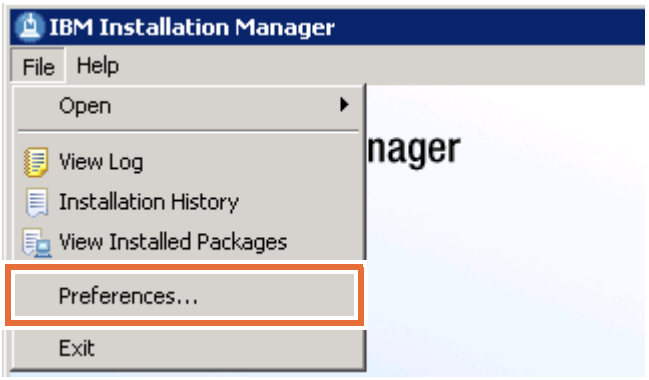


You can now use Installation Manager to install IBM software.

2.3. Installing WebSphere Application Server and the WebSphere Java SDK V7

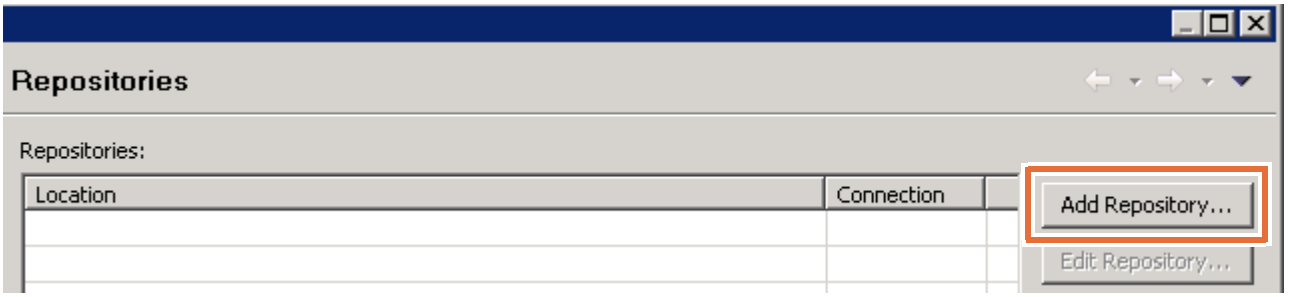
__ 1. Define the installation repository for WebSphere Application Server.

__ a. From the **File** menu, click **Preferences**.

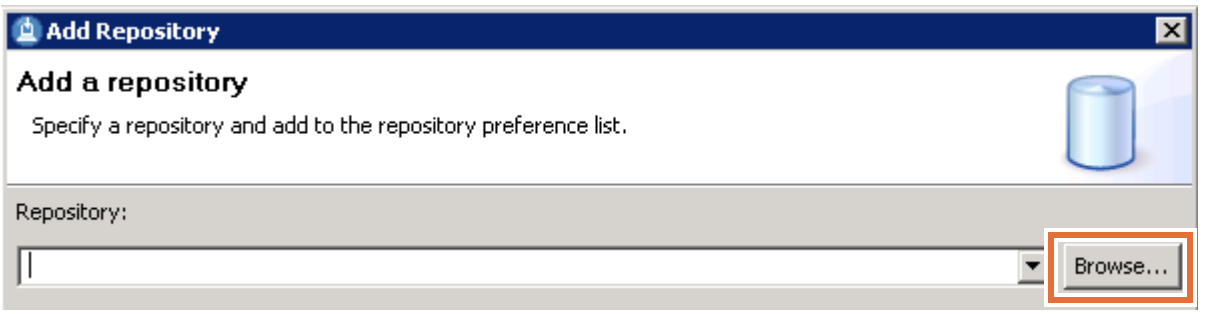


__ b. In the Preferences window, make sure that you are on the Repositories page.

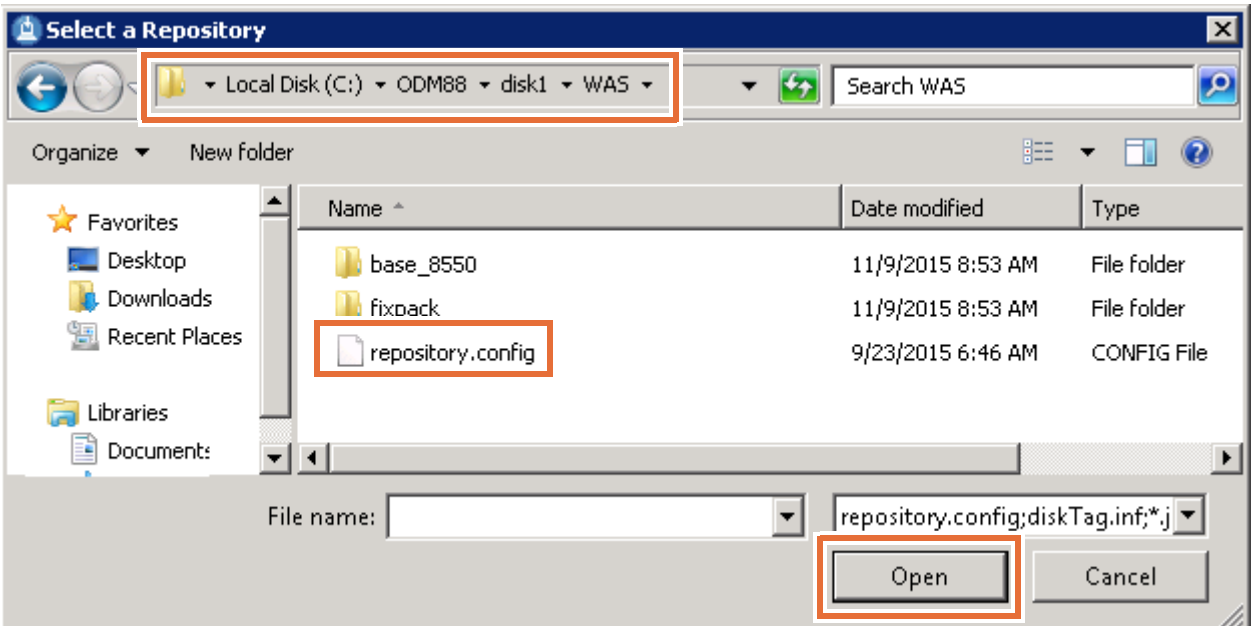
__ c. Click **Add Repository**.



__ d. In the Add Repository window, click **Browse**.



__ e. In the Select a Repository window, go to the **disk1 > WAS** folder, select **repository.config**, and click **Open**.



__ f. In the Add Repository window, click **OK** to add the repository for WebSphere Application Server.

__ 2. Add the installation repository for the WebSphere SDK Java Technology Edition.

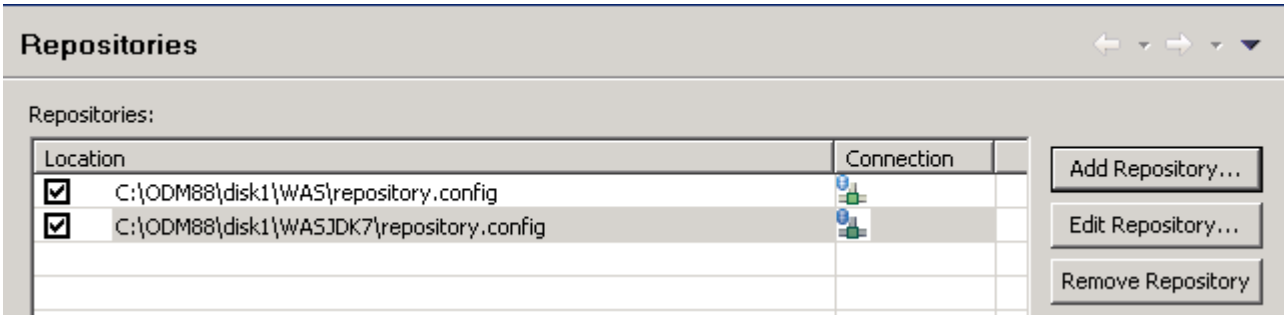
__ a. On the Repositories page, click **Add Repository**.

__ b. In the Add Repository window, click **Browse** and go to the **disk1 > WASJDK7** folder.

__ c. Select **repository.config** and click **Open**.

__ d. In the Add Repository window, click **OK** to add the **WASJDK** repository.

On the Repositories page, the two installation repositories are listed.



Troubleshooting

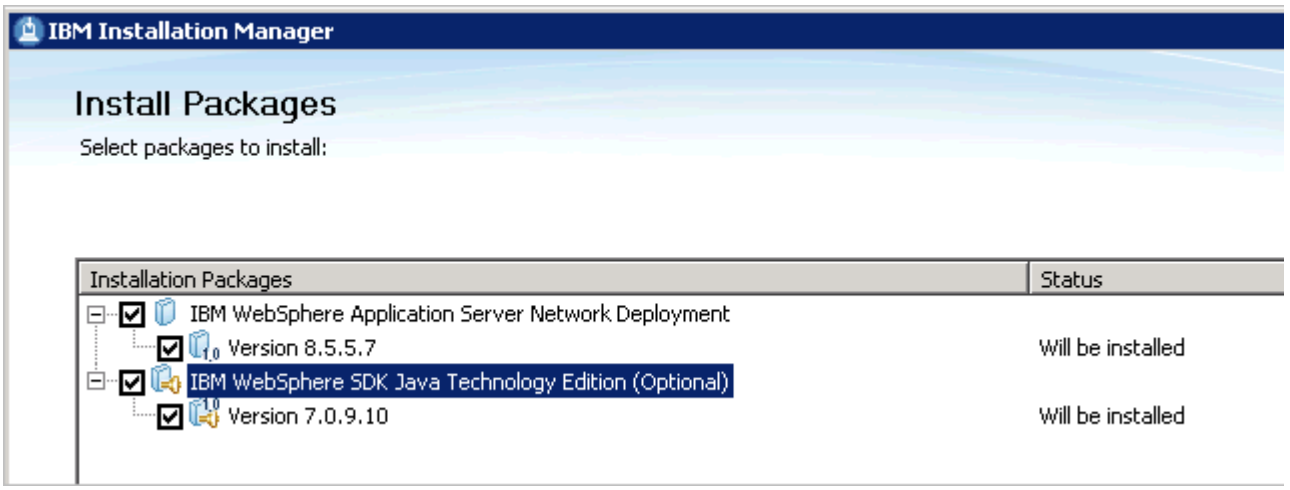
If the **Connection** column shows errors, check the repository locations.

__ 3. Click **OK**.

__ 4. Select the installation packages for WebSphere Application Server and the WebSphere SDK, and accept the license agreement.

__ a. In the main IBM Installation Manager window, click **Install**.

__ b. On the installation package selection page, select **IBM WebSphere Application Server Network Deployment** and **IBM WebSphere SDK Java Technology Edition (Optional)**.



__ c. Click **Next**.

__ d. On the licenses page, select **I accept the terms in the license agreement**, and click **Next**.

__ 5. Change the default directory for shared resources to: `C:\IBM\IMShared` and click **Next**.

__ 6. Change the location for the WebSphere Application Server installation to: `C:\IBM\WebSphere\AppServer`. and click **Next**.

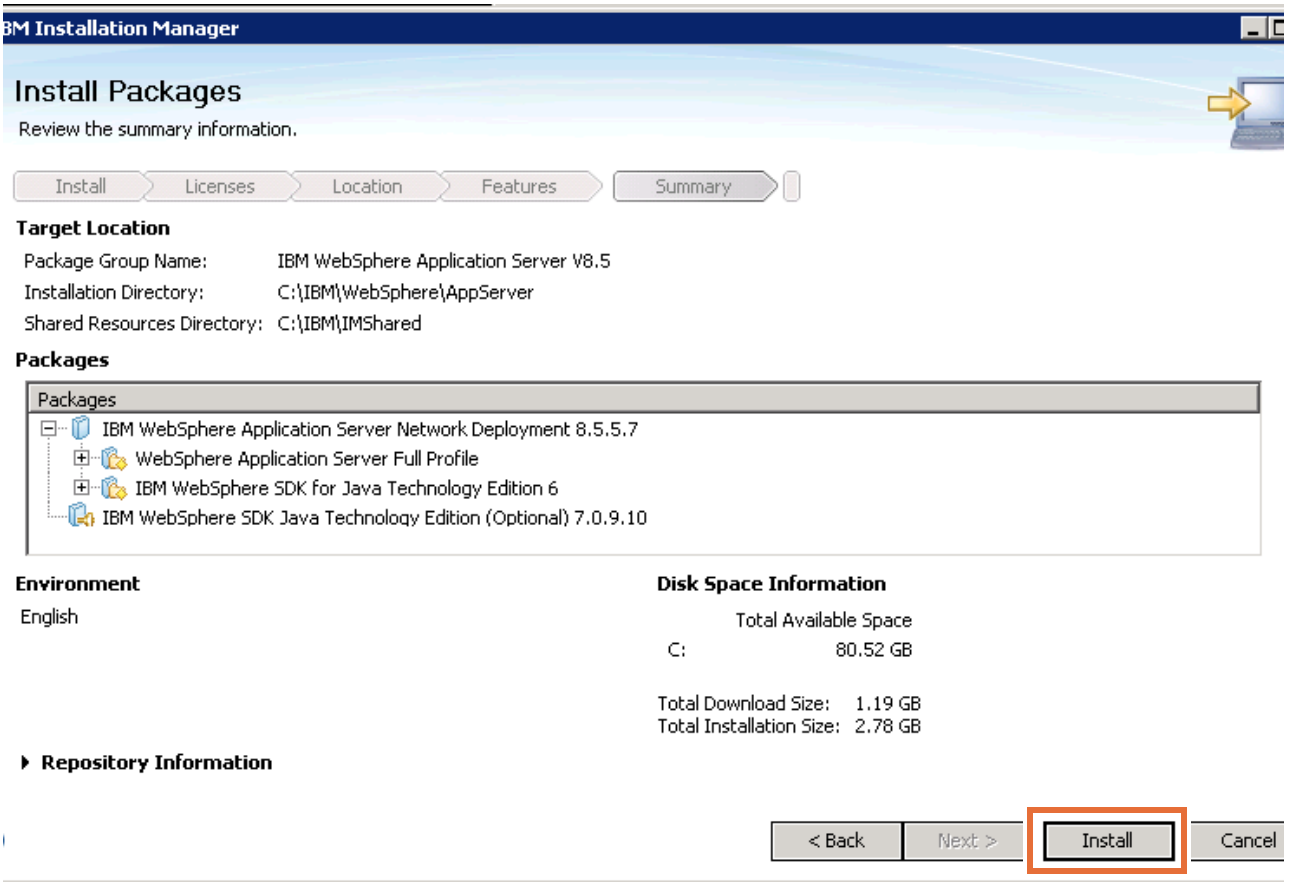
Important

Do **not** install WebSphere Application Server in the `Program Files` or `Program Files (x86)` directories because it might lead to user privilege limitations.

__ 7. On the translations page, keep the default translation option of **English** and click **Next**.

__ 8. On the features to install page, keep the default options selected and click **Next**.

__ 9. On the Summary page, click **Install**.



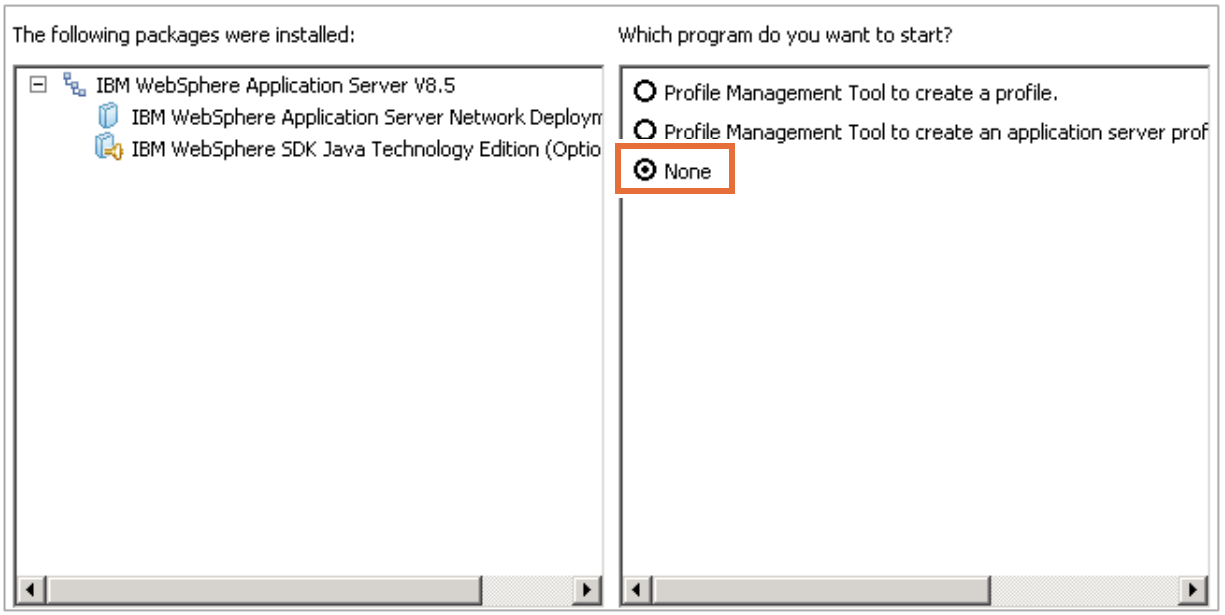
When the installation finishes, the Install Packages window shows that WebSphere Application Server Network Deployment and the WebSphere SDK Java Technology Edition are installed.

Information

When the installation is complete, the Install Packages window prompts you to use the Profile Management Tool to create a profile.

However, you do not need to create a profile because you create the ODM sample profile after you complete the ODM installation.

__ 10. On the right side of the Install Packages window, in the “Which program do you want to start” section, select **None**.



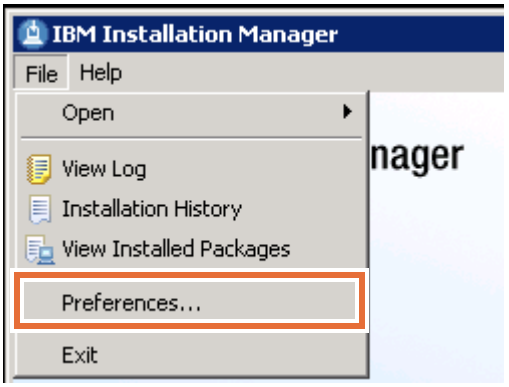
__ 11. Click **Finish**.

2.4. Installing Decision Server Insights

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

__ 1. Add the Operational Decision Manager Insights installer file as an IBM Installation Manager repository.

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



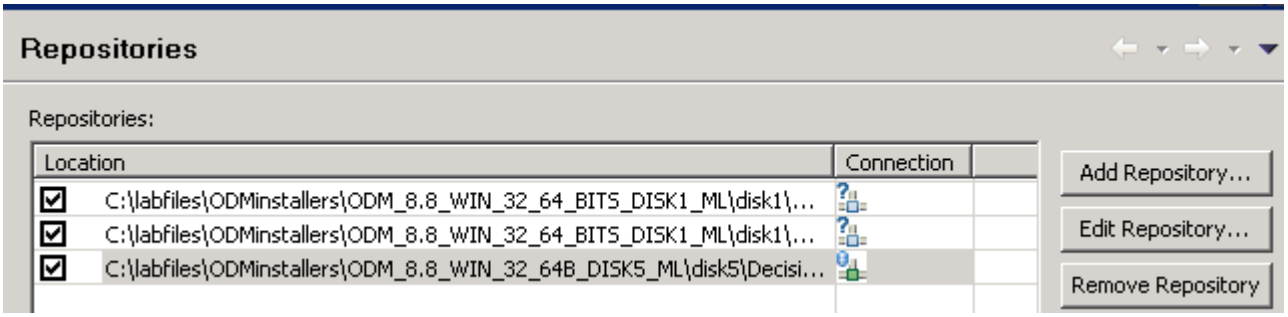
__ b. On the Repositories page, click **Add Repository**.

__ c. In the Add Repository window, click **Browse** and go to the ***<LabfilesDir>*** **> ODMinstallers > disk5 > DecisionServerInsights** folder.

__ d. Select **repository.config** and click **Open**.

__ e. In the Add Repository window, click **OK**.

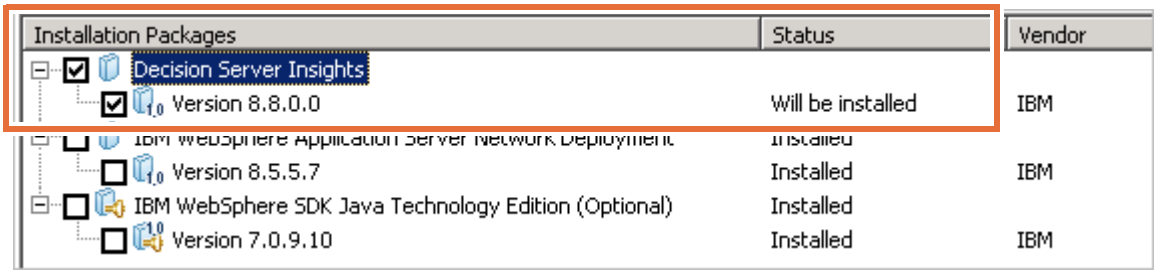
On the Repositories page, the Decision Server Insights installation repository is listed.



__ f. Click **OK** to close the Preferences window.

__ 2. In the IBM Installation Manager window, click **Install**.

__ 3. In the Installation Packages list, select **Decision Server Insights Version 8.8.0.0**, and click **Next**.



__ 4. In the Install Packages Licenses window, select **I accept the terms in the license agreement**, and click **Next**.

__ 5. In the Install Packages Location window, set the installation path.

__ a. Select **Create a new package group**.

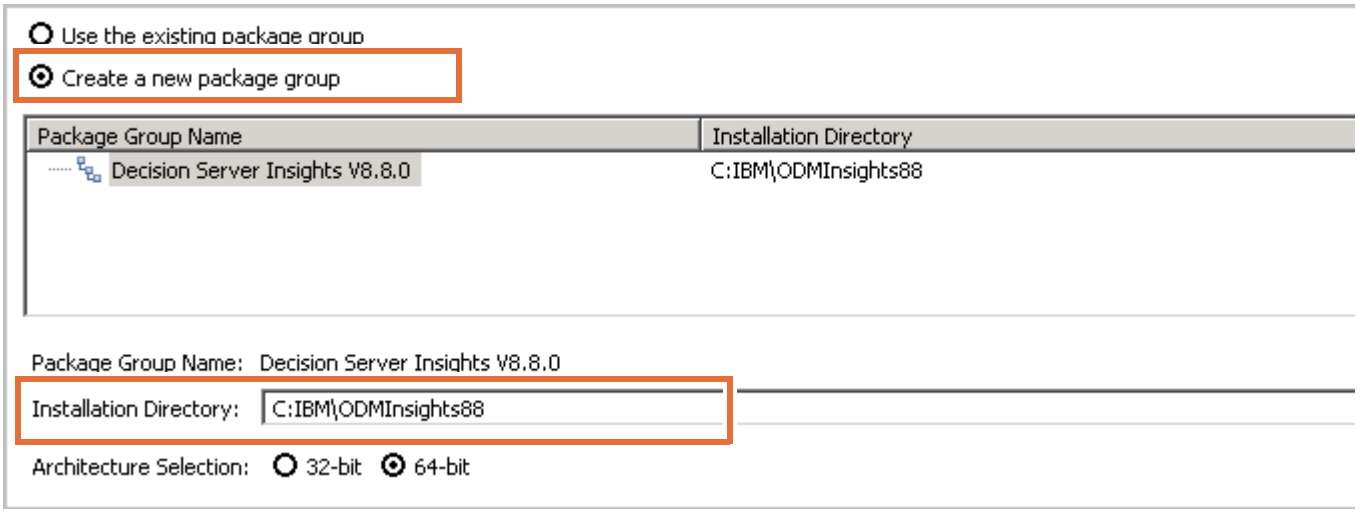
__ b. Change the directory path in the **Installation Directory** field to the following path:

`C:\IBM\ODMInsights88`

Important

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\ODM88` directory. Make sure that you install Decision Server Insights outside of `Program Files` and `Program Files (x86)` to avoid user privilege conflicts.



__ 6. Click **Next**.

__ 7. In the Install Packages “Select the translations to install” window, keep **English** as the selected language, and click **Next**.

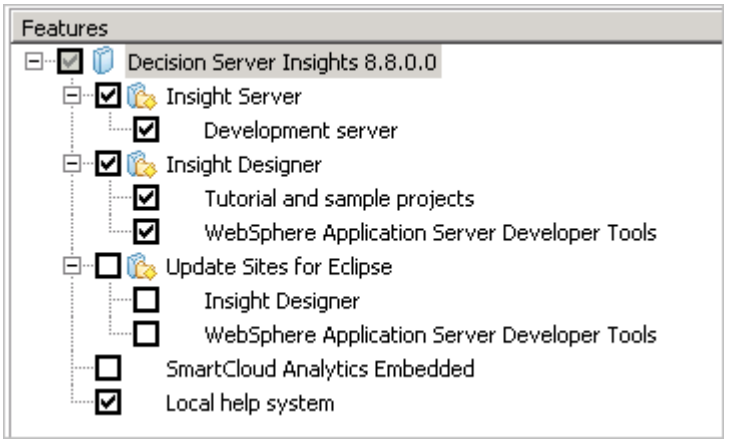
__ 8. In the Install Packages “Select the features to install” window:

__ a. Keep the default feature selections:

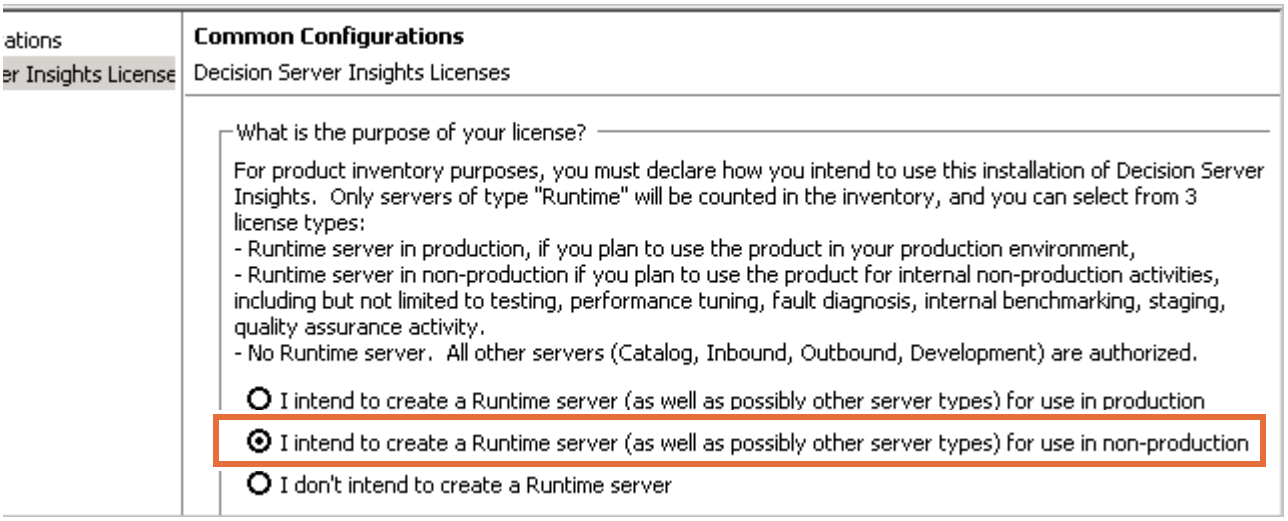
- **Insight Server: Development Server**
- **Insight Designer: Tutorial and sample projects, WebSphere Application Server Developer Tools**

__ b. Select **Local help system**.

__ c. Click **Next**.



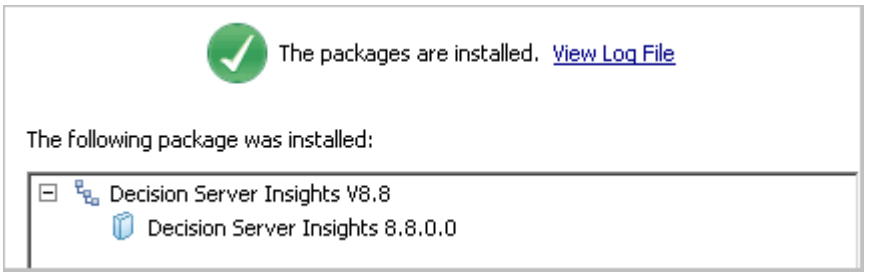
__ 9. In the Install Packages “Fill in the configurations for the packages” window, select **I intend to create a Runtime server (as well as possibly other server types) for use in non-production**, and click **Next**.



__ 10. In the Install Packages “Review the summary information” window, click **Install**.

The Decision Server Insights installation can take up to 15 minutes to complete.

When the installation is finished, you see a confirmation message that Decision Server Insights is installed.



__ 11. Click **Finish** to exit the Installation Summary window.

__ 12. Close IBM Installation Manager.

Section 3. Exploring your Decision Server Insights installation

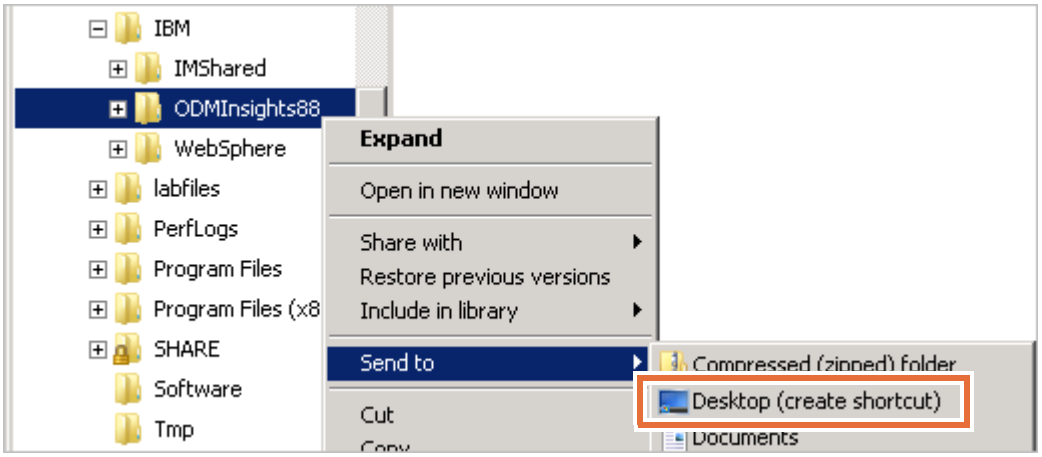
__ 1. In Windows Explorer, open the ***<InstallDir>*** (`C:\IBM\ODMInsights88`) directory.

Note

For this course, Decision Server Insights is installed in the `C:\IBM\ODMInsights88` directory.

__ 2. Create a desktop shortcut to the installation directory.

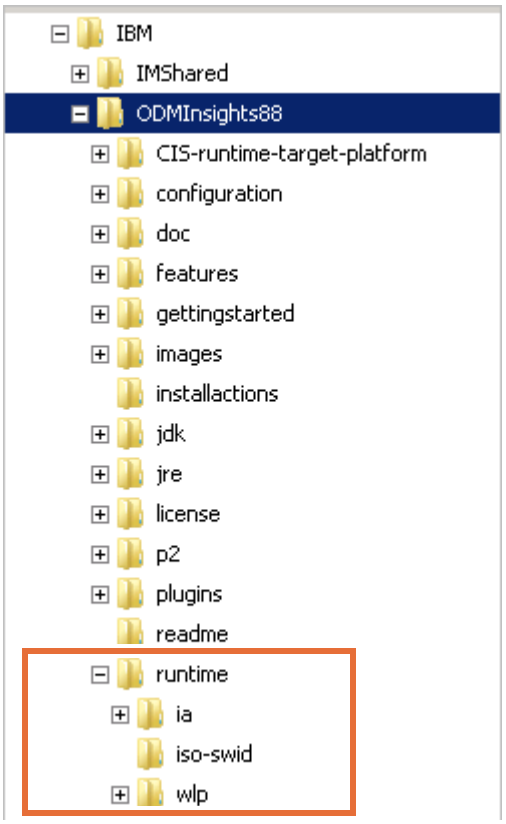
__ a. Right-click **ODMInsights88** and click **Send to > Desktop (create shortcut)**.



__ 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.

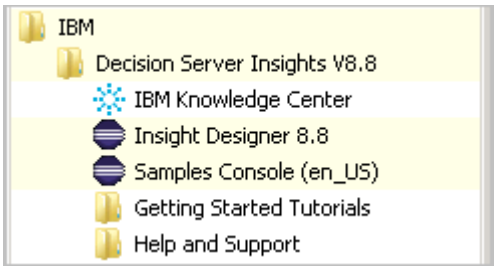
__ 3. Expand the `ODMInsights88` folder to view the installed files and folders, and notice the **runtime** folder.



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

__ 4. Close Windows Explorer.

__ 5. From the **Start** menu, click **All Programs > IBM > Decision Server Insights V8.8** and notice the Decision Server Insight menu options.



__ 6. Right-click **Insight Designer 8.8** and click **Send to > Desktop (create shortcut)**.

Throughout the labs, you work with **Insight Designer 8.8**. You can use this shortcut for quick access to the tool.

Section 4. 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.

4.1. Opening Insight Designer

__ 1. Open Insight Designer by either clicking **Insight Designer 8.8** in the **Start** menu (which should still be open) or by double-clicking your new **Insight Designer 8.8** 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:

<***LabfilesDir***>\workspaces\start

Where <***LabfilesDir***> refers to `C:\labfiles`.

__ b. Click **OK**.

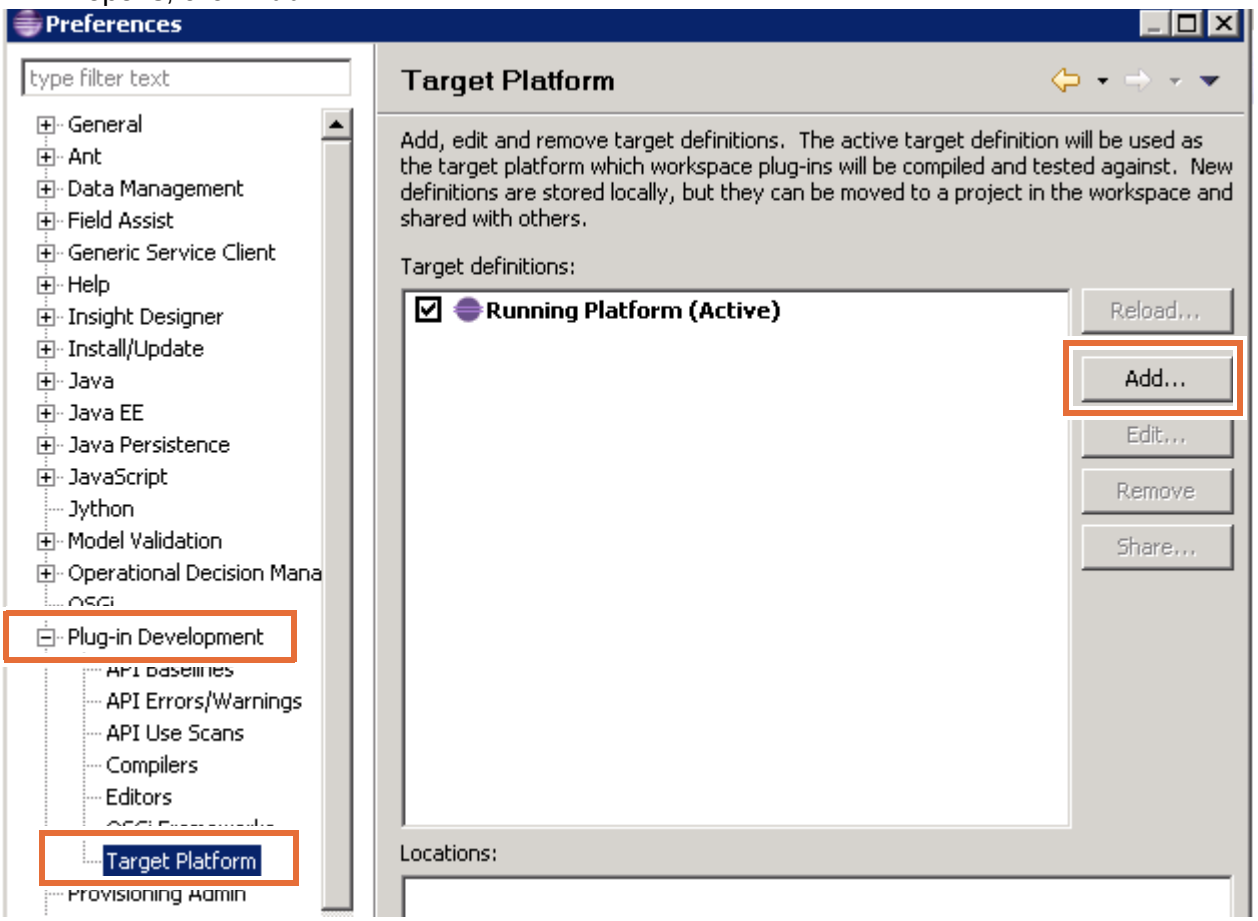
__ c. When the workspace opens, in the **Welcome** tab, click **Workbench**.

By default, Insight Designer opens in the Decision Insight perspective.

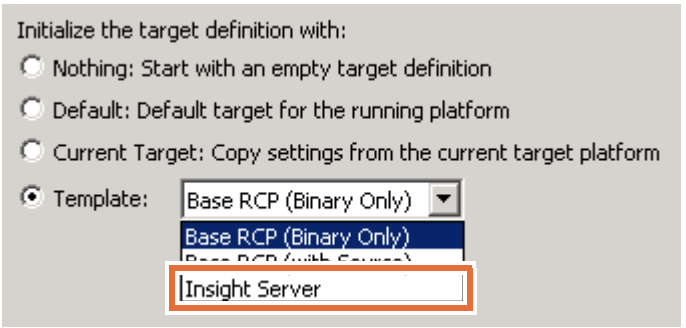
4.2. Setting the target platform

__ 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 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**.

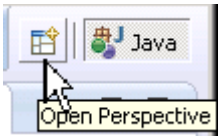
Note

As you learn during the labs, you can set the Target Platform automatically for imported projects.

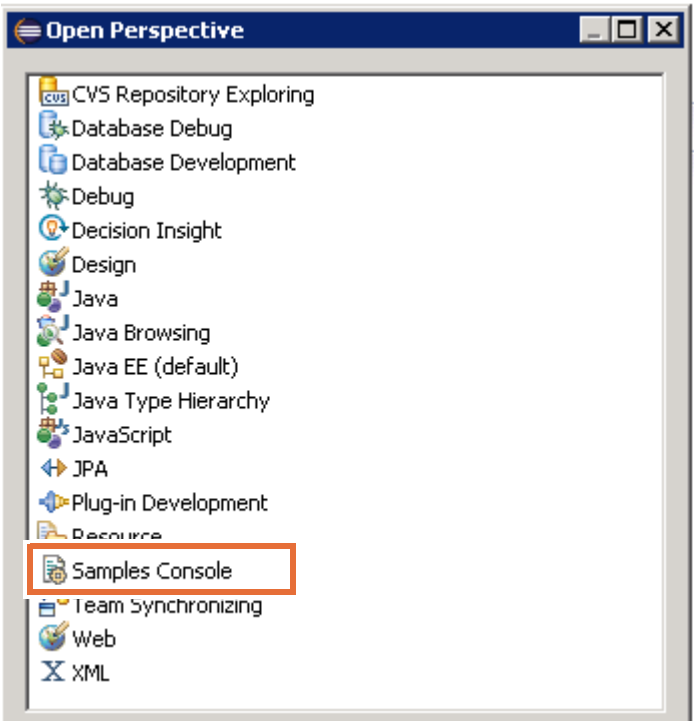
Section 5. Starting the server from Insight Designer

__ 1. Switch to the Samples Console perspective.

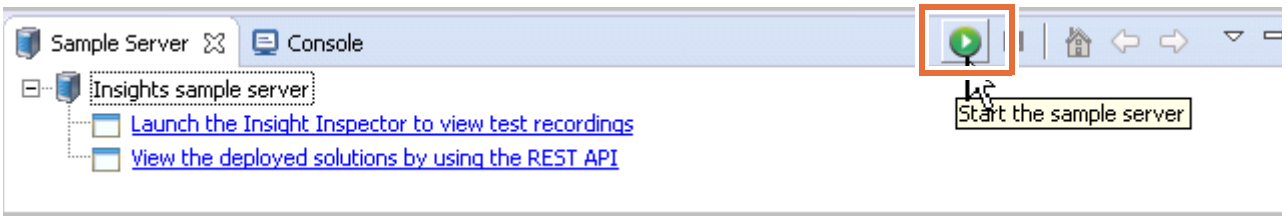
__ a. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.



__ b. In the Open Perspective window, 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 6. 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 the `propertyManager` utility to set the debug port to 6543.

Note

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

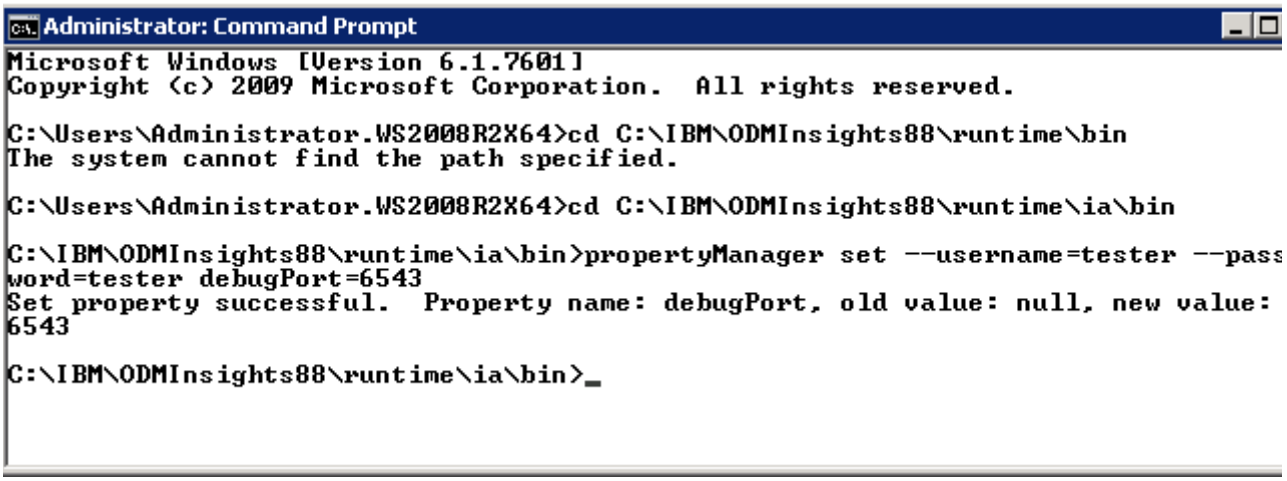
The `propertyManager` utility is a Decision Server Insights general administration script that runs from the `InstallDir/runtime/ia/bin` directory.

__ a. Open a command prompt window, go to the `C:\IBM\ODMInsights88\runtime\ia\bin` directory, and press Enter:

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ b. Type the following command, and press Enter.

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



__ 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\ODMInsights88`

The Liberty profile is managed through the `C:\IBM\ODMInsights88\runtime\wlp` directory.

__ b. Open the `server.xml` file in a text editor and scroll to the last entry in 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.

__ 4. Keep Insight Designer open for the next exercise.

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

This exercise looked at the installation of Decision Server Insights.

Exercise 2. Creating a solution in Insight Designer

Estimated time

00:15

Overview

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

Objectives

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. When prompted for a workspace, browse to:

<LabfilesDir>\workspaces\start

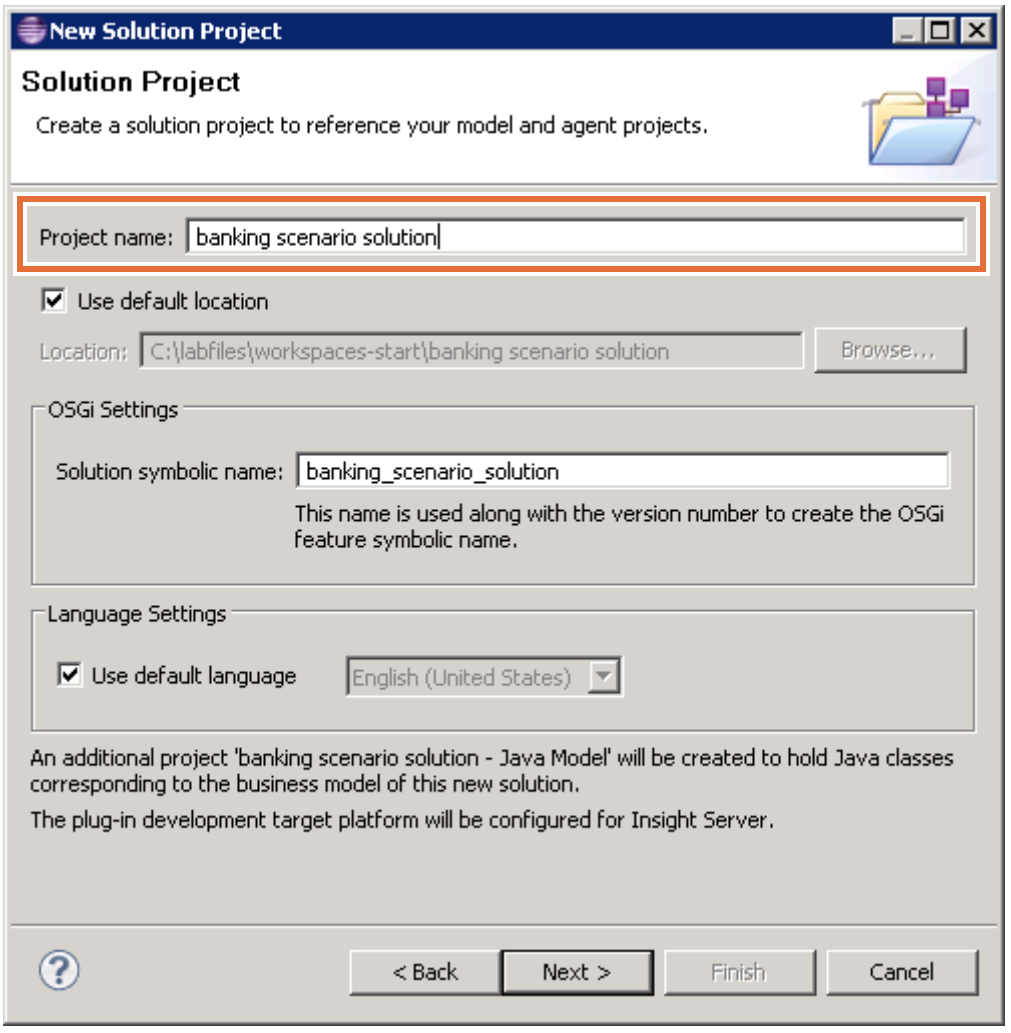
Where *<LabfilesDir>* refers to C:\labfiles.

In Insight Designer, you should be in the Decision Insight perspective (as indicated on the perspective toolbar).

__ 3. Create a solution project.

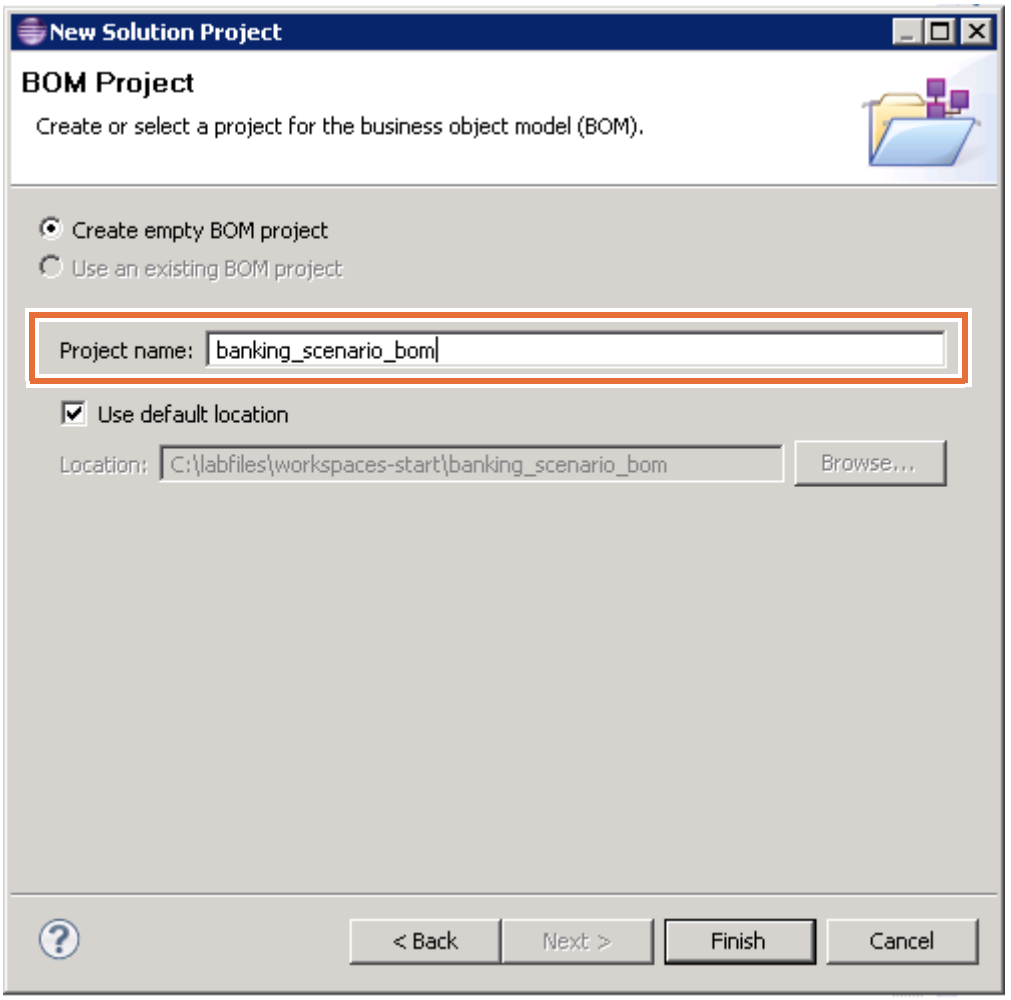
__ a. Click **File** > **New > Solution Project**.

__ b. In the **Project name** field, type: banking_scenario_solution



__ c. Click **Next**.

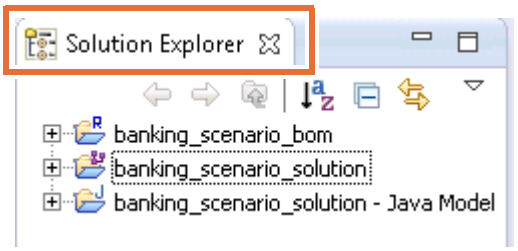
__ d. On the BOM Project page, type a project name for the BOM: `banking_scenario_bom`



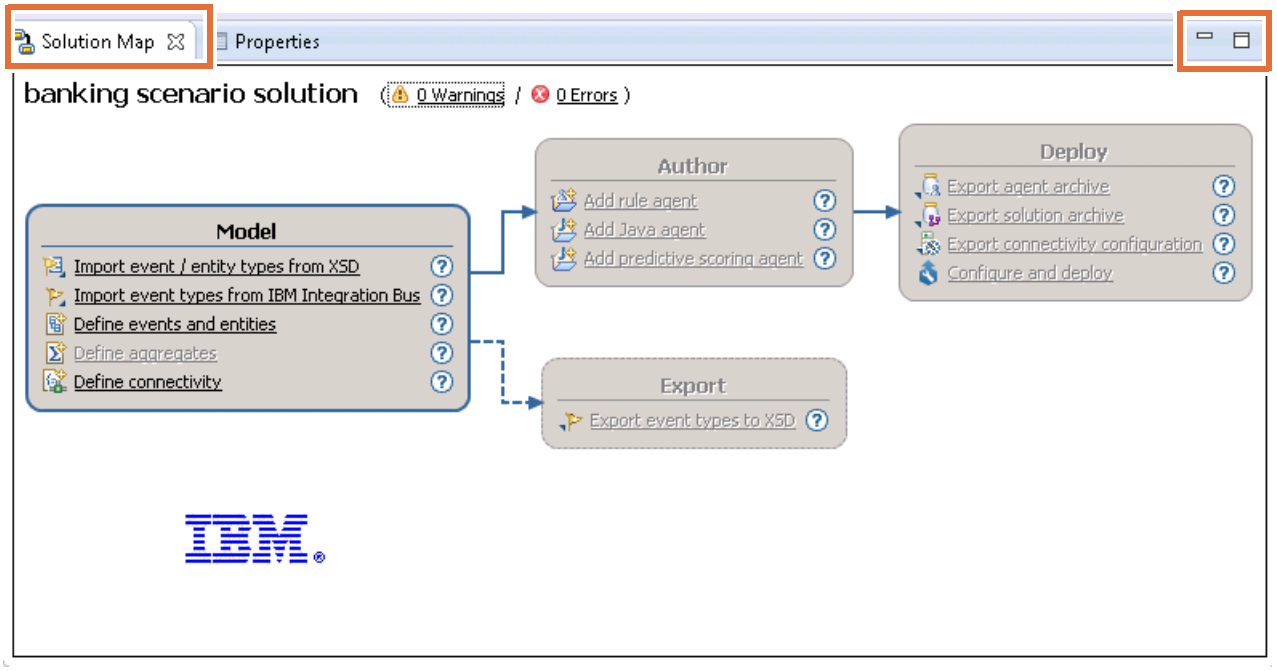
__ e. Click **Finish**.

You now have three projects in Solution Explorer.

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



__ 4. 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.

__ 5. Keep Insight Designer open for the next exercise.

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

Estimated time

00:30

Overview

This exercise covers how to create a business model.

Objectives

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"

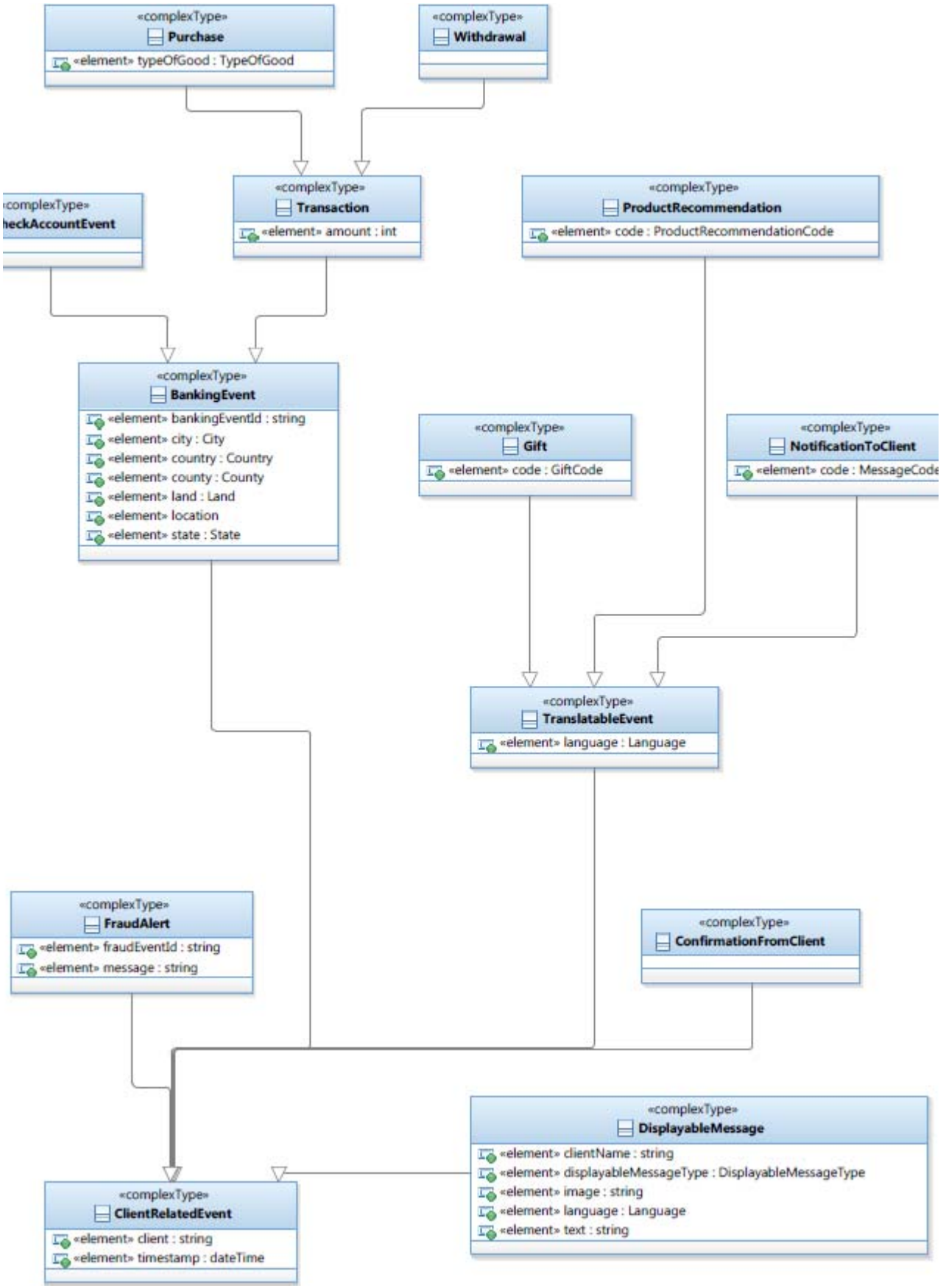
Requirements

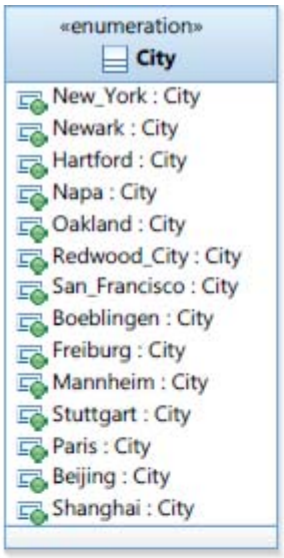
This exercise requires that you continue in the workspace that you created in Exercise 1, "Getting started with Decision Server Insights".

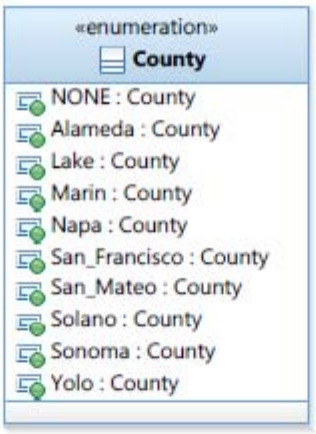
Section 1. Modeling the domain

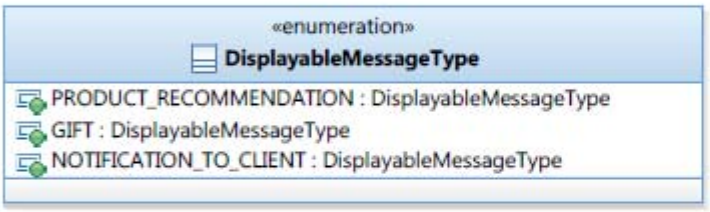
Before you can define how to process events, you must first define which events to monitor, which entities are involved, and the relationships between entities and events. Modeling the entities in your domain is key to creating a complete business model definition.

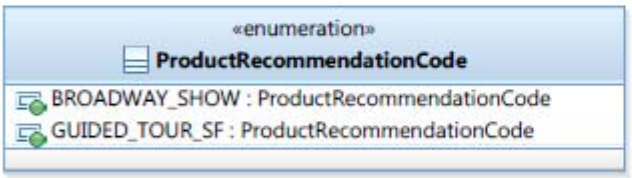
__ 1. Look at the business model diagrams that are shown here.

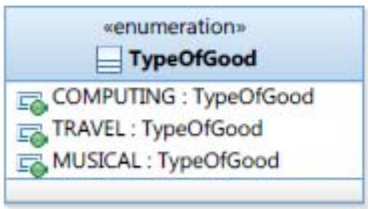


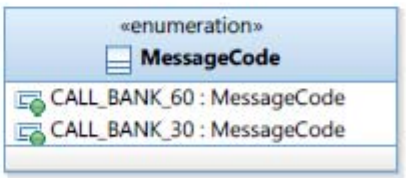


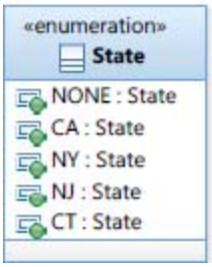


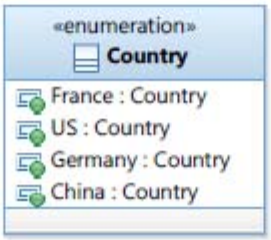


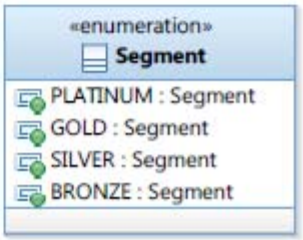




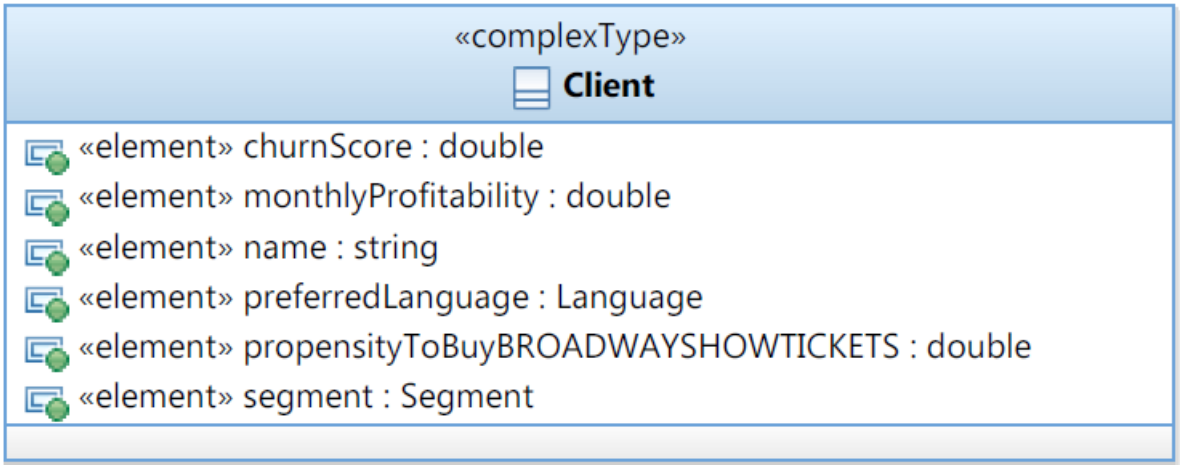




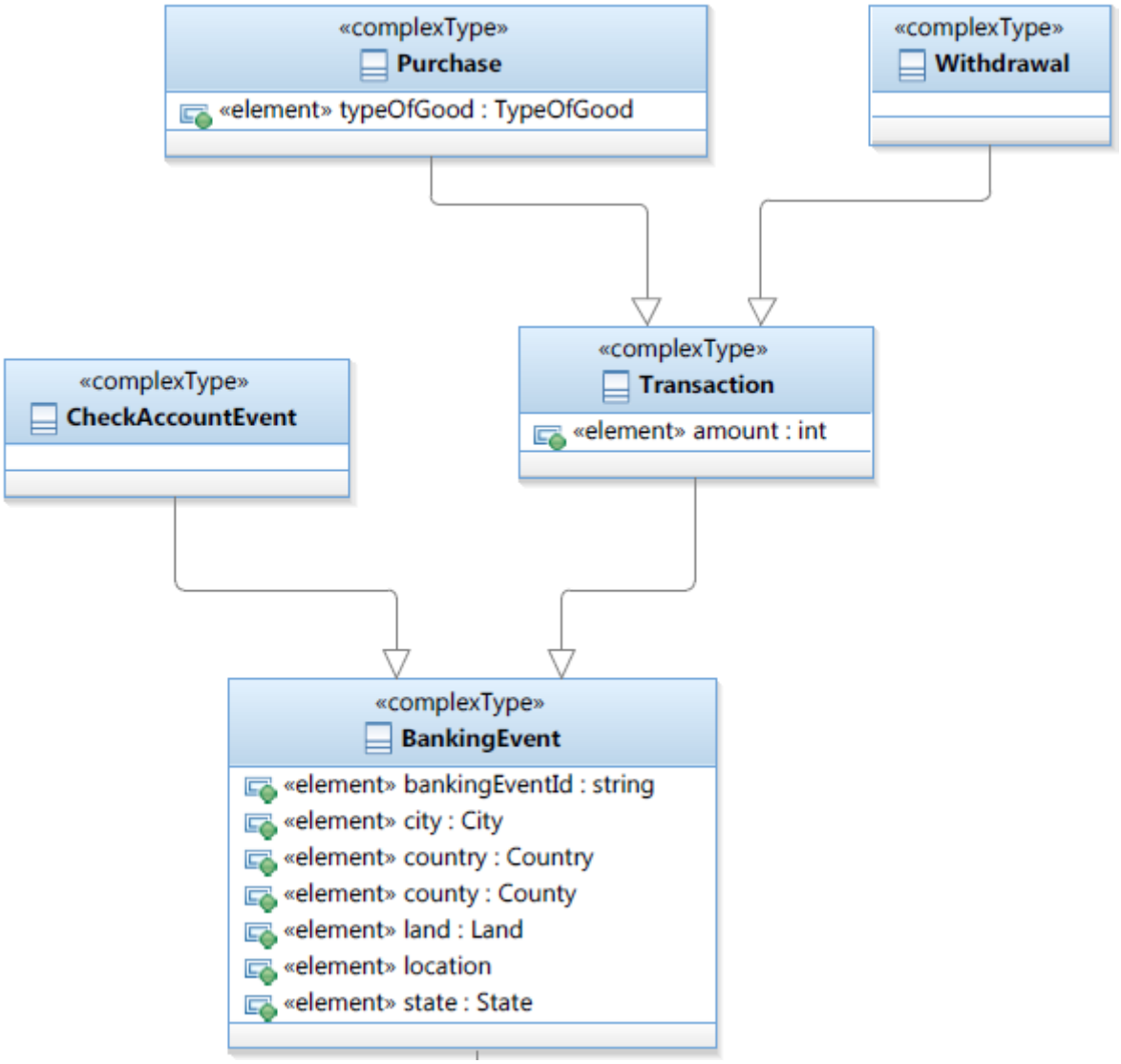




__ 2. Look at the `Client` class diagram, and note the attributes and types.



__ 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

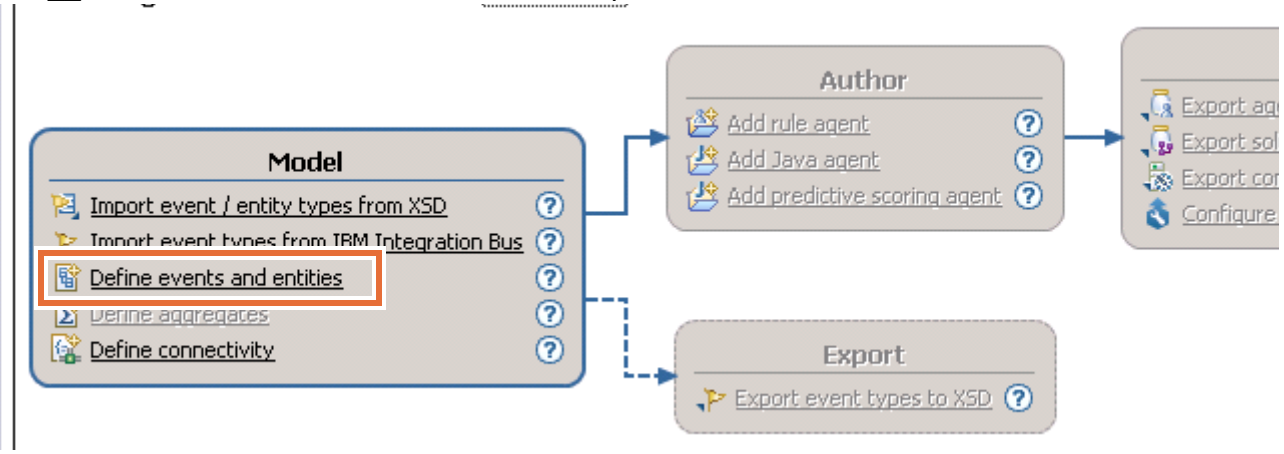
Next, you transfer the information from the domain models to a business model definition (BMD) file.

__ 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 **Package** field of the New Business Model Definition wizard, 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.

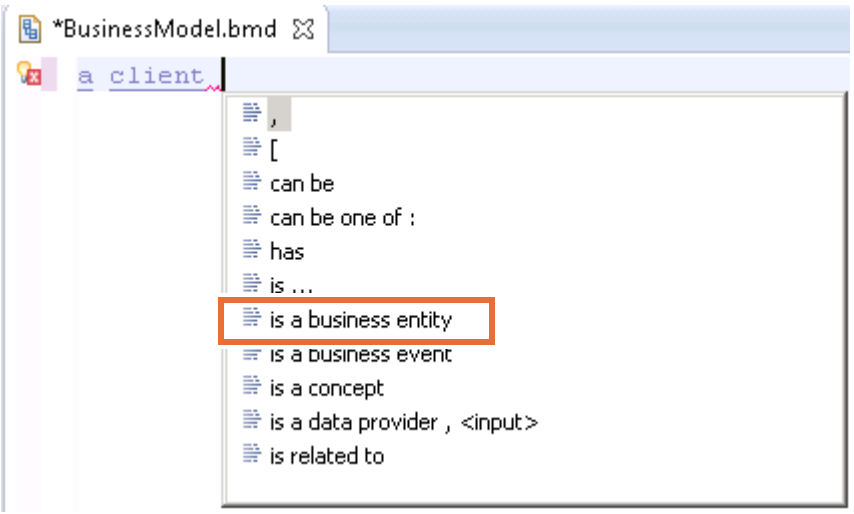
__ 3. Create an entity definition.

__ a. Look again at the Client object in the UML diagram and its attributes:

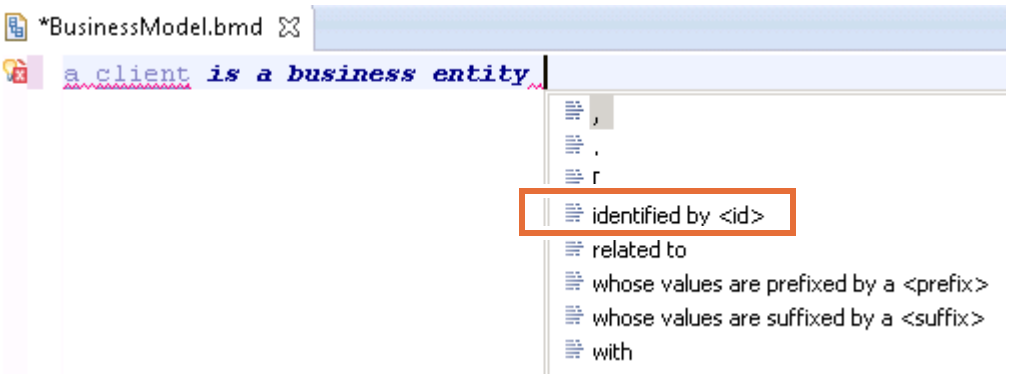
- `name`
- `segment`
- `churn score`
- `monthly profitability`
- `propensity to buy BROADWAYSHOWTICKETS`
- `preferred language`

__ b. In the BMD editor, start the definition by typing: `a client`

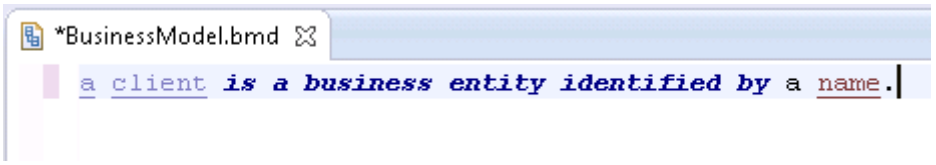
__ c. When you press Space and the editor prompts you for the next part of the statement, double-click **is a business entity**.



__ 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. Press Enter to start a new line.

__ g. Add an attribute to the `Client` entity by typing this line:

```
a client has a segment.
```

By default, attributes are of type String.

__ h. **Optional.** You can complete the `Client` by adding the remaining attributes:

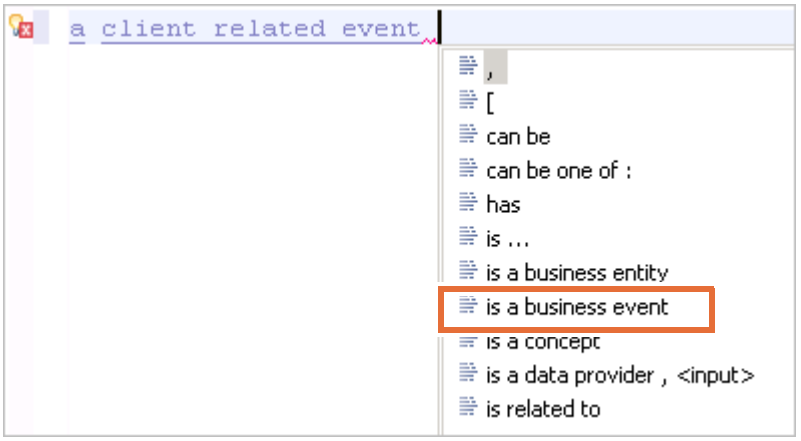
- `churn score`
- `monthly profitability`
- `propensity to buy BROADWAY_SHOW_TICKETS`
- `preferred language`

__ 4. Define an event.

__ a. Look again at the UML diagram, and note `BankingEvent`, which inherits from the `ClientRelatedEvent`, and the event attributes.

__ b. On a new line, 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**.

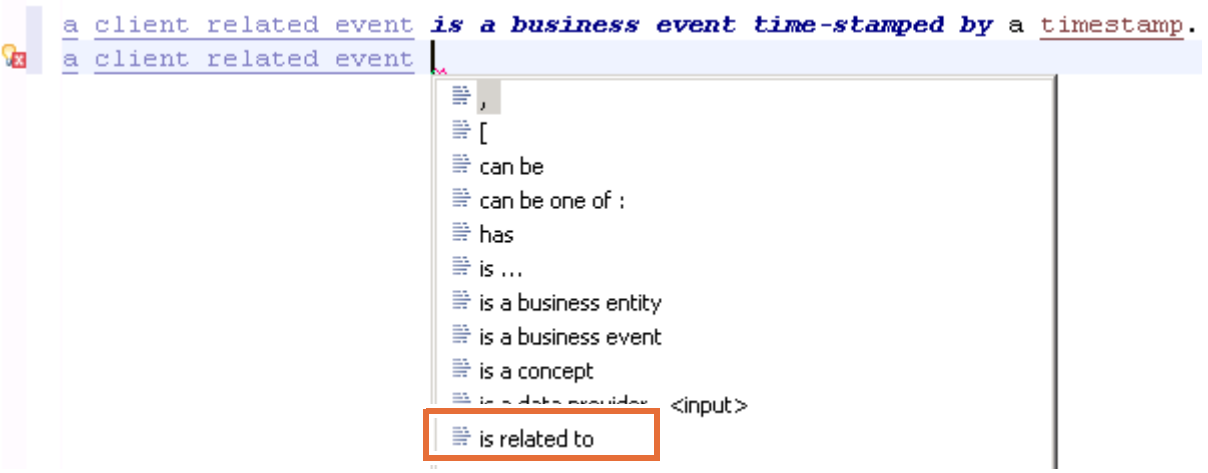


__ d. Next, choose **time-stamped by <id>**.

__ e. Type: `a timestamp.`

__ f. Press Enter to start a new line.

__ g. Define the relationship between the client-related event and the client by using the “is related to” construct.



__ h. Complete the statement by typing: `a client.`

Questions

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

__ i. On a new line, define `BankingEvent` by using the “is a” construct, and typing this line:

```
a banking event is a client related event.
```

Hint

To complete the model, you use a predefined model that is provided for you. The complete business model is defined in the `bmd.txt` file of the <*LabfilesDir*>**code** folder.

__ 5. To complete the business model, copy the content from the `BMD.txt` file in the ***<LabfilesDir>*\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

Estimated time

00:30

Overview

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.

Objectives

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:

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

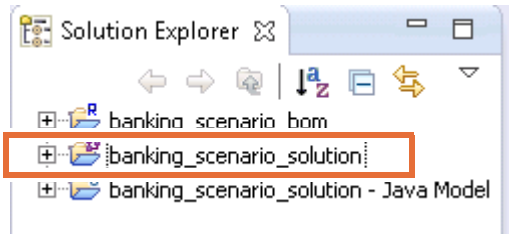
The rule must:

- 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

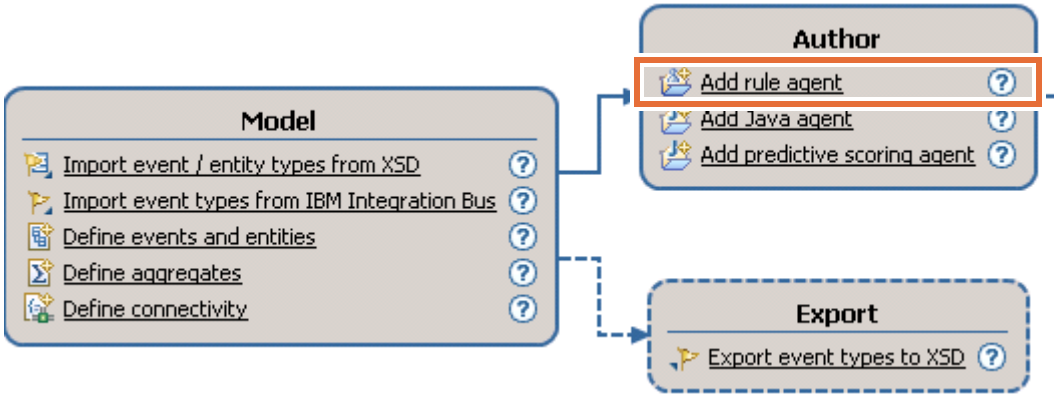
2.1. Creating the product recommendation rule agent

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 the `Client` entity.

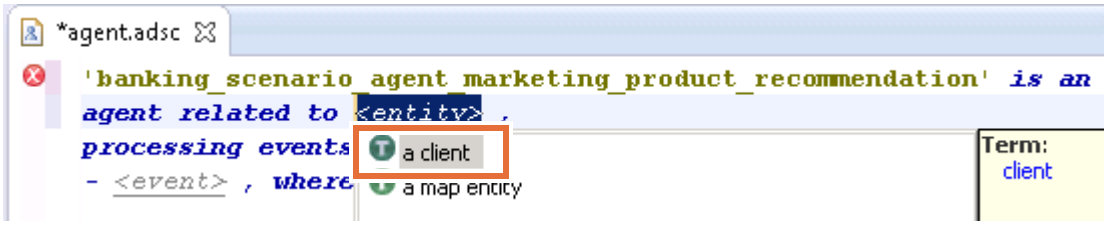
When the Decision Server Insights runtime environment receives any banking event for a client instance, a component of the environment, called the gateway, routes the event to the appropriate agents for that client. Each agent that subscribes to that event evaluates the rules against that event, along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

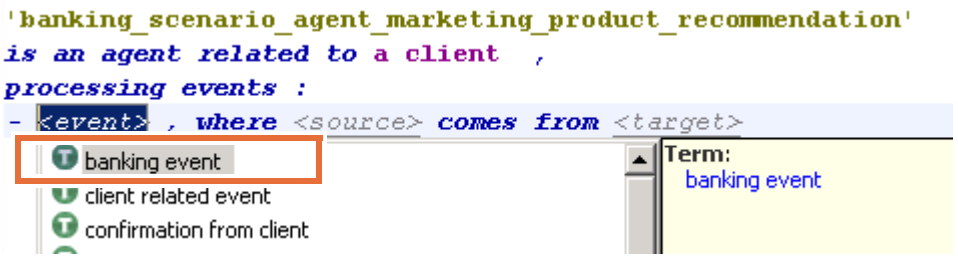
__ 1. Complete the descriptor to match this text:

```
'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
```

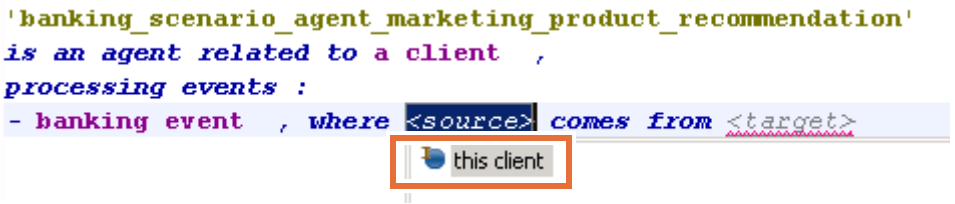
__ a. Click **entity**, and double-click **a client**.



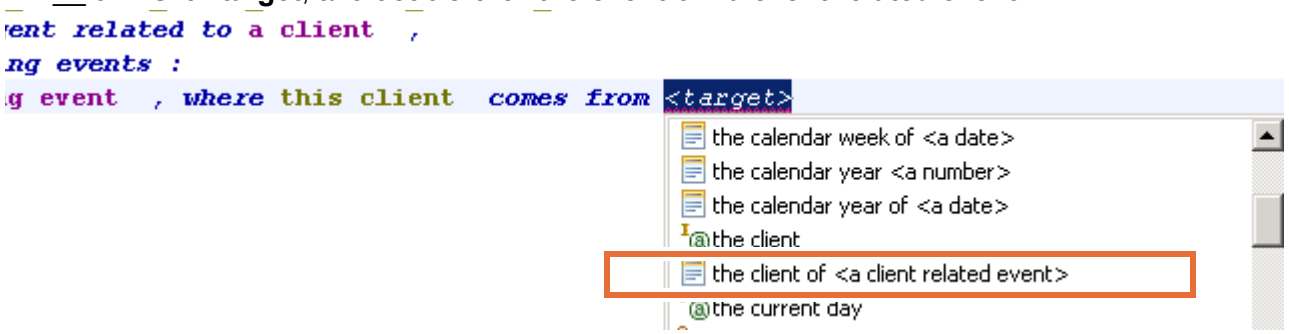
__ b. Click **event**, and double-click **banking event**.



__ c. Click **source**, and double-click **this client**.



__ d. Click **target**, and double-click **the client of <a client related event>**.



__ e. Click **a client related event**, and double-click **this banking event**.

Hint

You can copy and paste the agent descriptor text from the `'banking_scenario_agent_product_recommendation'` definition section of the `rule-agents.txt` file in the ***<LabfilesDir>*\code** directory.

__ 2. Complete the descriptor by copying the complete descriptor definition from the `rule-agents.txt` file and pasting it in the agent editor to overwrite the definition.

__ a. Go to the ***<LabfilesDir>*\code** directory.

__ b. Open the `rule-agents.txt` file and copy the `'banking_scenario_agent_product_recommendation'` definition section (highlight the agent description, and then press Ctrl+C).

__ c. In the agent editor, press Ctrl+A, and then press Ctrl+V.

Hint

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

__ 3. Save the `agent.adsc` file (Ctrl+S) and close it.

Section 3. Writing the San Francisco rule

Requirements

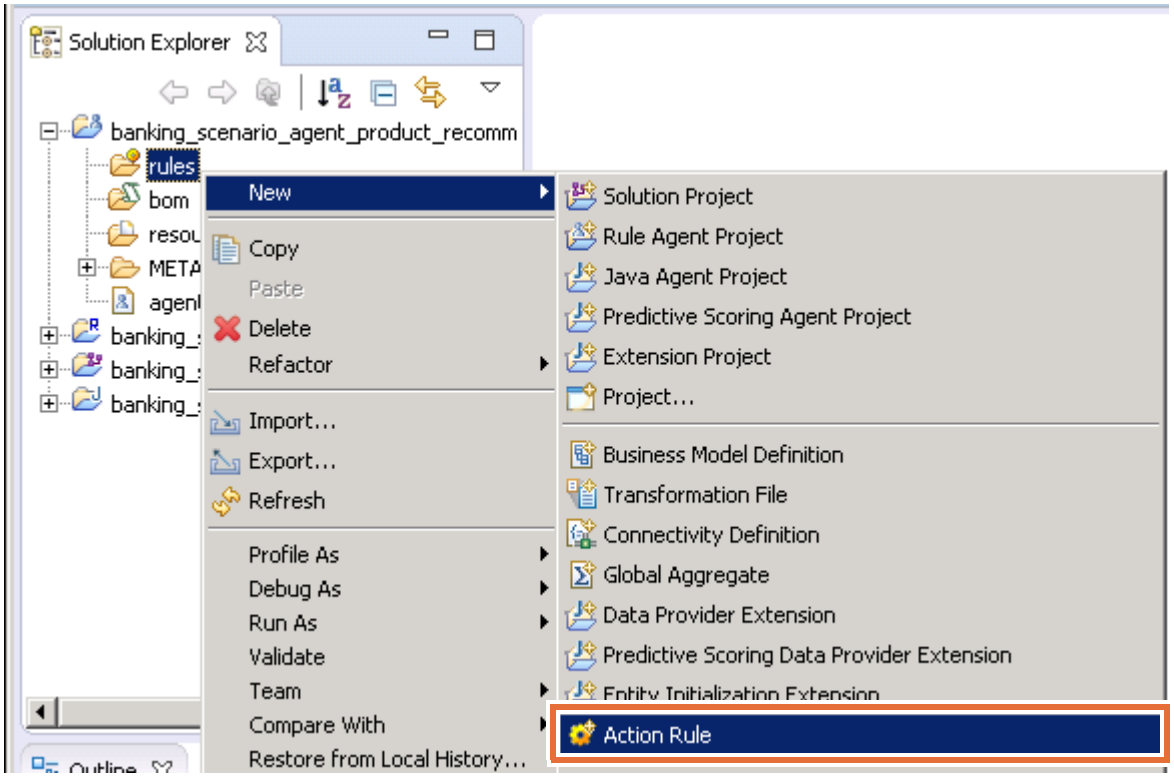
The rule is triggered by an incoming event when clients access their accounts while they are in California. Your rule conditions also test that the client did not already receive this same recommendation recently.

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.

__ b. Right-click the **rules** folder and click **New > Action Rule**.



__ c. In the **Name** field of the New Action Rule wizard, type: `Recommend guided tour of San Francisco`

__ d. Click **Finish**.

The new rule opens in the rule editor.

__ 2. In the rule editor, 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 `rule-agents.txt` file in the ***<LabfilesDir>*\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

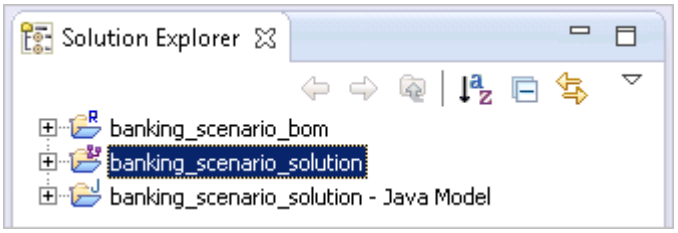
The product recommendation agent emits an event to send a message to the client's mobile device. However, `Client` entities have a preferred language attribute. To personalize the recommendations to use in the client's language, some post-processing is required before messages can be sent.

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 in the client's language.

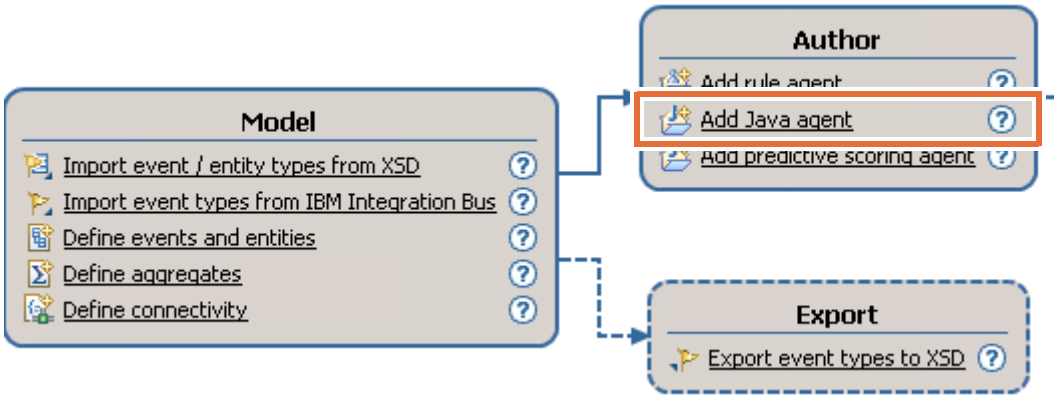
This multi-agent architecture enables separation of logic:

- 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.Translat
ionAgent' 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 Descriptor section of the `java-agent.xml` file in the ***<LabfilesDir>*\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 ***<LabfilesDir>*\code** directory is open.

__ d. Copy the import statements from the **Import statements to add** section of the `java-agent.xml` file to the import section of the `TranslateAgent.java` file.

__ e. 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

    }

}
```

__ f. Save your work to make sure that you do not have compilation errors.

__ g. 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

Estimated time

00:30

Overview

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

Objectives

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:

Requirements

This exercise requires that you switch workspaces to use the ***<LabfilesDir>*\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.

1.1. Creating the workspace

__ 1. If you closed Insight Designer after the previous exercise, open it from the **Start** menu by clicking **All Programs > IBM > Decision Server Insights V8.8 > Insight Designer 8.8**.

__ 2. If Insight Designer is still open from the previous exercise, switch to a new workspace.

__ a. From the **File** menu, click **Switch Workspace > Other**.

__ b. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

`C:\labfiles\workspaces\newyork`

__ c. Click **OK** to close the Workspace Launcher.

__ 3. Close the **Welcome** view.

The Decision Insight perspective opens.

__ 4. Import the 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 archive file** and click **Browse**.

__ d. Go to the ***<LabfilesDir>*** and select the `workspace1-ny.zip` file and click **Open**.

__ e. Click **Finish**.

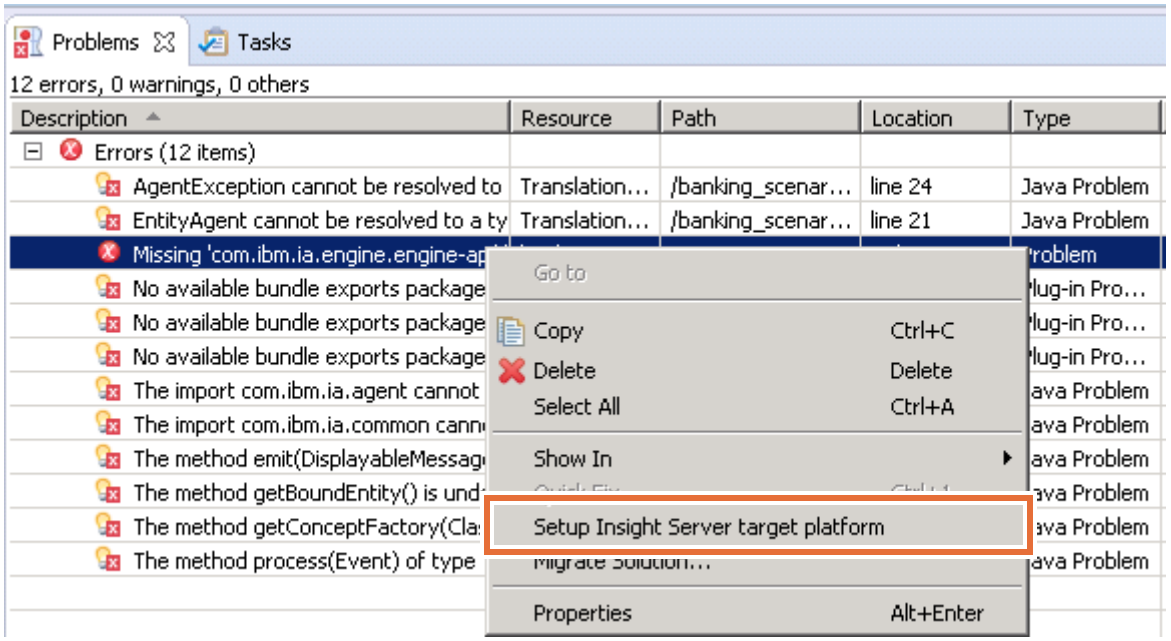
Your workspace now contains all the required projects. The projects might contain errors, but you resolve these errors during the next steps.

Troubleshooting

If you get a generation error window open, click **OK** to ignore it. Make sure that you wait for the workspace to build completely.

1.2. Setting the target platform

__ 1. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



__ 2. Wait for the project to rebuild completely.

You should not have errors after the workspace is rebuilt.

1.3. 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 the properties file 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).

The path should be:

`C:/IBM/ODMInsights88/runtime/wlp/usr/servers/cisDev/resources/security/key.jks`

__ b. Look for the `debugservers` property, and make sure that it is set to port 6543, as shown here:

`debugservers=localhost:6543`

__ 3. Close the `testdriver.properties` file.

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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** directory 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.

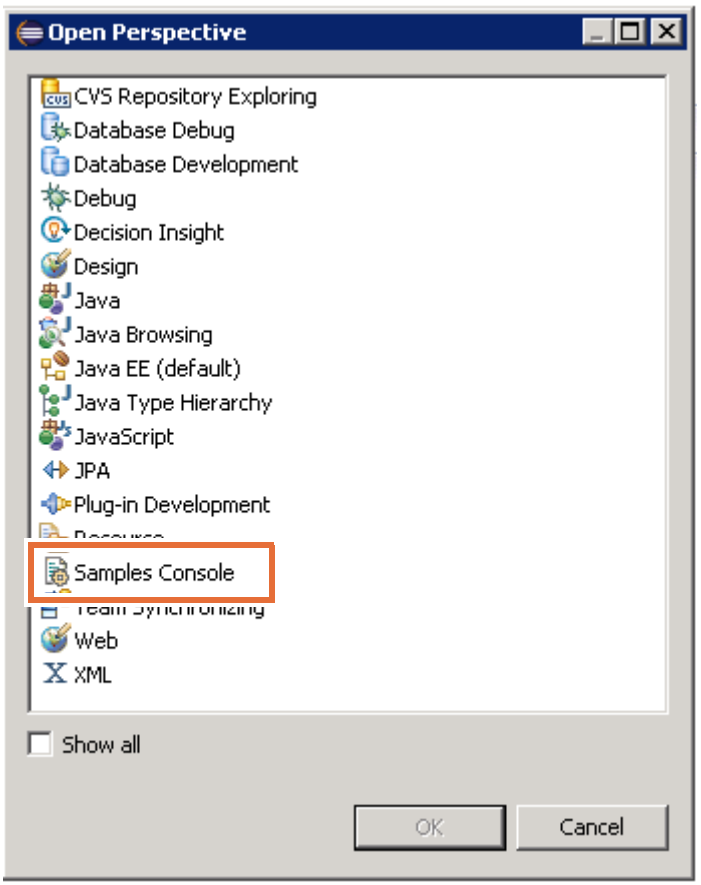
3.1. Starting the server

__ 1. Make sure that the server is started.

__ a. Switch to the Samples Console perspective.

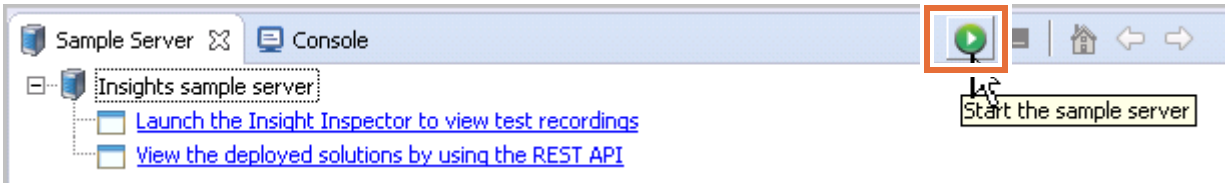
__ b. Click the **Open Perspective** icon in the upper-right corner of the Eclipse window.

__ c. In the Open Perspective window, select **Samples Console**.

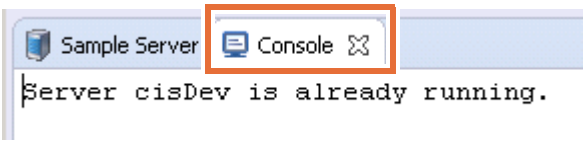


The Samples Console opens.

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



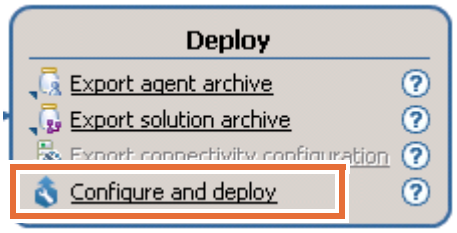
The default server is called cisDev. If the server is running when you click Start, the Console opens with the message that the server is already running.



3.2. Deploying the solution

__ 1. Switch back to the Decision Insight view.

__ 2. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.

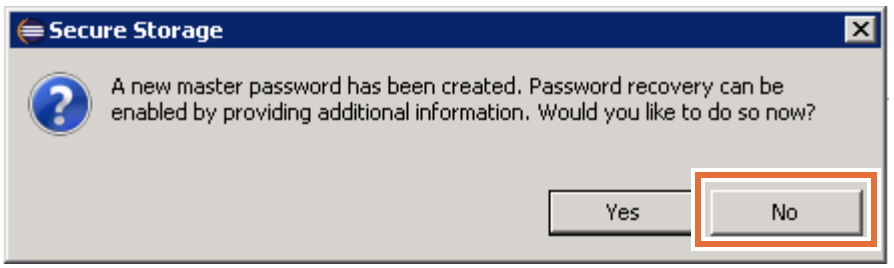


__ 3. In the **Deployment configuration name** field of the “Configure and Deploy” wizard, select **local** and click **Next**.

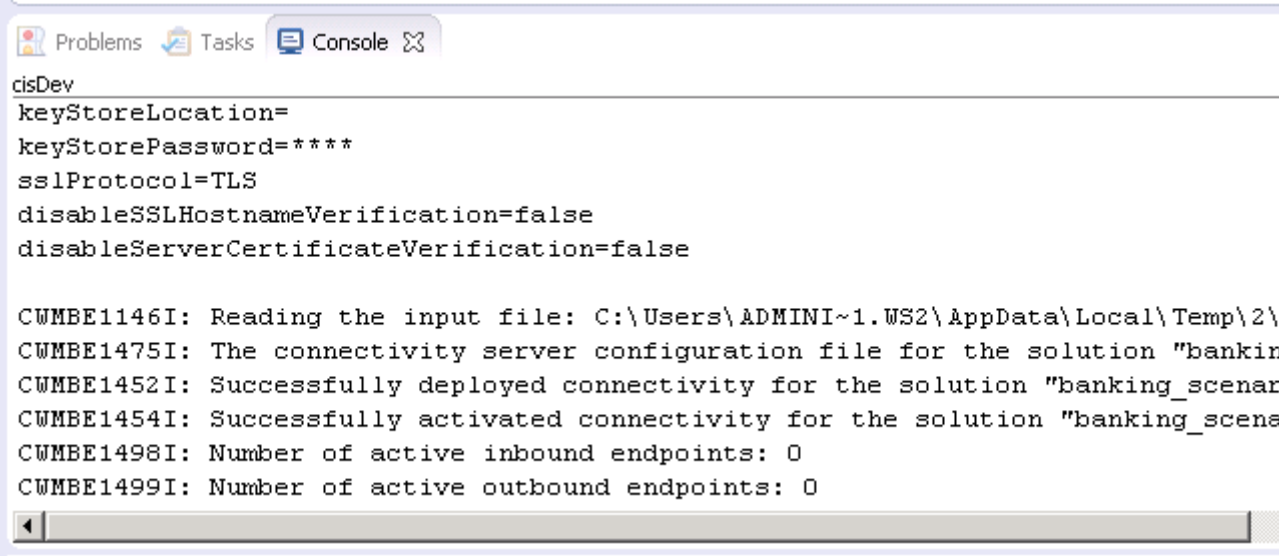
__ 4. Leave the default values and click **Finish**.

Note

If you see a Secure Storage window, click **No**.



Deployment takes a few moments. After deployment is complete, you see some messages that state that the solution and connectivity are successfully deployed.

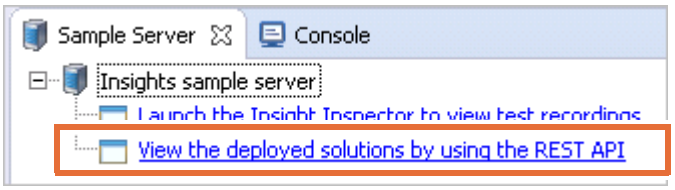


3.3. Verifying deployment

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.

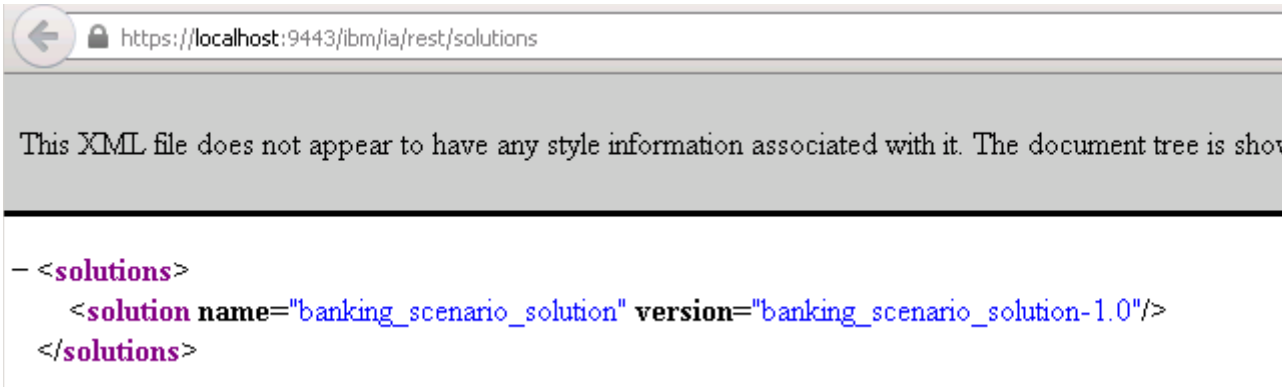
__ 2. Click **View the deployed solutions by using the REST API**.



__ 3. When the browser window opens, accept any security warnings and continue. For example, in Mozilla, click **I understand the Risks**, click **Add Exception**, and click **Confirm Security Exception**.

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

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



Note

The browser automatically switches to a secure connection that uses `https://localhost:9443`.

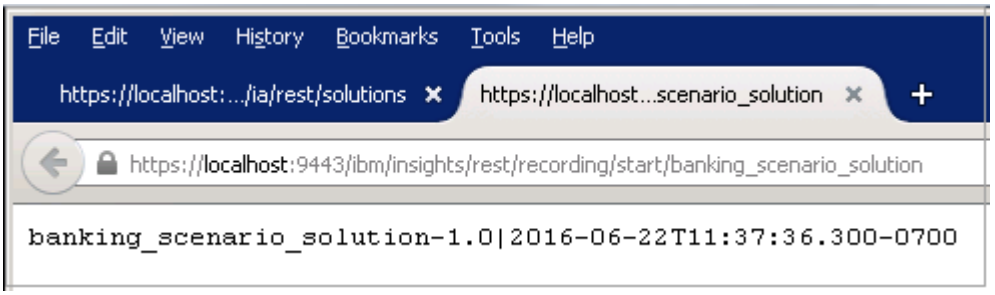
3.4. Preparing the recording of events in Insight Inspector

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:

`http://localhost:9080/ibm/insights/rest/recording/start/banking_scenario_solution`

The browser returns a message with the solution name and a time stamp. You use this time stamp as the recording ID to delete this recording after your tests are finished.



Note

The browser automatically switches to a secure connection that uses `https://localhost:9443`.

__ 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.

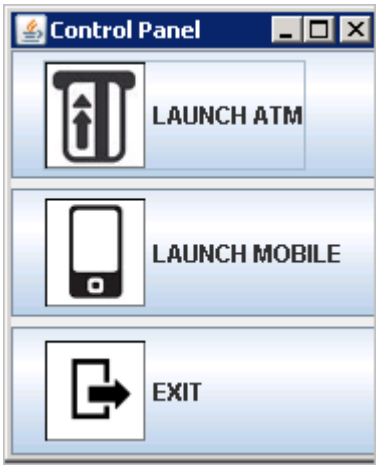
Section 4. Testing the solution

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 open.



Troubleshooting

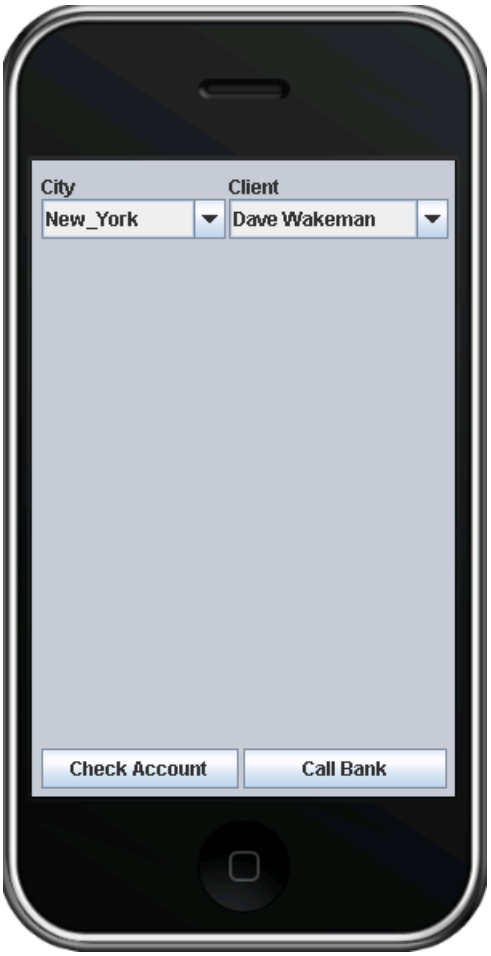
If you do not see the Control Panel, minimize all windows and open applications.

The Control Panel might open in the upper-left corner of the desktop.

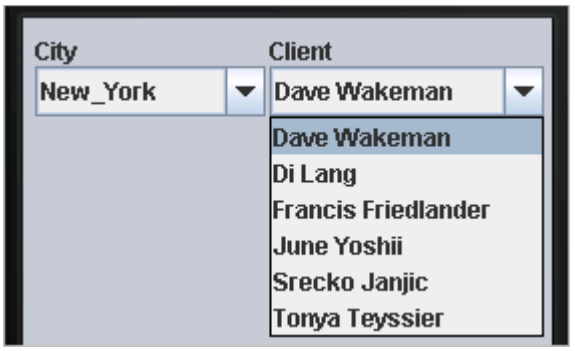
The Control Panel application is designed specifically for this lab. You can use it to launch pseudo-mobile devices and ATMs to interact with the solution.

__ 3. Click **LAUNCH MOBILE**.

A mobile device interface opens.



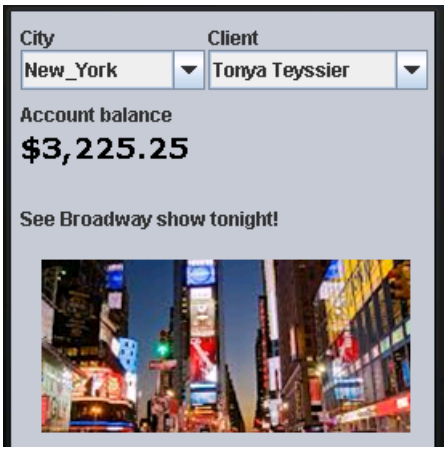
__ 4. Expand the **Client** list to see the list of clients that are available for this exercise.



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 runtime.

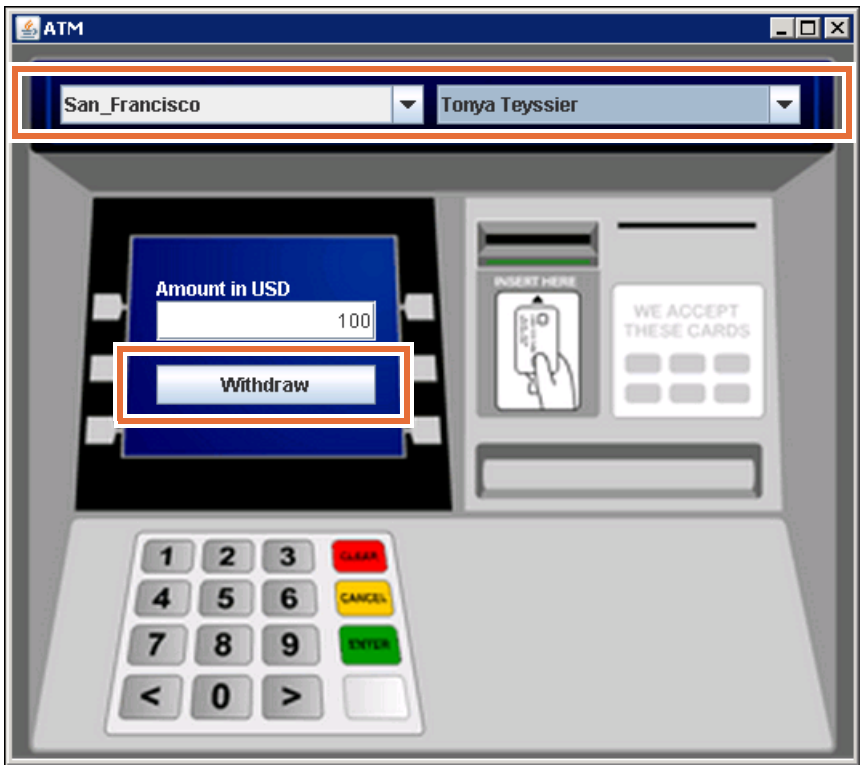
The check account event is a banking event that includes Tonya's ID. Thus, according to the statement in the .adsc file, it is sent to an instance of the banking_scenario_agent_product_recommendation rule agent that is bound to the Tonya entity. The event triggers the New York rule, which sends a product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

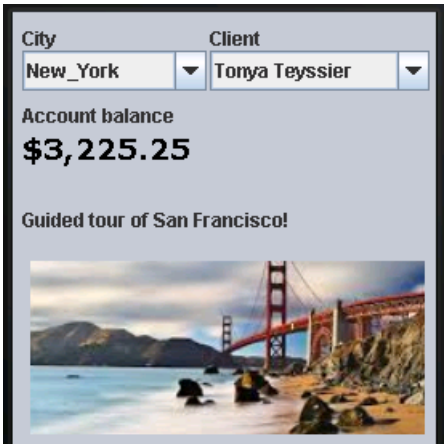
__ 6. Keep the mobile interface open and submit a withdrawal event from a different location.

__ 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**

__ b. Wait about 20 seconds and then 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.

__ c. Switch the ATM location back to **San_Francisco** and submit another withdraw event.

You should see the San Francisco message on the mobile interface.

__ d. Wait about 20 seconds, then 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 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**.

The messages that you see in the mobile interface should be in French.

__ 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**.

The messages that you see in the mobile interface should be 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.

Troubleshooting

If you do not see the messages as expected, you might need to pause between submitting events to allow the processes to complete.

__ 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. 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 runtime, 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_solution`

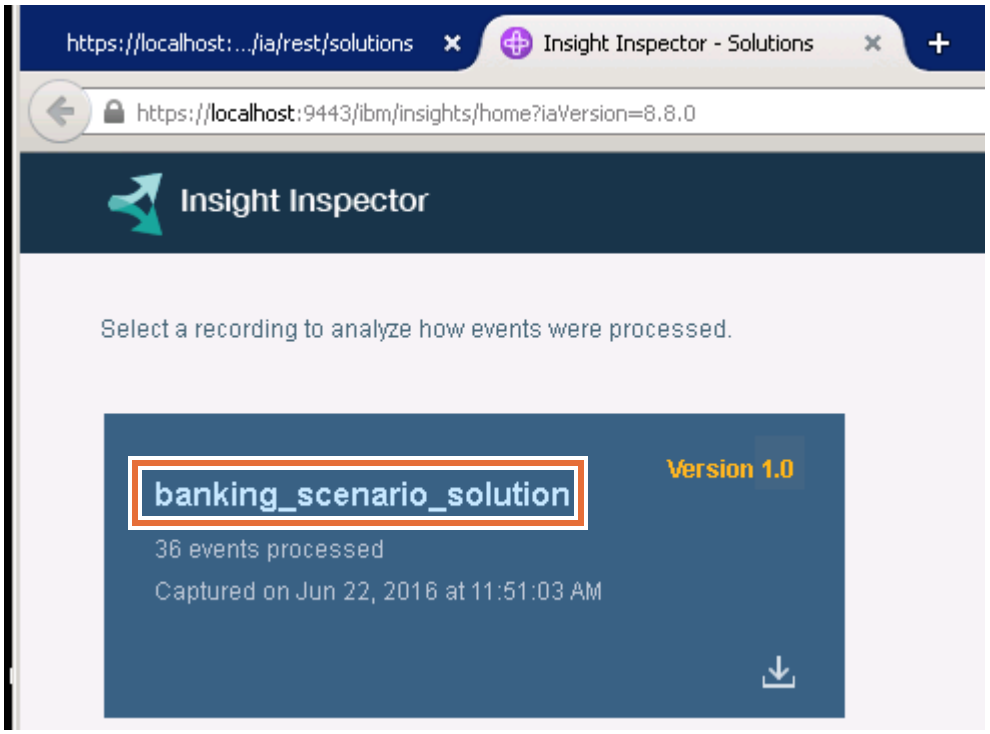
The browser returns a message that the recording stopped. You can now open Insight Inspector so see recorded events for the banking solution.

__ 2. Open Insight Inspector by one of the following methods:

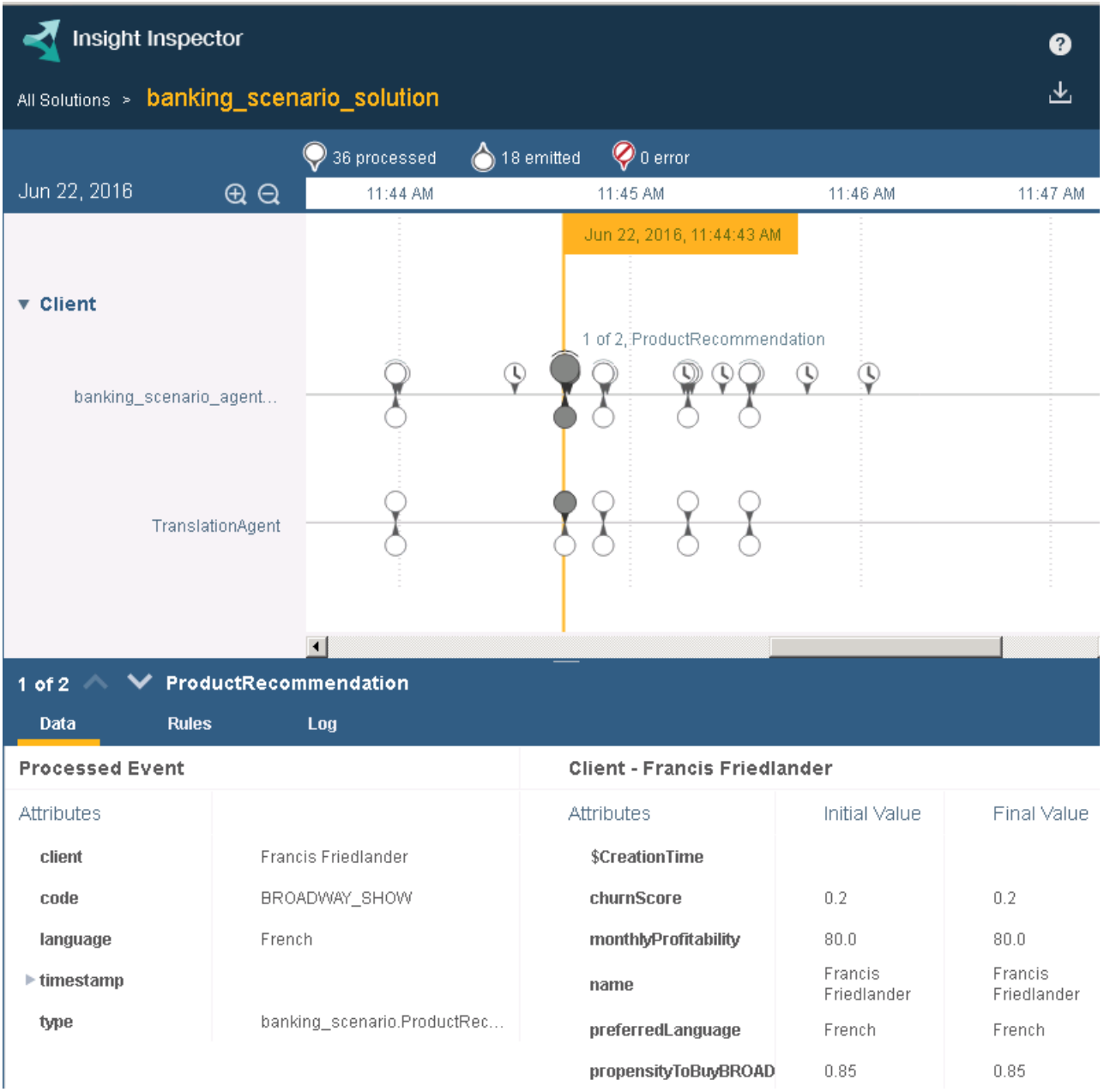
- Open a browser to this URL: `http://localhost:9080/ibm/insights`
- In the Sample Console perspective, click **Launch the Insight Inspector to view test recordings**

Your banking solution is listed on the home page.

__ 3. View the events by clicking **banking_scenario_solution**.



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



Insight Inspector shows a timeline that has the 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.

Note

If you could complete this exercise successfully, you can switch to the **<LabfilesDir>\workspace2-agg** workspace to see the solution projects.

Exercise 6. Using global aggregates in rules

Estimated time

01:15

Overview

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

Objectives

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***. Next, 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:

Requirements

For this exercise, you continue working in your current workspace or switch to a clean workspace and import the projects that are provided for this exercise.

Important

To switch to a new workspace:

__ 1. From the **File** menu, click **Switch Workspace > Other**.

__ 2. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

```
C:\labfiles\workspaces\aggregates
```

__ 3. Click **OK** to close the Workspace Launcher.

__ 4. Import the start 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 archive file** and click **Browse**.

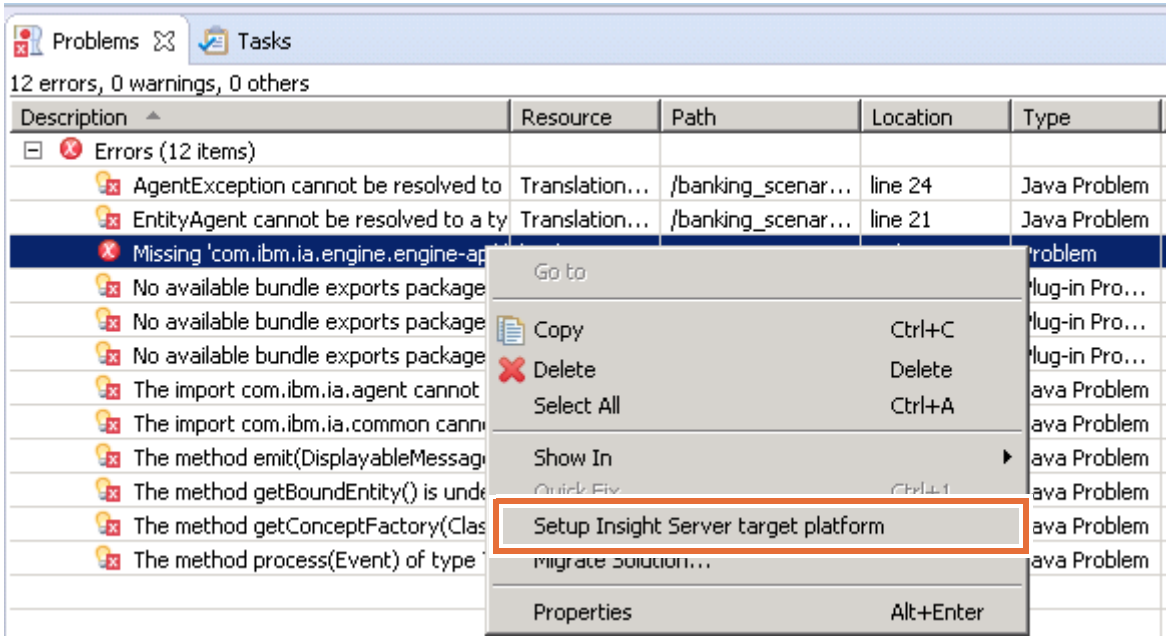
__ d. Go to the ***<LabfilesDir>***, select the `workspace2-agg.zip` file, and click **Open**.

__ e. Click **Finish**.

Your workspace now contains all the required projects.

__ 5. Resolve project errors.

__ a. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



__ b. Wait for the project to rebuild completely.

You should not see any errors after the workspace is rebuilt.

Section 1. 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.

1.1. 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.

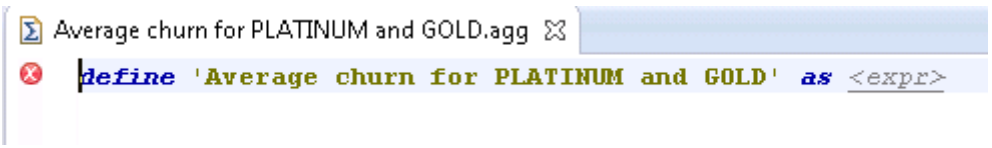
__ a. Click **File > New > Global Aggregate**.

__ 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`

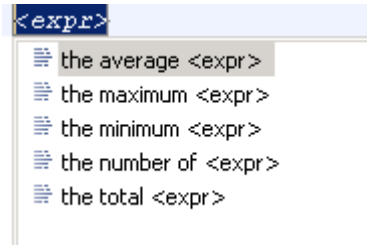
__ 3. Define the aggregate.

__ a. In the editor, click **<expr>**.

Questions

Consider these questions:

- Which of the expressions in the list do you choose?
  - the average
  - the minimum
  - the number of
  - the total



- 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
```

__ 4. Save your work and close the `Average churn for PLATINUM and GOLD.agg` file.

1.2. Creating the `Average monthly profitability for PLATINUM and GOLD` aggregate

__ 1. Create the aggregate definition file.

__ a. Click **File > New > Global Aggregate**.

__ 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
```

__ 3. 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 2. 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.

2.1. 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 Solution Explorer, click **banking_scenario_solution** to open the **Solution Map** view.

__ b. In the **Author** goal, click **Add rule agent**.

__ c. In the **Project name** field, type: `banking_scenario_agent_churn_prevention`

__ d. 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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

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

2.2. 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`

__ c. 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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

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

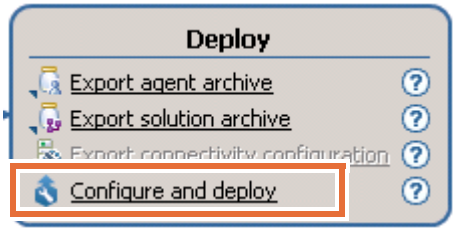
Important

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 3. Deploying the solution

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

__ 1. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.



__ 2. In the **Deployment configuration name field** of the Configure and Deploy wizard, select **local** and click **Next**.

__ 3. Leave the default values and click **Finish**.

Deployment takes a few moments. After deployment is complete, in the console, you see messages that the deployment was successful.

Section 4. Accessing your solution aggregates and entities through the REST API

4.1. Viewing the aggregates

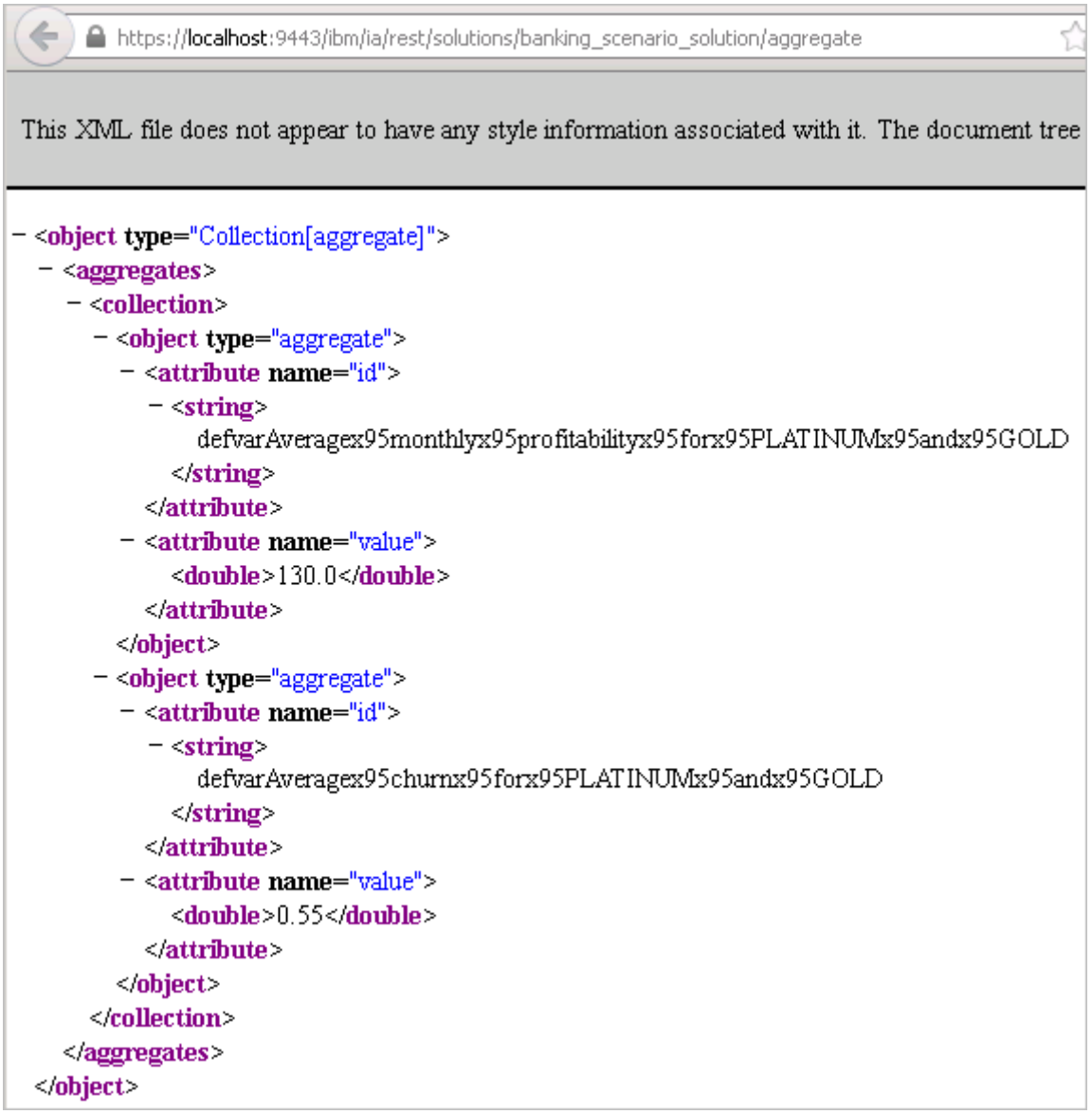
To view the deployed aggregates, you can use REST to access them in a browser.

__ 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 lists the aggregates that are defined for your solution.



Note

The browser automatically switches to a secure connection that uses `https://localhost:9443`.

4.2. Viewing the aggregates

__ 1. Open the browser and enter the following URL to inspect the entities with the REST API:

`http://localhost:9080/ibm/ia/rest/solutions/banking_scenario_solution/entity-types/banking_scenario.Client/entities`

The attributes for each of the client entities are listed, including the churn score and profitability. In the example here, you see the churn score and profitability for Di Lang.

For each client, look through the values, which should match these values:

Client	Churn score	Profitability
Dave Wakeman	0.20	80.00
Di Lang	0.20	81.00
Francis Friedlander	0.20	80.00
June Yoshii	**0.90**	**180.00**

Client	Churn score	Profitability
Srecko Janjic	**0.90**	**180.00**
Tonya Teyssier	**0.90**	**179.00**
Average / Global Aggregate	**0.55**	**130**
Factor	1.50	1.3
Rule threshold = Factor x Average	**0.825**	**169**

__ 2. 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.

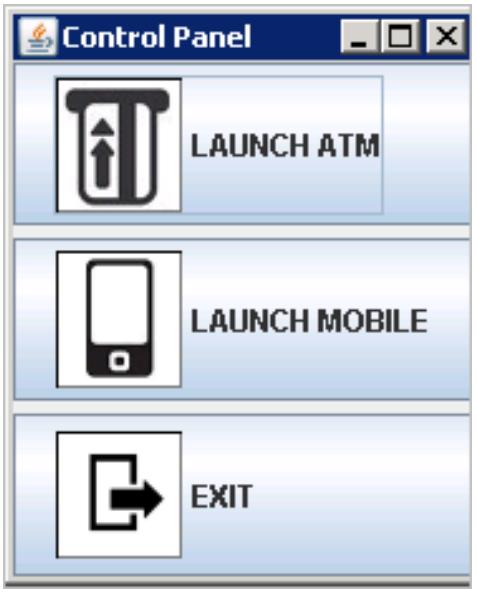
Section 5. Testing the solution

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.

On the mobile device for Tonya Teyssier, you see the message:
`Enjoy a free dinner on Broadway!`

__ 4. Open the `Reward churn client New York` rule to understand the rule behavior in relation to the churn score and profitability values that you saw in Activity 4, "Accessing your solution aggregates and entities through the REST API," on page 6-10.

__ 5. 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

Estimated time

00:45

Overview

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

Objectives

After completing this exercise, you should be able to:

- Use event aggregates and shared aggregates in rules

Introduction

This exercise includes these sections:

Requirements

For this exercise, you must switch to a clean workspace and import the projects that are provided for this exercise.

Section 1. Setting up your workspace

In Insight Designer, you switch to a new workspace for this exercise.

__ 1. Switch to a clean workspace.

__ a. From the **File** menu, click **Switch Workspace > Other**.

__ b. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

`C:\labfiles\workspaces\fraud`

__ c. Click **OK** to close the Workspace Launcher.

__ 2. Import the start 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 archive file** and click **Browse**.

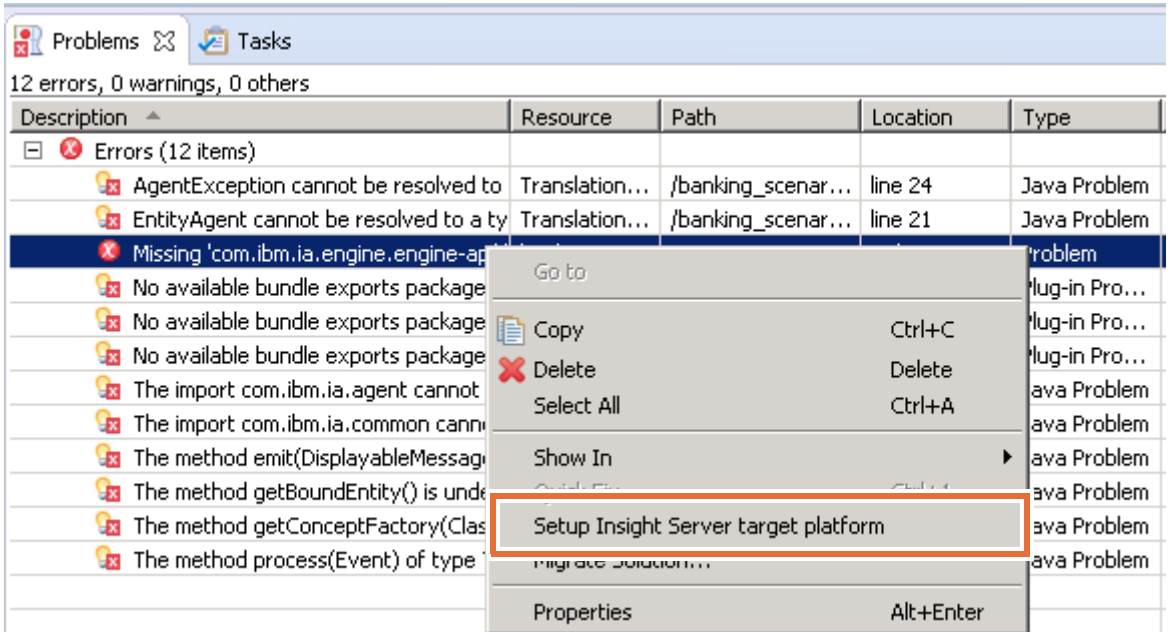
__ d. Go to the ***<LabfilesDir>*** and select the `workspace3-fraud.zip` file and click **Open**.

__ e. Click **Finish**.

Your workspace now contains all the required projects.

__ 3. Resolve project errors.

__ a. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



__ b. Wait for the project to rebuild completely.

You should not see any errors after the workspace is rebuilt.

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 `rule-agents.txt` file in the ***<LabfilesDir>*\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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** folder and format the text by pressing Ctrl+Shift+F.

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

Important

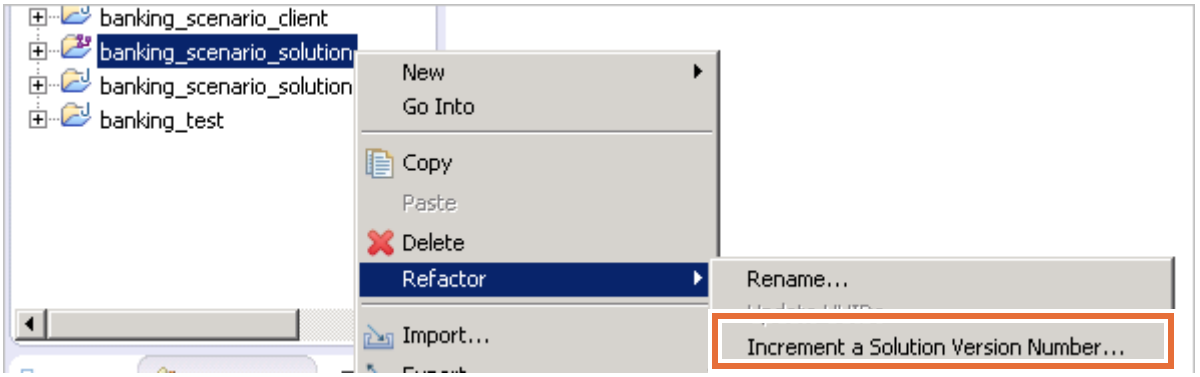
After you create a rule, you should test to ensure that the rule works before you continue writing more rules. For this banking scenario, use the test client that is provided for you.

Section 3. Deploying the solution

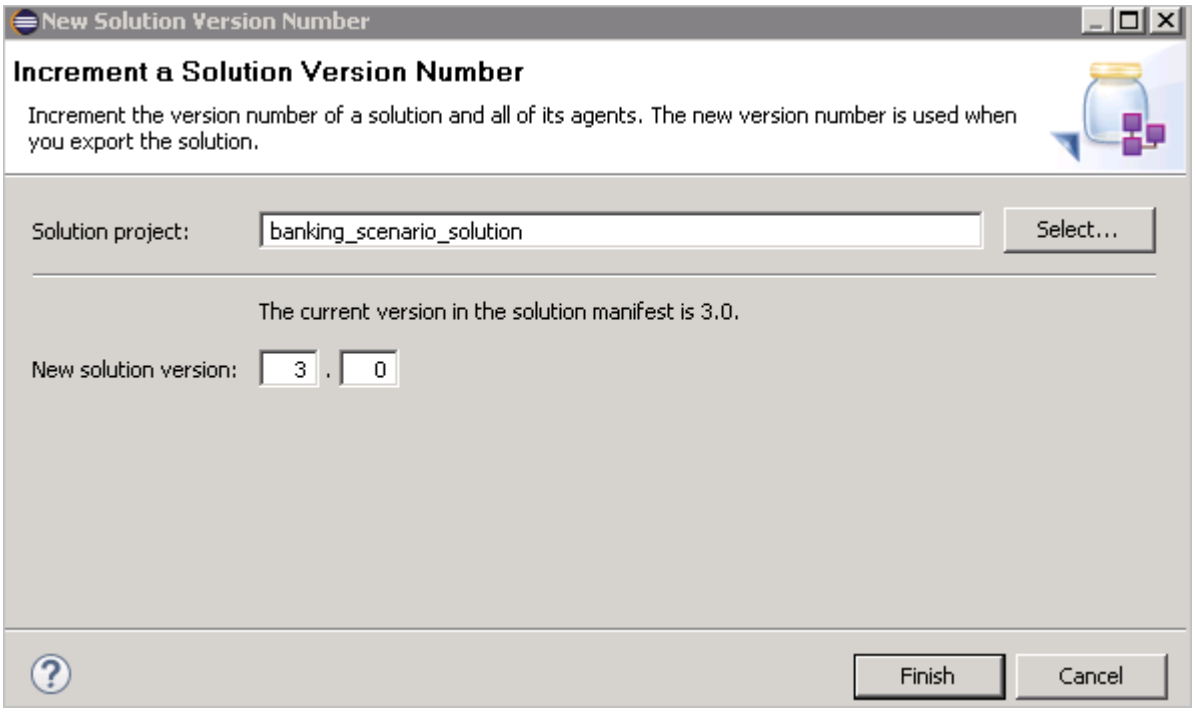
In this section, you deploy the solution.

3.1. Increment the version number

__ 1. In Solution Explorer, right-click **banking_scenario_solution**, and click **Refactor > Increment a Solution Version Number**.



__ 2. In the **New solution version** field, increment the major version. For example, if the version was **2.0**, change it to **3.0**.



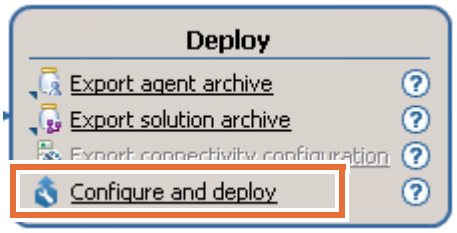
Note

Your version number might be different from the screen capture.

__ 3. Click **Finish** and wait for the workspace to rebuild.

3.2. Deploying the solution

__ 1. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.



__ 2. In the **Deployment configuration name field** of the Configure and Deploy wizard, select **local** and click **Next**.

__ 3. Leave the default values and click **Finish**.

Deployment takes a few moments. After deployment is complete, in the console, you see messages that state that the deployment was successful.

3.3. 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. Delete the previous recording.

__ a. Copy the version and time stamp of the previous recording.

- If the browser is still open from when you stopped the previous recording, you can copy the recording ID, which includes the solution name, version number, and time stamp.
- If the browser is closed, run the 'start recording' command:

```
http://localhost:9080/ibm/insights/rest/recording/start/banking_scenario_solution
```

The browser returns a message that includes the recording ID for the previous recording.

__ b. Using the recording ID for the previous recording, type the delete command in the browser.

```
http://localhost:9080/ibm/insights/rest/recording/delete/banking_scenario_solution-recordingID
```

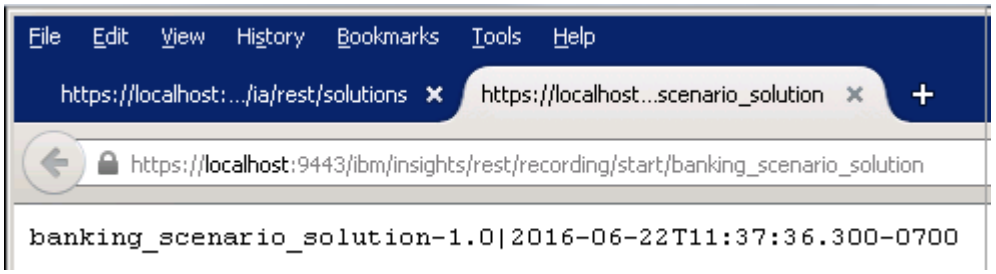
For example:

```
http://localhost:9080/ibm/insights/rest/recording/delete/banking_scenario_solution-2.0|2016-06-06T12:05:29.153-0700
```

__ 2. Open a new tab or browser window and type this URL:

```
http://localhost:9080/ibm/insights/rest/recording/start/banking_scenario_solution
```

The browser returns a message with the solution name and a time stamp. You use this time stamp as the recording ID to delete this recording after your tests are finished.



__ 3. 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

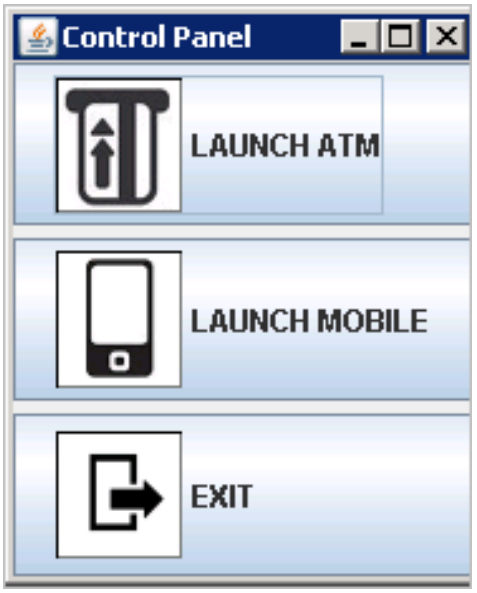
In this section, you test the solution and analyze the activity by using Insight Inspector.

4.1. 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 ATM** and send Withdraw events for Francis Friedlander.

__ a. Set the **Name** field to **Francis Friedlander**.

__ b. Make sure that the **Amount in USD** field is set to `100` and click **Withdraw** five times to send five withdrawal events.

__ c. Change the value in the **Amount in USD** field to 1000 and click **Withdraw** one time.

__ 4. In the test client Control Panel, click **Exit** to close the test client.

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

4.2. Analyze the test results in Insight Inspector

Now that you submitted events to the runtime, 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_solution`

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

__ 2. Open Insight Inspector, by typing the following URL in a browser:

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

Your banking solution is listed on the home page.

__ 3. Click **banking_scenario_solution** to view the newly captured recording.

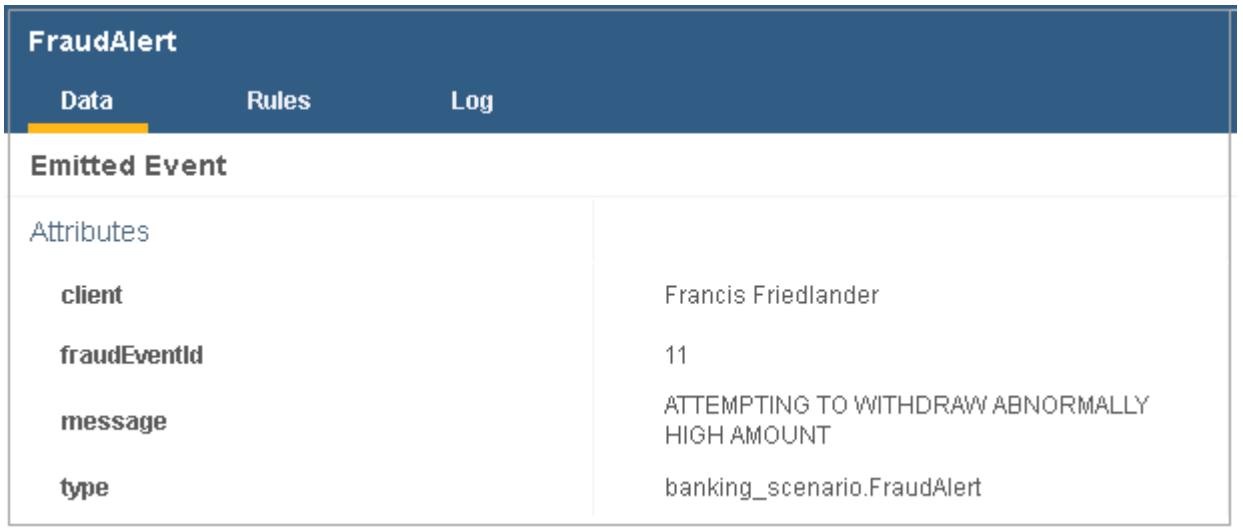
In the timeline of events for banking_scenario_agent_fraud_detection, the last event also has an emitted event icon.

__ 4. Click the emitted event icon.

__ 5. In the Data view, notice the attributes for the emitted event and their values, including the message value.

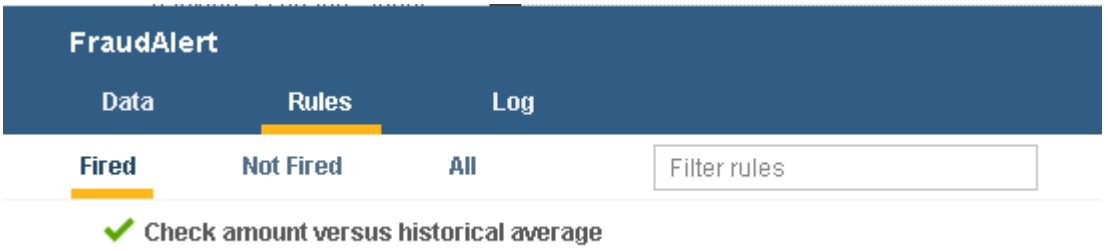
ATTEMPTING TO WITHDRAW ABNORMALLY HIGH AMOUNT

The amount of the last withdrawal event is 100 and the amount of the first five events is 1000. The fraud alert was successfully generated and emitted.

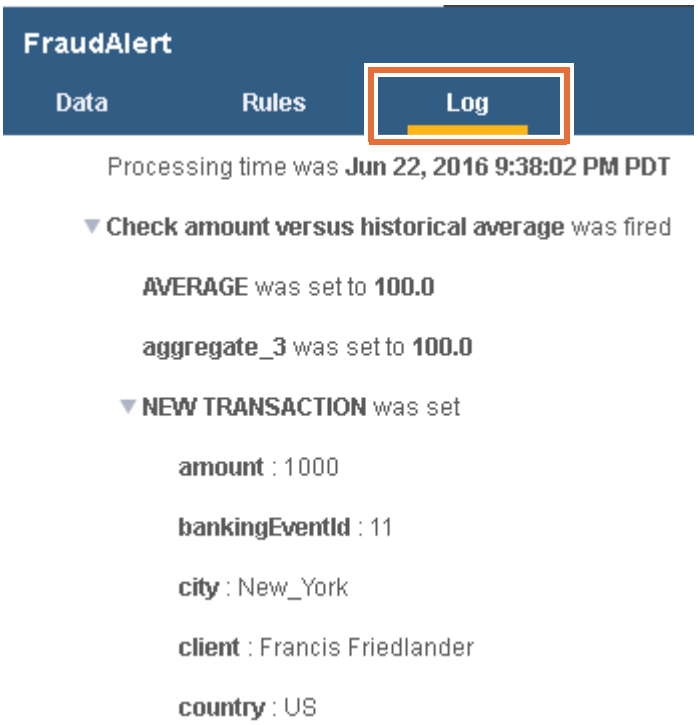


The Rules and Log are available to review next to the Data view of Insight Inspector.

__ 6. Click the **Rules** tab to view which rules fired.



__ 7. Click the **Log** tab to view the processing log.



__ 8. After you finish testing and reviewing the results, close Insight Inspector and click **Exit** on the Control Panel.

Section 5. Creating a shared aggregate

In this section, you create a shared aggregate.

5.1. Review the Check amount versus historical average rule

__ 1. Expand the **banking_scenario_agent_fraud_detection > rules** folder and double-click the `Check amount versus historical average` rule to open it.

__ 2. Review the rule to understand the **definitions** and the **if** statements.

```
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' ;
```

This rule defines two variables to determine the average transactions during a period of 50 days.

Next, you create a shared aggregate and rewrite this rule.

5.2. Create a shared aggregate in the BOM

__ 1. Expand **banking_scenario_bom > bom > banking_scenario** and double-click **BusinessModel.bmd**.

__ 2. Scroll down to the Entities section and append the following line to the client entity:

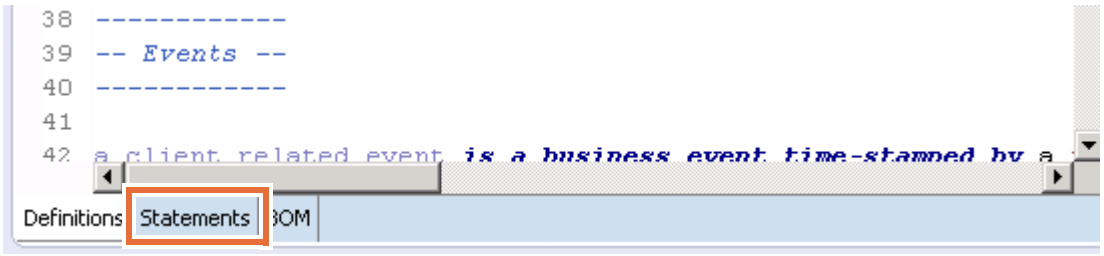
```
a client has a average withdrawal (numeric).
```

Hint

You can copy and paste this text from the `bmd-agg.txt` file in the ***<LabfilesDir>*\code** folder and format the text by pressing Ctrl+Shift+F.

__ 3. Save your work and wait for the workspace to build.

__ 4. In the BOM editor, click the **Statements** tab.



__ 5. In the Shared Aggregates section, append the following statement:

```
the average withdrawal of a client is aggregated from withdrawals ,
    where this client comes from the client of each withdrawal
  as the average amount of all withdrawals during the last period of 50 days
  defaulting to null if there are less than 5 events
  available for 5 days.
```

Hint

You can copy and paste this text from the `bmd-agg.txt` file in the ***<LabfilesDir>*\code** folder and format the text by pressing Ctrl+Shift+F.

This business model statement defines the aggregation of values for the `average withdrawal` as the average number of `withdrawal events` that the client makes.

__ 6. Save your work.

5.3. Rewrite the rule

__ 1. Compare the statement for the shared aggregate to the original rule in Section 5.1, "Review the Check amount versus historical average rule".

Questions

Consider the following questions:

- Can this statement replace parts of the rule? Which parts?
- How would you rewrite the rule?

__ 2. Open the `Check amount versus historical average` rule and replace the contents with the following text:

```
when a withdrawal occurs , called 'NEW TRANSACTION'
if
  the amount of 'NEW TRANSACTION' is more than 3 * the average withdrawal of
'the client'
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' ;
```

__ 3. Save your work.

__ 4. Redeploy your solution by following the steps in Section 3, "Deploying the solution". Make sure to delete the old recording for Insight Inspector and restart the recording.

__ 5. Test the solution by following the steps in Section 4, "Testing the solution"

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

Estimated time

00:30

Overview

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

Objectives

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:

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. If you are not in the **Decision Insight** perspective, switch to it now.

__ 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.

__ a. Enter the following rule into the rule editor:

```
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

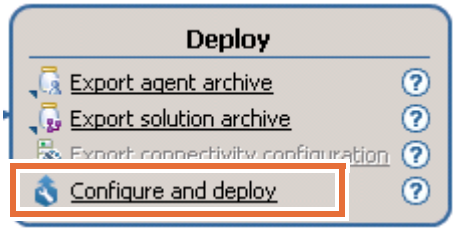
You can copy and paste this text from the `rule-agents.txt` file in the **<*LabfilesDir*>\code** folder and press Ctrl+Shift+F to format the text.

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

Section 2. Deploying the solution

In this section, you export and deploy the solution.

__ 1. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.



__ 2. In the **Deployment configuration name field** of the Configure and Deploy wizard, select **local** and click **Next**.

__ 3. Leave the default values and click **Finish**.

Deployment takes a few moments. After deployment is complete, in the console, you see messages that state that the deployment is successful.

Section 3. Setting up the map

3.1. Adding the map feature to the server

__ 1. Go to the Samples Console perspective.

__ 2. Stop the cisDev server.

__ 3. Add the Insight Map Viewer feature to the `server.xml` file for the cisDev server.

__ a. Open the <***InstallDir***>`\runtime\wlp\usr\servers\cisDev` directory.

__ b. Open the `server.xml` file in Notepad++.

__ c. Add this line in the **featureManager** section:

`<feature>ia:iaMaps-8.8.0</feature>`

```
<featureManager>
    <feature>restConnector-1.0</feature>
    <feature>ssl-1.0</feature>
    <feature>ia:iaRuntime-8.8.0</feature>
    <feature>ia:iaDispatcher-8.8.0</feature>
    <feature>ia:iaAnalytics-8.8.0</feature>
    <feature>ia:iaHTTPGateway-8.8.0</feature>
    <feature>ia:iaConnectivityInboundHTTP-8.8.0</feature>
    <feature>ia:iaConnectivityOutboundHTTP-8.8.0</feature>
    <feature>usr:banking_scenario_solution-2.0</feature>
    <feature>ia:iaMaps-8.8.0</feature>
</featureManager>
```

__ d. Save the file.

__ 4. Start the server.

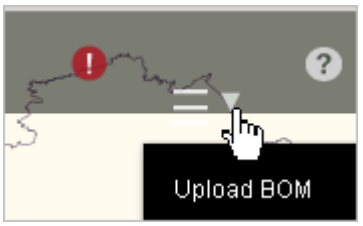
3.2. Opening the map

__ 1. Open a browser to this URL:

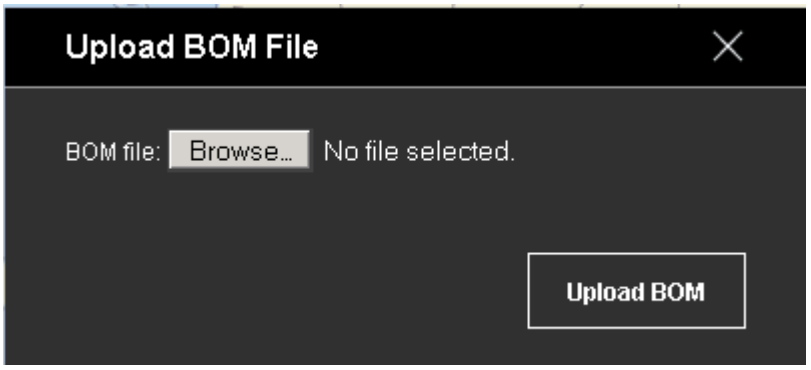
`http://localhost:9080/ibm/maps/application.html`

It might take a few minutes to load the map.

__ 2. In the upper-right part of the map, click the menu arrow and select **Upload BOM**.

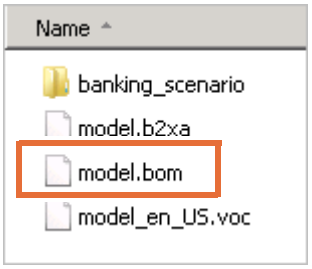


__ 3. In the **BOM file** field, click **Browse**.



__ 4. Go to your workspace and select the `model.bom` file. For example:

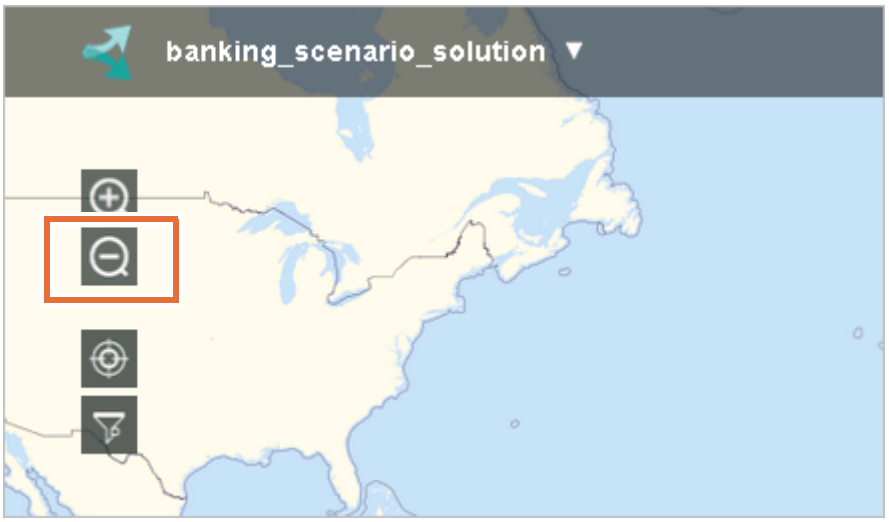
`<LabfilesDir>\workspaces\your_workspace\banking_scenario_bom\bom\model.bom`



__ 5. Upload the BOM file by clicking **Upload BOM**.

In the upper-left corner, you should see **banking_scenario_solution**.

__ 6. Zoom out to see the full map by clicking the zoom-out icon.



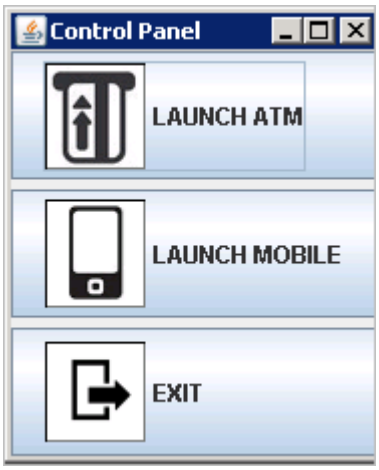
Section 4. Testing the solution

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. 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.



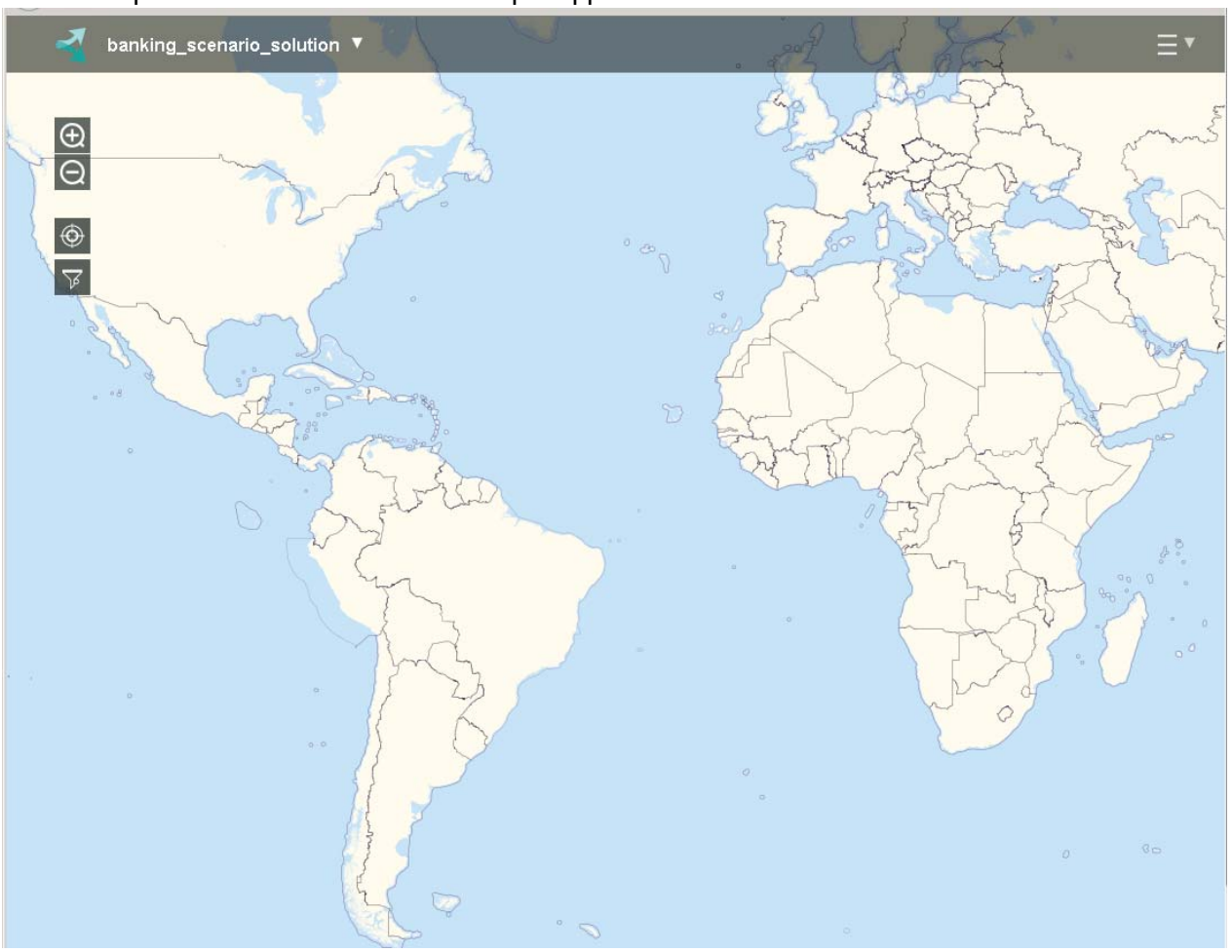
__ 3. Click **LAUNCH MOBILE**.

__ 4. 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.

__ 5. Make sure that the browser that is running the map is open at this URL.

`http://localhost:9080/ibm/maps/application.html`



__ 6. Set the ATM options:

- ATM 1: **New_York** and **Di Lang**
- ATM 2: **San_Francisco** and **Di Lang**

__ 7. Click **Withdraw** first on ATM 1, and then on ATM 2.

After June withdraws cash on ATM 2, you see a fraud alert on the map. Review the `Check distance to recent events` rule to understand the behavior.

__ 8. Optional. Continue testing with other locations.

__ 9. After you finish testing, click **EXIT** in the Control Panel.

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

Estimated time

00:30

Overview

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

Objectives

After completing this exercise, you should be able to:

- Test for the absence of events

Introduction

This exercise includes these sections:

Requirements

For this exercise, you continue working in your current workspace or switch to a clean workspace and import the projects that are provided for this exercise.

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, which includes the solution to the previous exercise.

__ 1. To switch workspaces:

__ a. From the **File** menu, click **Switch Workspace > Other**.

__ b. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

`C:\labfiles\workspaces\fraud-manage`

__ c. Click **OK** to close the Workspace Launcher.

__ 2. Import the start 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 archive file** and click **Browse**.

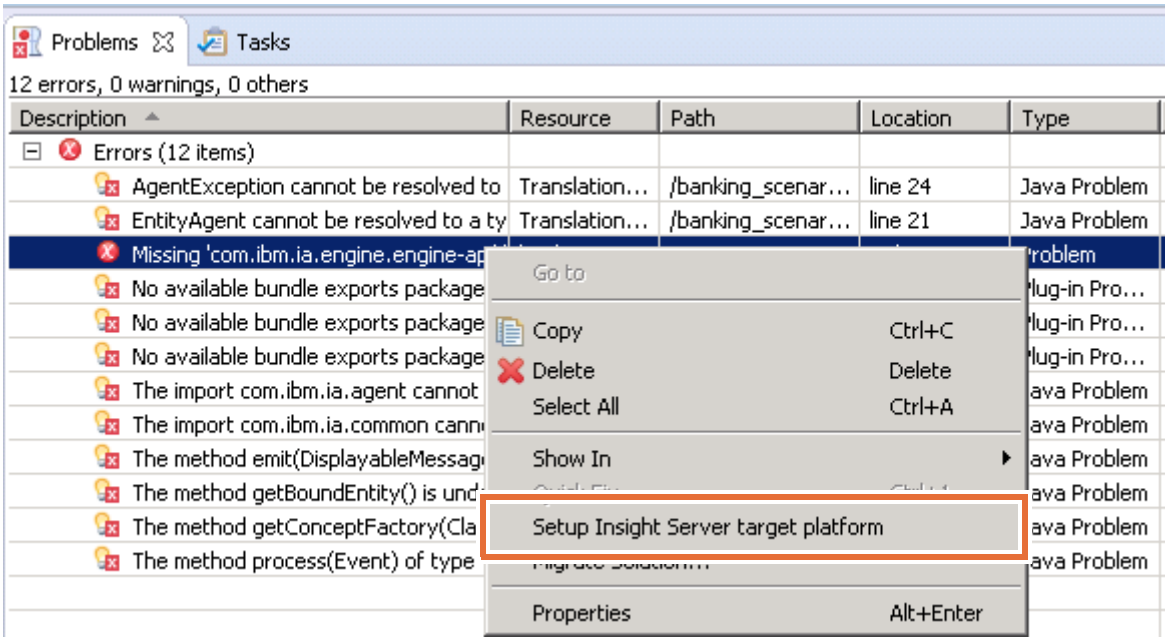
__ d. Go to the ***<LabfilesDir>*** and select the `workspace4-absent.zip` file and click **Open**.

__ e. Click **Finish**.

Your workspace now contains all the required projects.

__ 3. Resolve project errors.

__ a. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



__ b. Wait for the project to rebuild completely.

You should not see any errors after the workspace is rebuilt.

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. Select the solution project and 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 `rule-agents.txt` file in the ***<LabfilesDir>*\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.

```
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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** folder.

Note

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.

__ 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 `rule-agents.txt` file in the ***<LabfilesDir>*\code** folder.

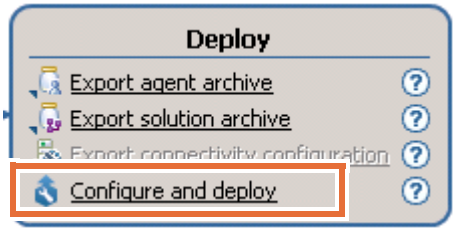
__ 3. Save the rule (Ctrl+S).

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

__ 1. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.



__ 2. In the **Deployment configuration name field** of the Configure and Deploy wizard, select **local** and click **Next**.

__ 3. Leave the default values and click **Finish**.

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 MOBILE** and select **June Yoshii** as the client.

__ 4. 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.

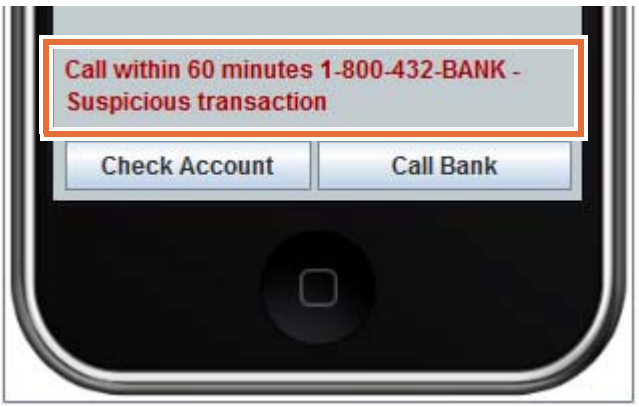
__ b. Set the ATM options:

- ATM 1: **New_York** and **June Yoshii**
- ATM 2: **Newark** and **June Yoshii**

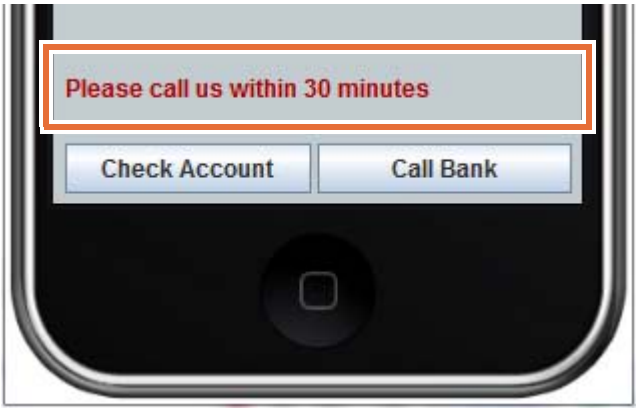
__ c. Click **Withdraw** first on ATM 1, and then on ATM 2.

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.



__ 5. Review the `Ask client to confirm within 60 minutes` rule to understand the rule behavior in relation to the alert.

__ 6. Review the `Remind after 30 min if no action taken` rule to understand the rule behavior in relation to the reminder message.

__ 7. Redo Step 4, 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.

__ 8. 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.Testing solutions

Estimated time

00:30

Overview

This exercise covers how to create and test a solution with the Test Client.

Objectives

After completing this exercise, you should be able to:

- Create a test client project
- Run a test scenario

Introduction

This exercise includes these sections:

Requirements

For this exercise, you continue working in your current workspace.

Section 1. Creating a test client

In this section, you create and run a test client to test the solution that you worked on in previous exercises.

1.1. Create the test client project

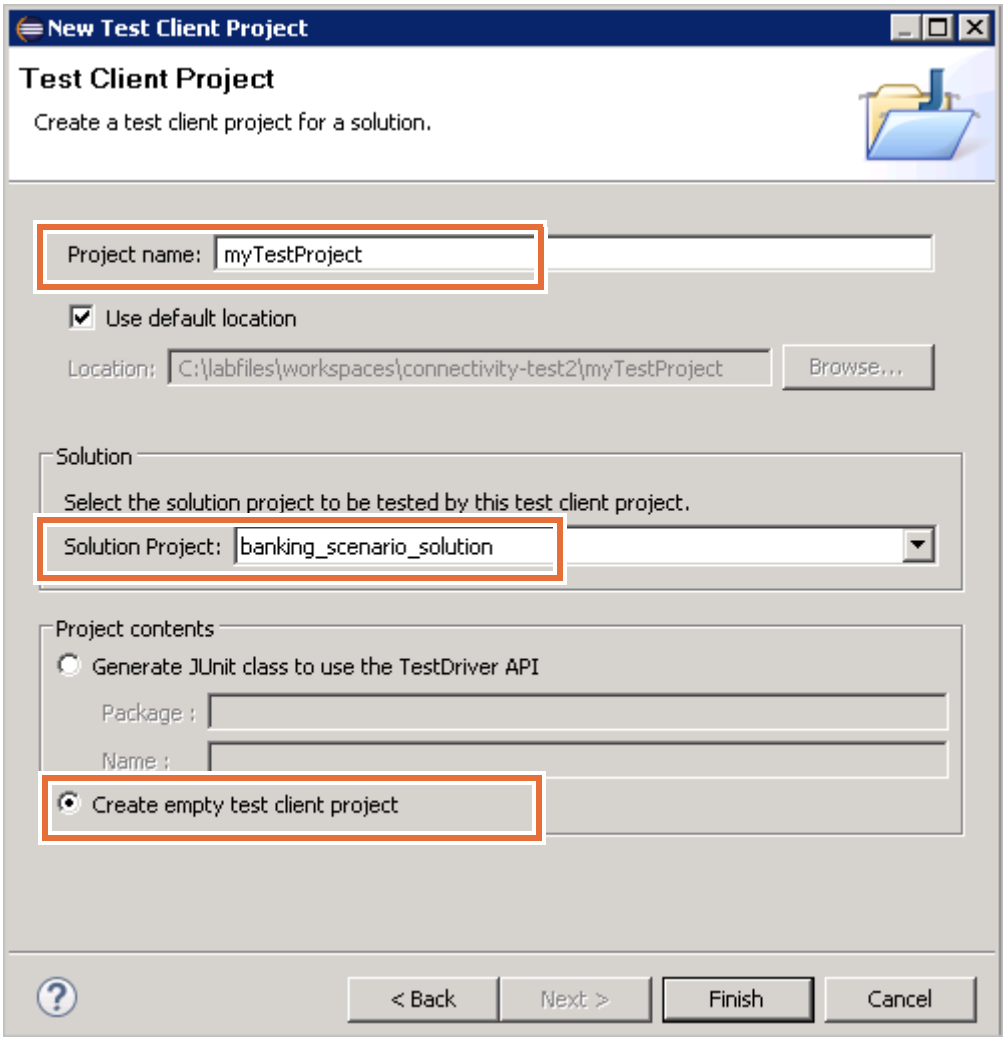
__ 1. In Insight Designer, make sure that you are in the Decision Insight perspective.

__ 2. From the **File > New > Other** menu, click **Insight Designer > Test Client Project** and click **Next**.

__ 3. In the **Name** field, type: `myTestProject`

__ 4. In the **Solution Project** field, make sure that **banking_scenario_solution** is selected.

__ 5. In the **Project contents** section, select **Create empty test client project**.



__ 6. Click **Finish**.

The new test client project is generated and opens in Solution Explorer.

1.2. Define the test client artifacts

__ 1. Create the common definitions file.

__ a. Expand **myTestProject** and right-click the **Common Definitions** folder of **myTestProject**, click **New > Other > Insight Designer > Common Definitions File** and click **Next**.

__ b. In the **Name** field, type `commondef` and click **Finish**.

__ c. Add these definitions.

```
define 'francis' as a new client where
    the name is "Francis Friedlander" ,
    the churn score is 0.2 ,
    the monthly profitability is 80 ,
    the preferred language is French  ,
    the propensity to buy BROADWAY_SHOW_TICKETS is 0.85 ,
    the segment is GOLD   ;
```

Hint

You can copy and paste this text from the `test-client.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

__ d. Save the file and close it.

__ 2. Create an entity loader

__ a. Right-click the **Entity Loaders** folder and click **New > Other > Insight Designer > Entity Loader** and click **Next**.

__ b. In the **Name** field, type `clients` and click **Finish**.

__ c. Add these entities.

```
using definitions from "commondef" ;
load francis ;
```

Hint

You can copy and paste this text from the `test-client.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

__ d. Save the file and close it.

__ 3. Create an event sequence

__ a. Right-click the **Event Sequences** folder and click **New > Other > Insight Designer > Event Sequence** and click **Next**.

__ b. In the **Name** field, type `fraud` and click **Finish**.

__ c. Add these events.

```
using definitions from "commondef"  ;

define atm1 as a new withdrawal where
    the client is francis ,
   the amount is 100  ,
    the banking event id is "0021" ,
    the city is Paris  ,
    the location is the point with 2.3508 as longitude and 48.8567 as
latitude ;

define atm2 as a new withdrawal where
    the client is francis ,
   the amount is 100  ,
    the banking event id is "0022" ,
    the city is Beijing  ,
    the location is the point with 116.4074 as longitude and 39.9042 as
latitude ;

emit atm1 , time-stamped 06/01/2016 10:00:00 AM -0500 ;
emit atm2 , time-stamped 5 minutes later  ;
```

Hint

You can copy and paste this text from the `test-client.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

__ d. Save the file and close it.

__ 4. Create a test scenario.

__ a. Right-click the **Test Scenarios** folder and click **New > Other > Insight Designer > Test Scenario** and click **Next**.

__ b. In the **Name** field, type `fraud` and click **Finish**.

__ c. Add these definitions and close the file.

```
load entities from "clients" ;
check that the client "Francis Friedlander" exists ;
submit events from "fraud"    ;
```

Hint

You can copy and paste this text from the `test-client.txt` file in the ***<LabfilesDir>*\code** folder and press Ctrl+Shift+F to format the text.

__ d. Save the file and close it.

Section 2. Testing the solution with the test client

__ 1. In the Solution explorer, expand **myTestProject > Test Scenarios**, right-click **fraud.tscn**, and click **Run As > Test Scenario**.

The Run Configurations window opens.

__ 2. Create a Test Scenario configuration.

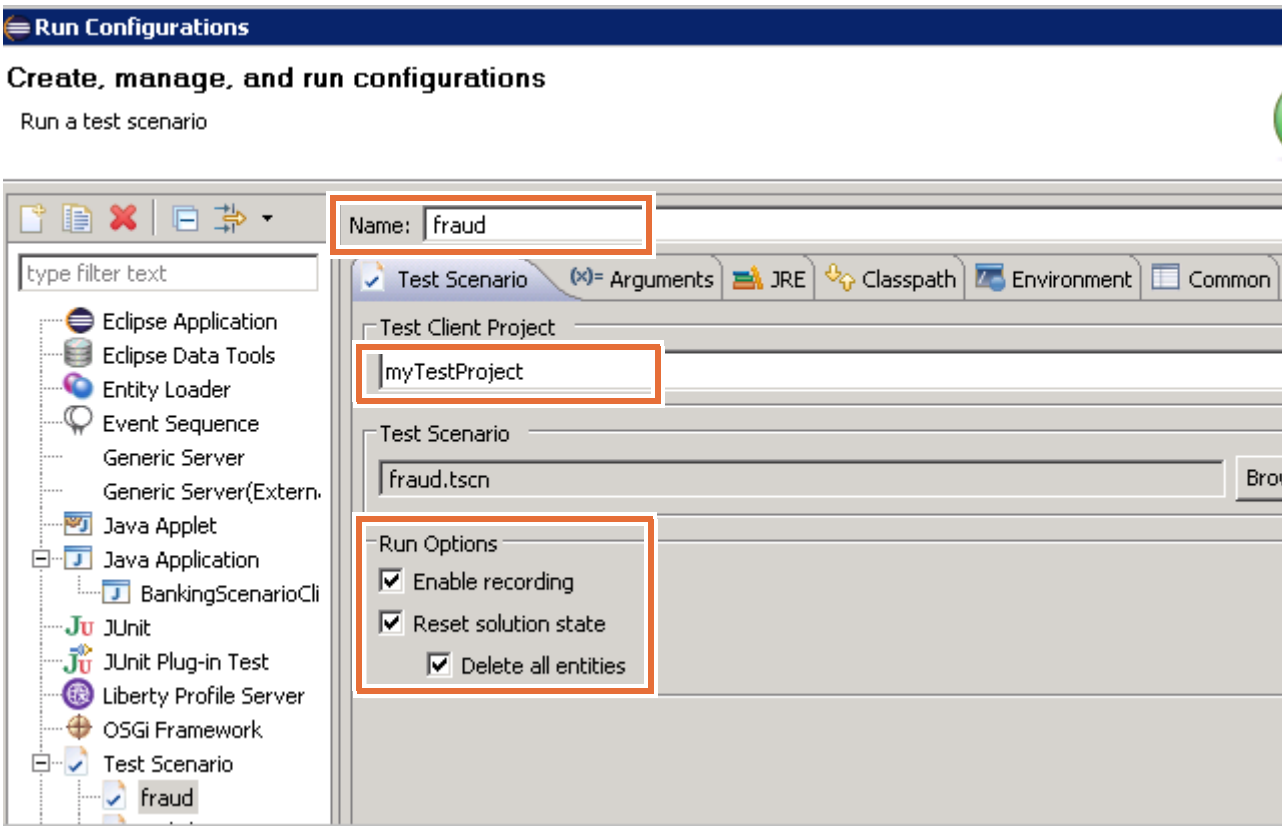
__ a. Set the **Name** field to: `fraud`

__ b. In the **Test Client Project** field, make sure that `myTestProject` is selected.

__ c. In the **Test Scenario** field, click **Browse**, and browse to the `fraud.tscn` file.

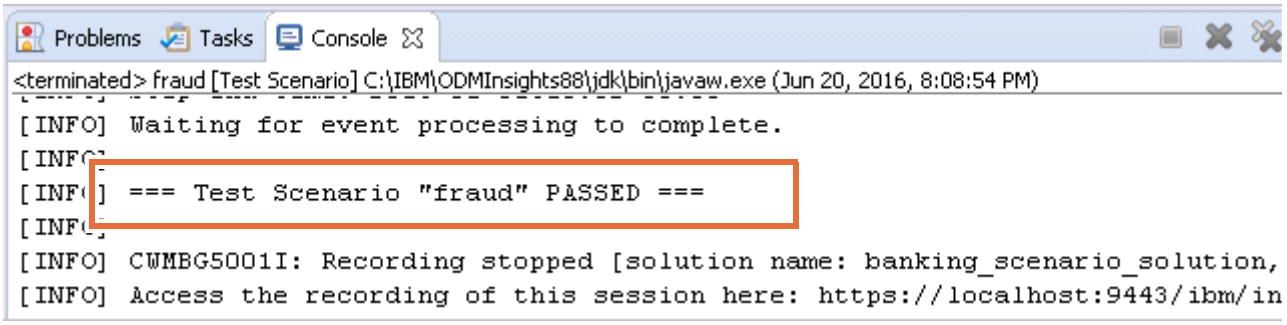
__ d. In **Run Options**, select:

- Enable recording
- Reset solution state
- Delete all entities



__ 3. Click **Run**.

After the test finishes, you see a message in the Console that the test scenario passed.



In the Console, the “passed” message is followed by a link to the Insight Inspector recording of this test.

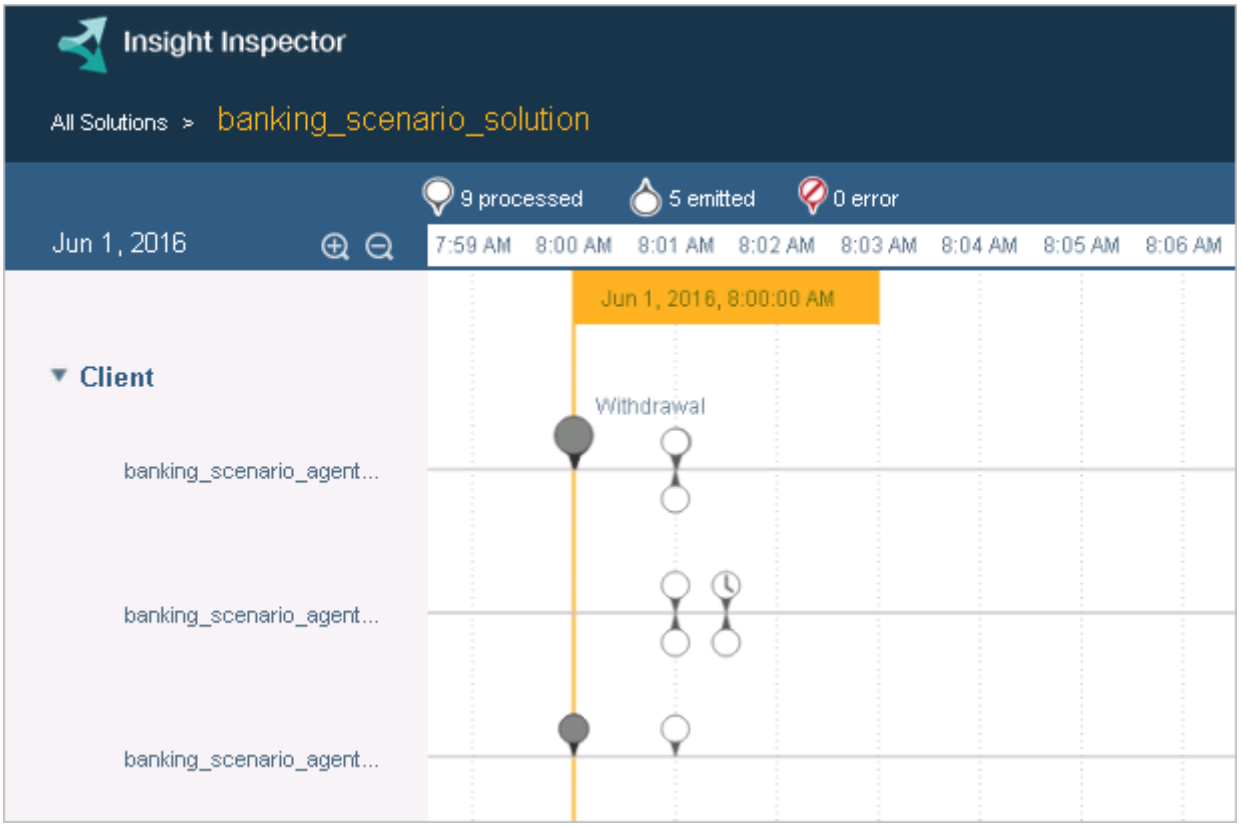
__ 4. Copy the link from the console and paste it in a browser to open Insight Inspector.

2.1. Analyze the test results in Insight Inspector

__ 1. In the browser open to the Insight Inspector link from the previous step, scroll to **June 1** to find the events that were processed and emitted.

Note

Use the zoom tools so that you can better see the different events.

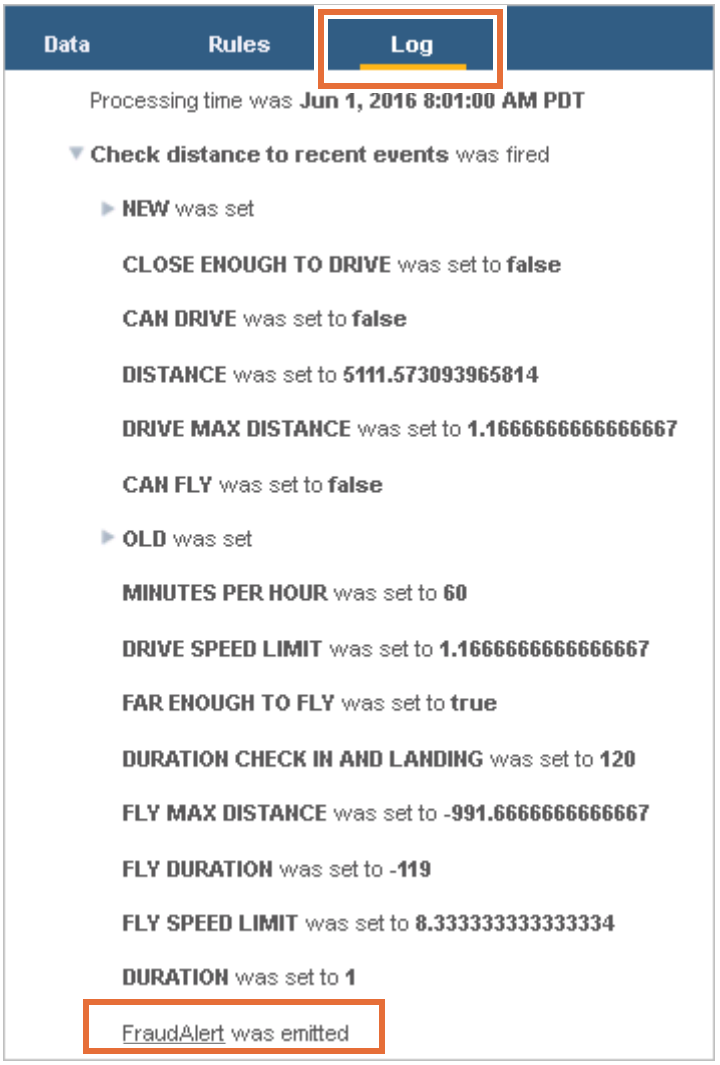


__ 2. Click the first submitted “withdrawal” event to see the description in the **Data** section of the lower pane. In the **Attributes** list, the **city** is **Paris**.

__ 3. Click the second submitted “withdrawal” event and note that the **city** is **Beijing**.

__ 4. Click the **Rules** tab to see which rule fired.

__ 5. Click the **Log** tab to see how the rule was processed.



__ 6. After you finish reviewing the test results, close the browser.

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 11. Using the Map Viewer

Estimated time

00:30

Overview

This exercise shows how to configure and use the Map Viewer to test a solution.

Objectives

After completing this exercise, you should be able to:

- Configure Insight Map Viewer
- Run a solution and view entities that are displayed on Insight Map Viewer

Introduction

This exercise includes these sections:

Requirements

For this exercise, you continue working in your current workspace or switch to a clean workspace and import the projects that are provided for this exercise.

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, which includes the solution to the previous exercise.

__ 1. To switch workspaces:

__ a. From the **File** menu, click **Switch Workspace > Other**.

__ b. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

`C:\labfiles\workspaces\geo`

__ c. Click **OK** to close the Workspace Launcher.

__ 2. Import the start 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 archive file** and click **Browse**.

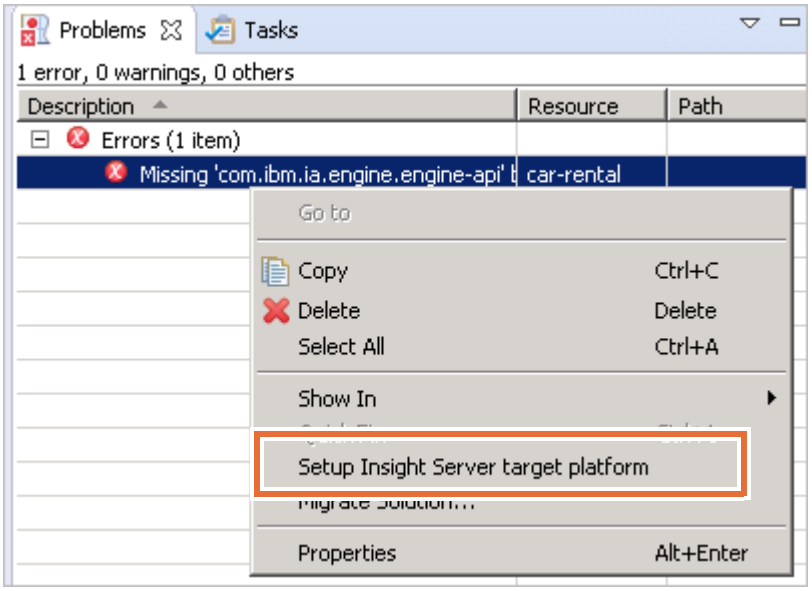
__ d. Go to the ***<LabfilesDir>*** and select the `car-rental.zip` file and click **Open**.

__ e. Click **Finish**.

Your workspace now contains all the required projects.

__ 3. Resolve project errors.

__ a. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



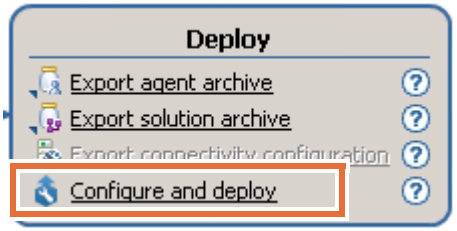
__ b. Wait for the project to rebuild completely.

You should not see any errors after the workspace is rebuilt.

Section 2. Deploying the solution

In this section, you deploy the solution.

__ 1. In the Solution Map view, in the **Deploy** goal, click the **Configure and deploy** link.



__ 2. In the **Deployment configuration name** field of the Configure and Deploy wizard, type **local** and click **Next**. Leave the default values and click **Finish**.

Deployment takes a few moments.

Section 3. Preparing to test the solution

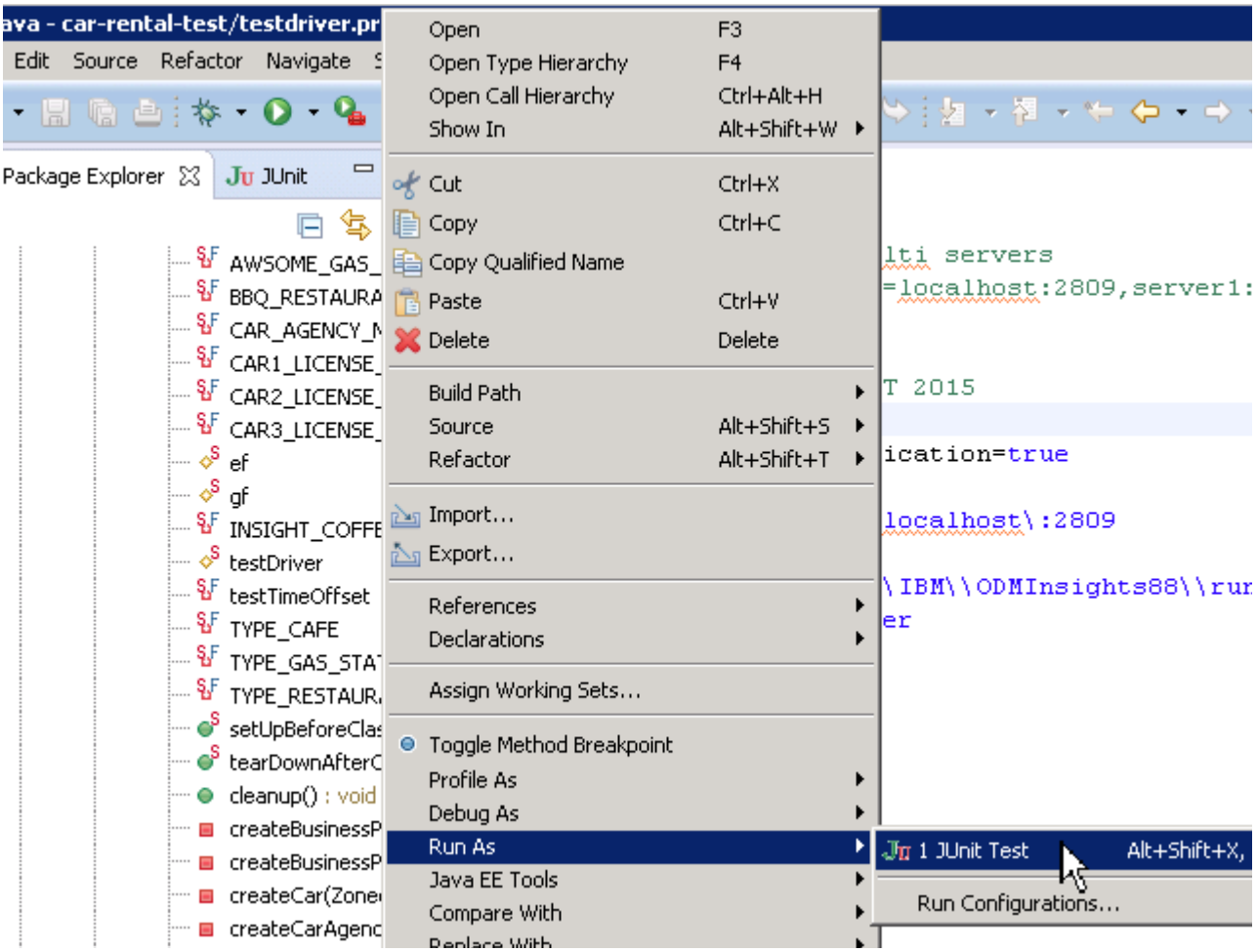
In this section, you prepare Insight Map Viewer, customize how entities are displayed on the map, and test the solution. During the test run, you see the entities move on the map.

3.1. Preparing the test

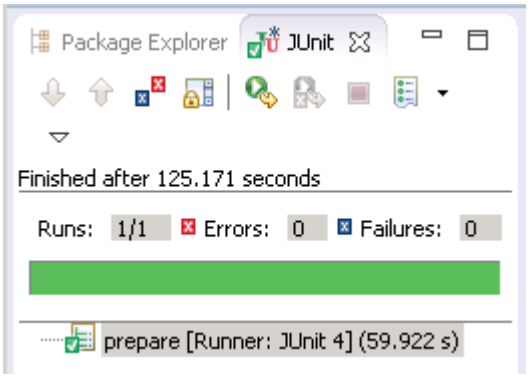
__ 1. In Insight Designer, under the `car-rental-test` project, double-click the `testdriver.properties` file to open it.

__ 2. Switch to the Java perspective, expand **car-rental-test project > src > cars > CarRentalTest.java**.

__ 3. Right-click the **prepare()** method, and choose **Run As > JUnit Test** from the menu.



__ 4. Wait until the JUnit test finishes, and shows the successful run in the **JUnit** view.

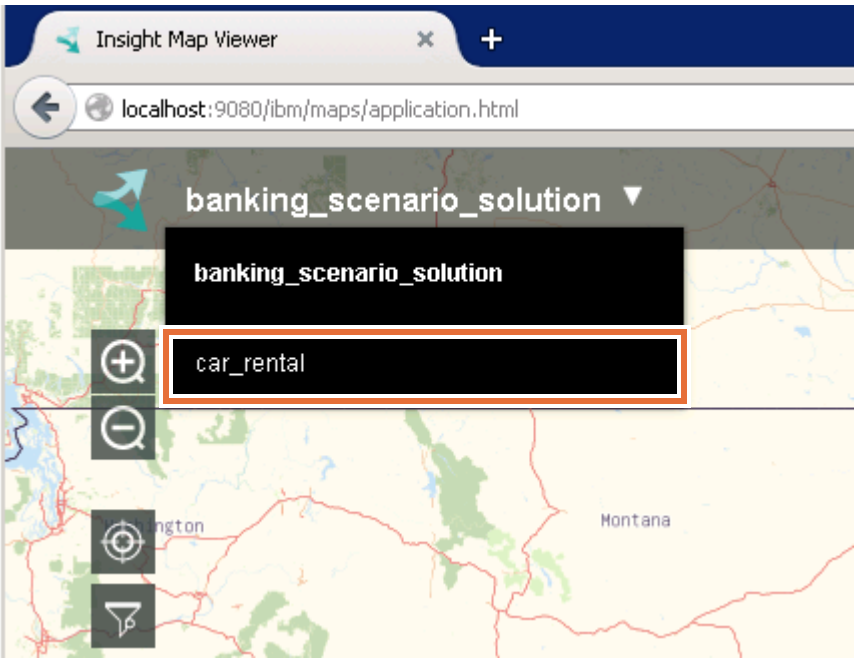


3.2. Configure the solution map with the BOM

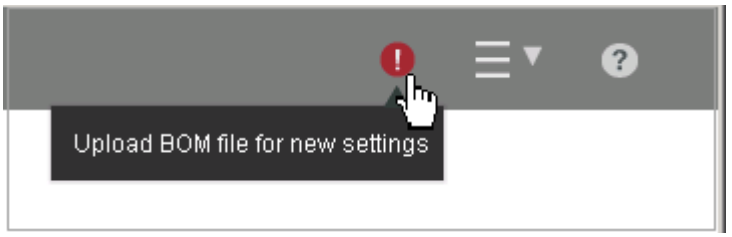
__ 1. Open a browser window and enter the following URL:
`http://<host>:<port>/ibm/maps/application.html`

For example: `http://localhost:9080/ibm/maps/application.html`

__ 2. When the map is open, click the menu icon in the upper-left corner, and select **car_rental** to load the `car_rental` solution.



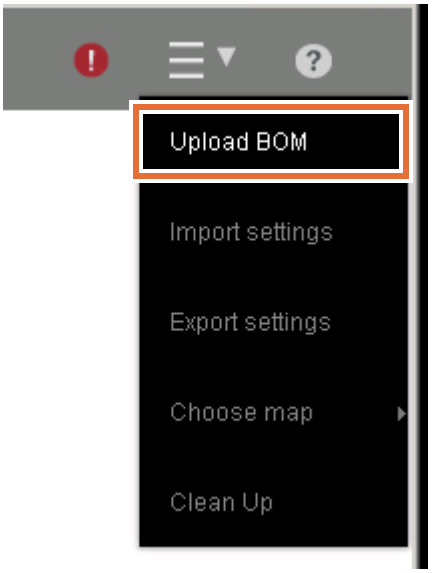
__ 3. On the upper-right corner of the map, you see a warning icon that indicates that no BOM is set for the map.



Note

You might need to reload the web page to see the warning icon.

__ 4. Click the menu next to the warning icon, and click **Upload BOM**.

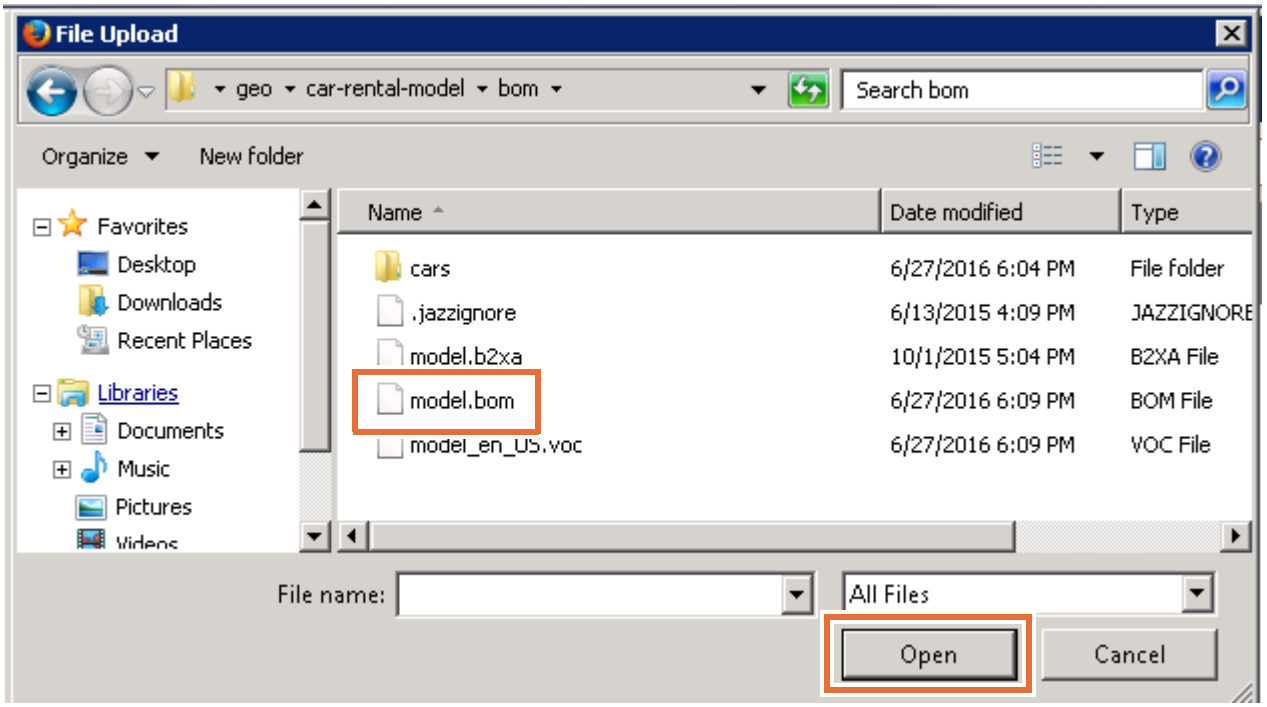


__ 5. When the Upload BOM window opens, click **Browse**.

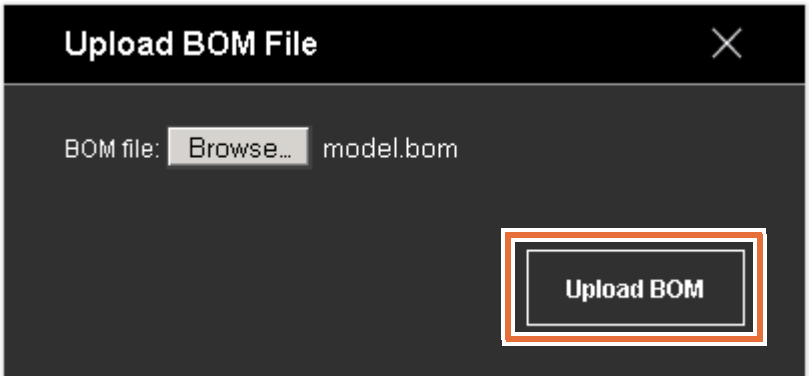
__ 6. In the File Upload window, go to the workspace folder that has the BOM that is used in this exercise:

<LabfilesDir>\workspaces\geo\car-rental-model\bom

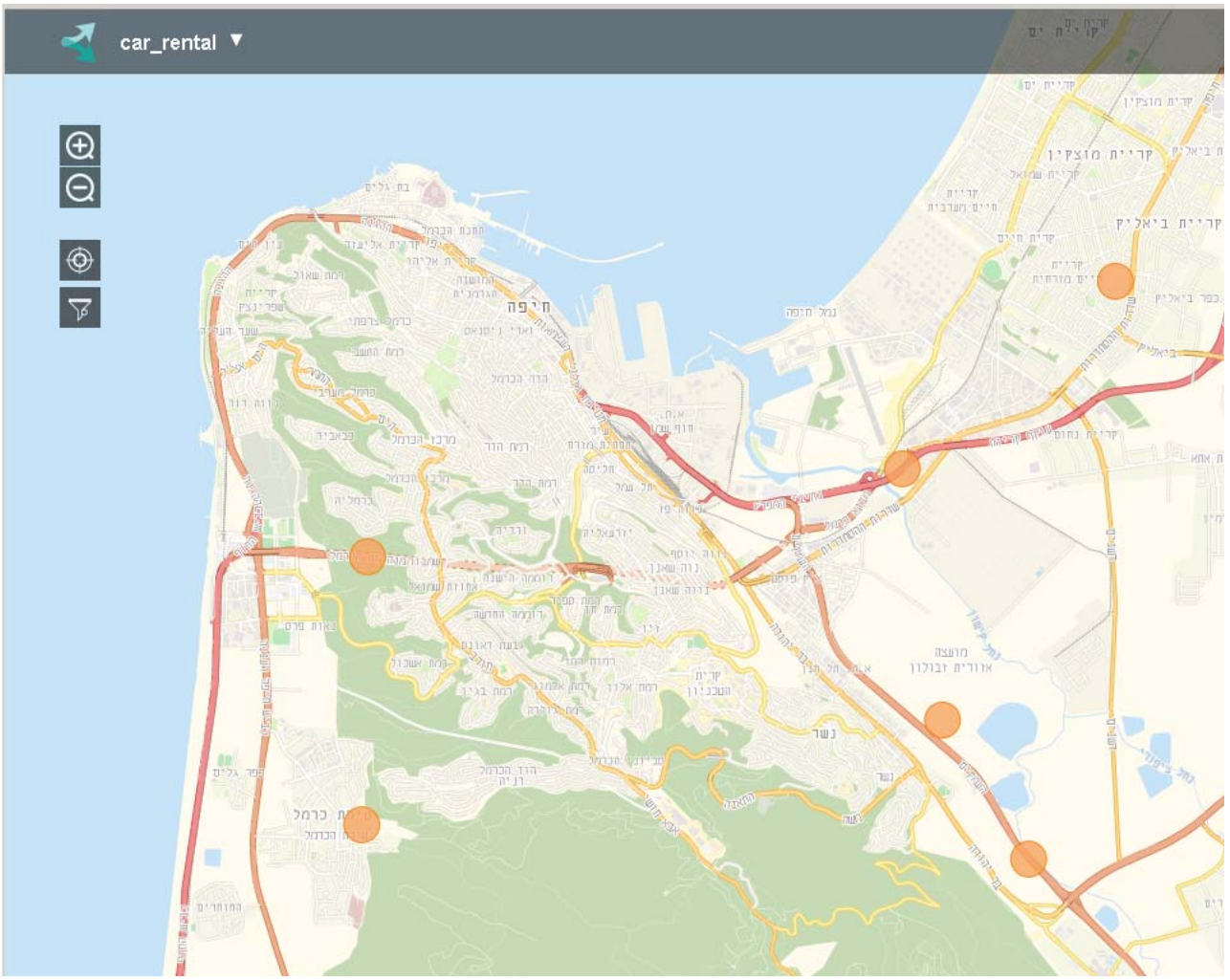
__ 7. Select the **model.bom** file, and click **Open**.



__ 8. Click **Upload BOM**.

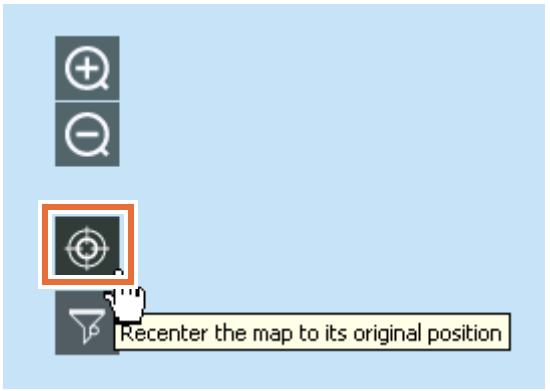


The map reloads with six entities, which are represented as orange circles.



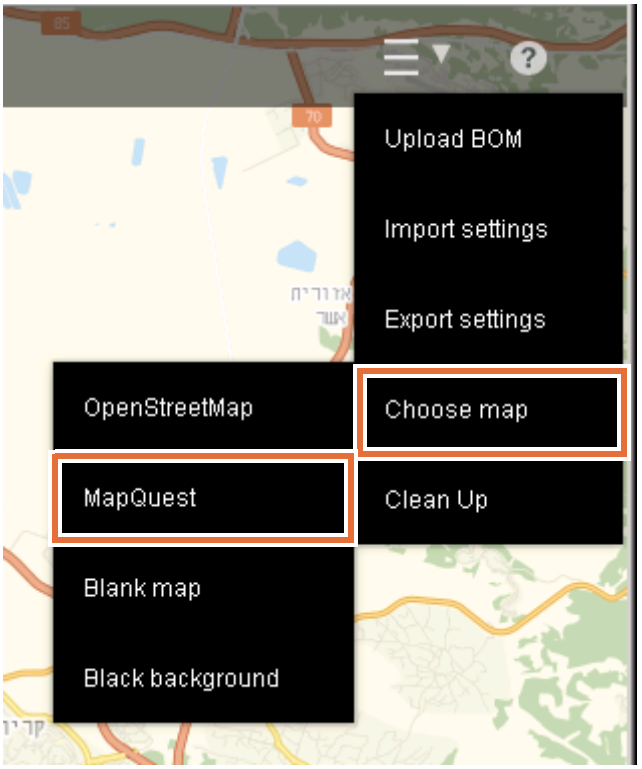
Troubleshooting

If the map does not load, click the “Recenter the map to its original position” icon on the left side of the map.



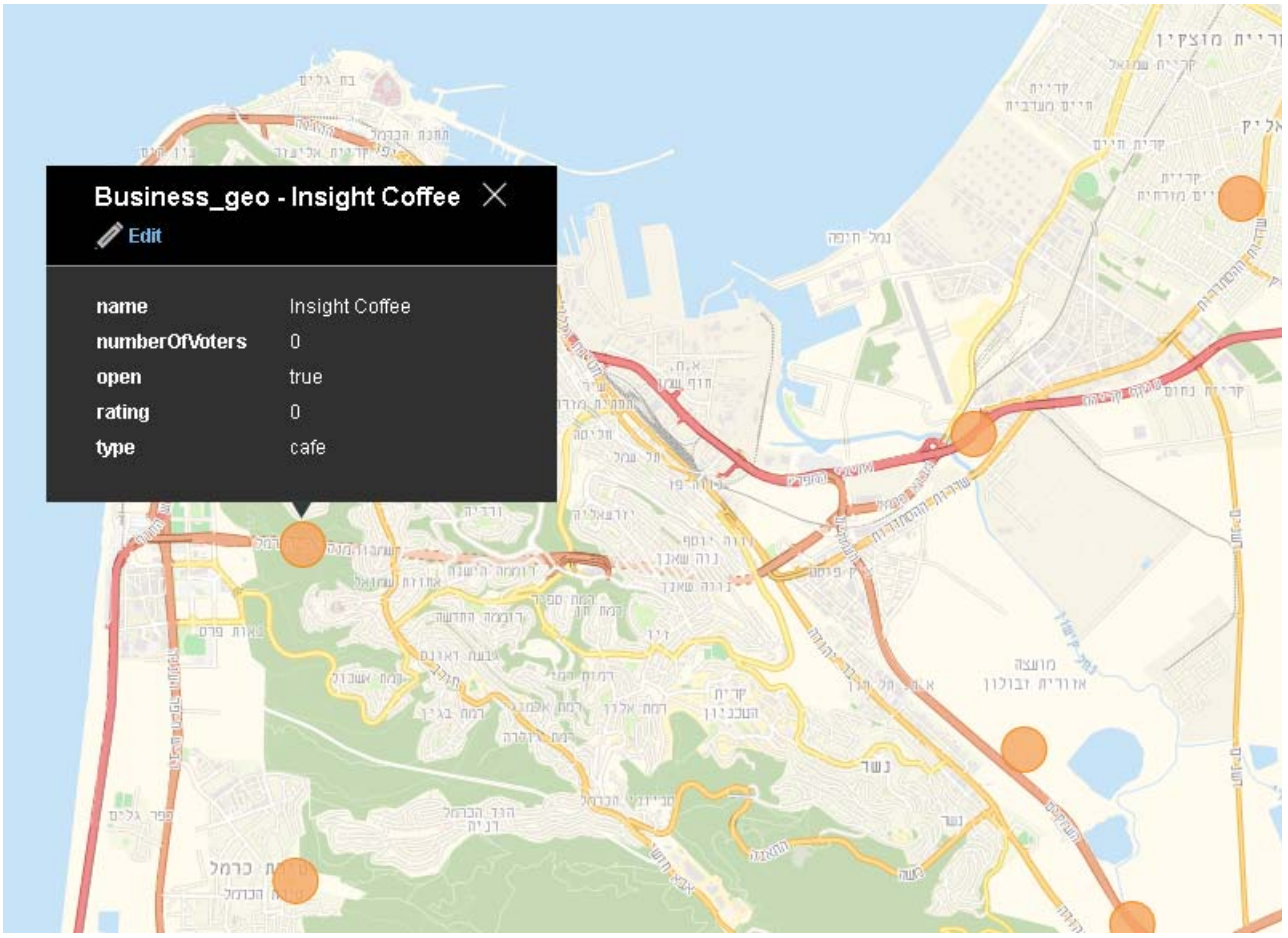
Information

By default, the map is loaded with OpenStreetMap (OSM) overlay. You can change the map viewer to MapQuest. From the upper-right drop menu, click **Choose Map > MapQuest**. The map viewer switches to use MapQuest overlay.



3.3. Viewing and customizing entities

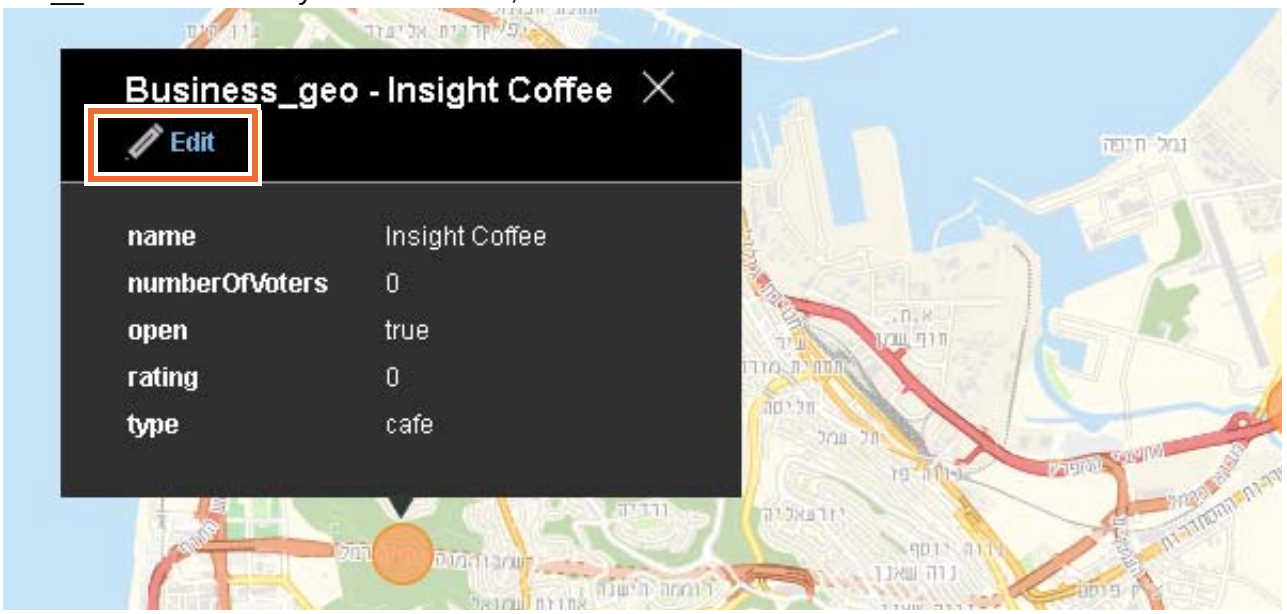
__ 1. Click any of the orange circles on the map to see the business name, type, and other attributes that are associated with the entity.



__ 2. Customize the display of a business entity.

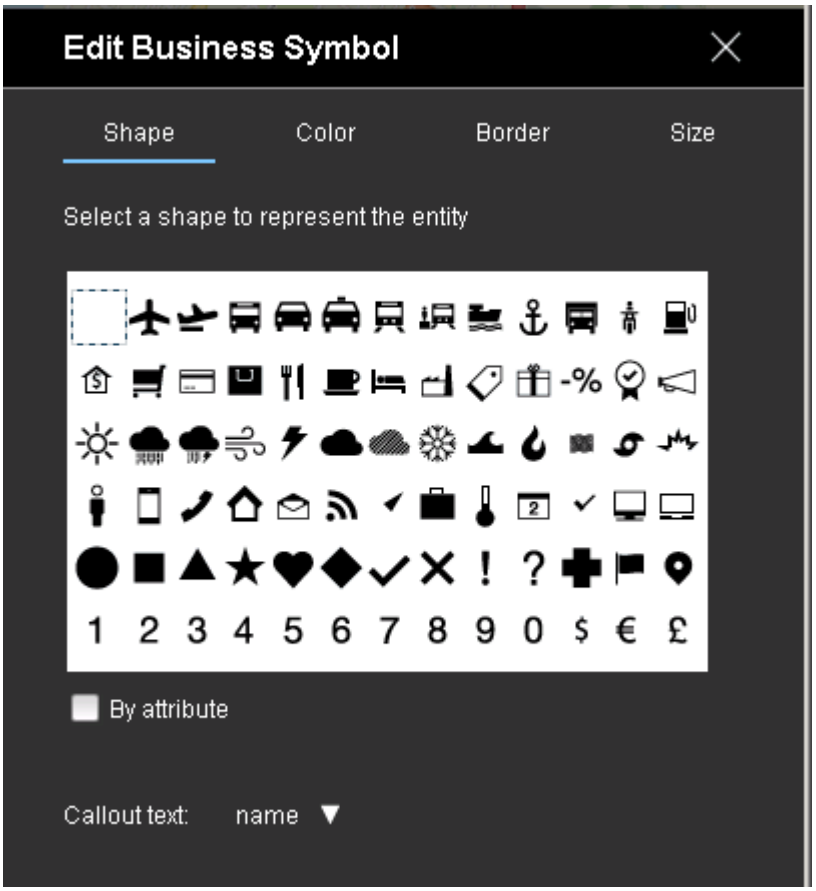
__ a. Click one of circles that presents a business, such as: Insight Coffee.

__ b. On the entity details window, click **Edit**.



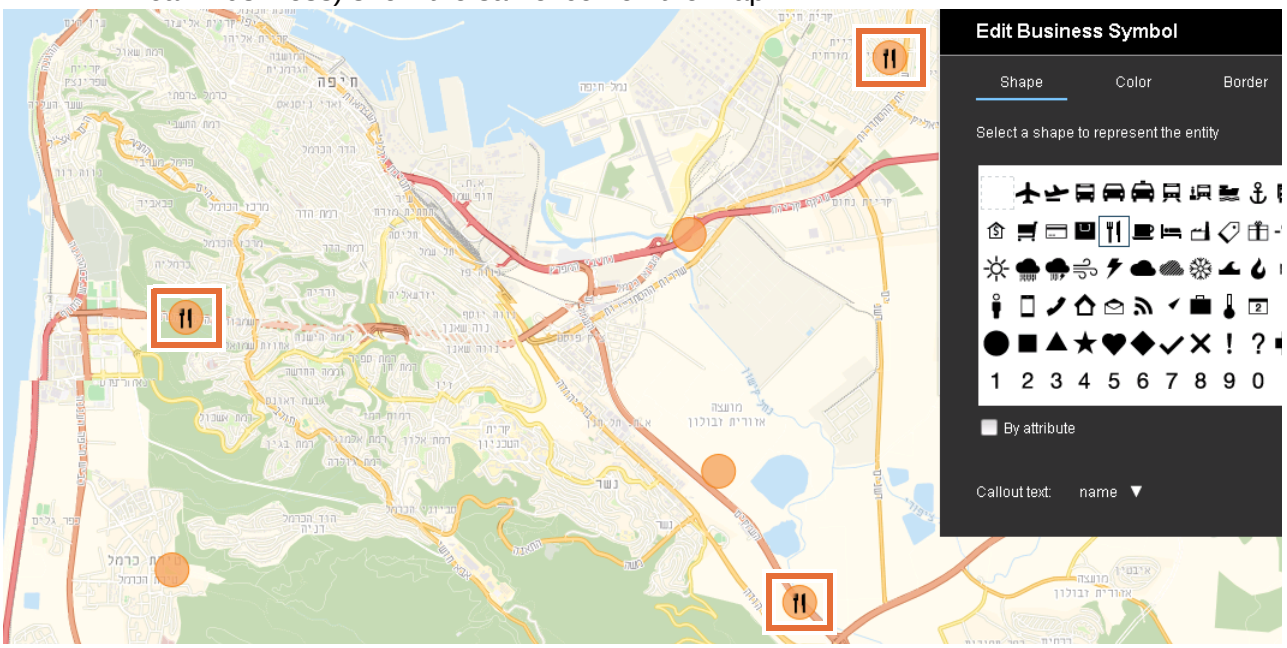
The Edit Business Symbol window opens on the right side of the map.

In the next steps, you go through each tab to select the shape, color, border, and size that correspond to the type of entity that you want to represent.



__ c. In the **Shape** tab, choose a shape to represent the business.

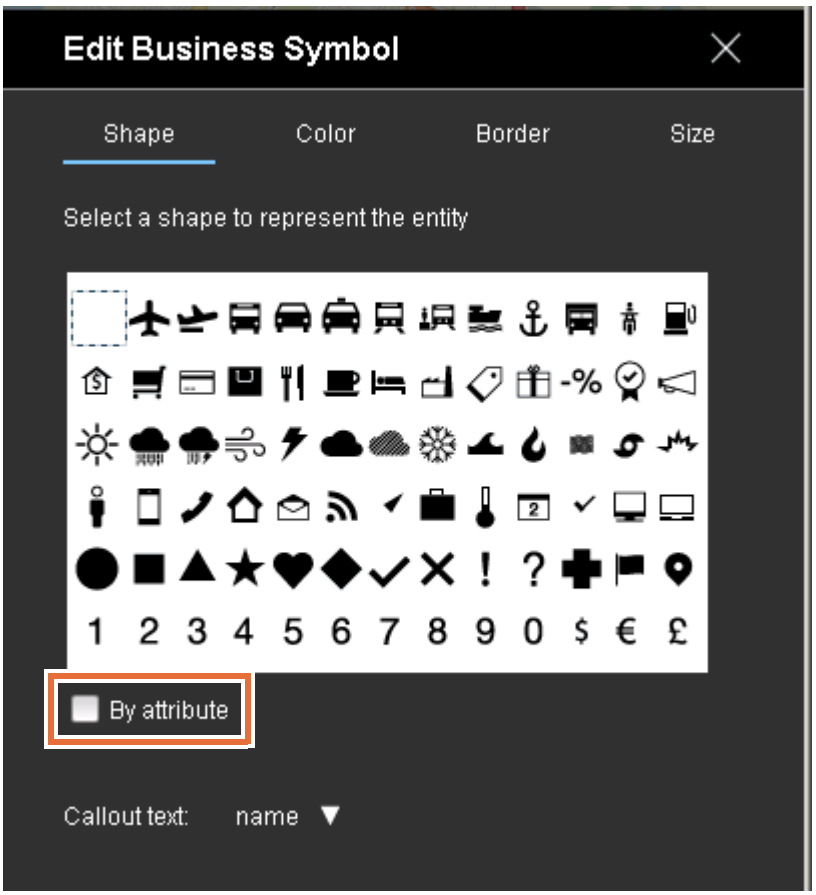
After you select an icon, all the entities with the same type (which, in this case, is `car.Business`) show the same icon on the map.



You can also define the shape based on a specific attribute of the entity.

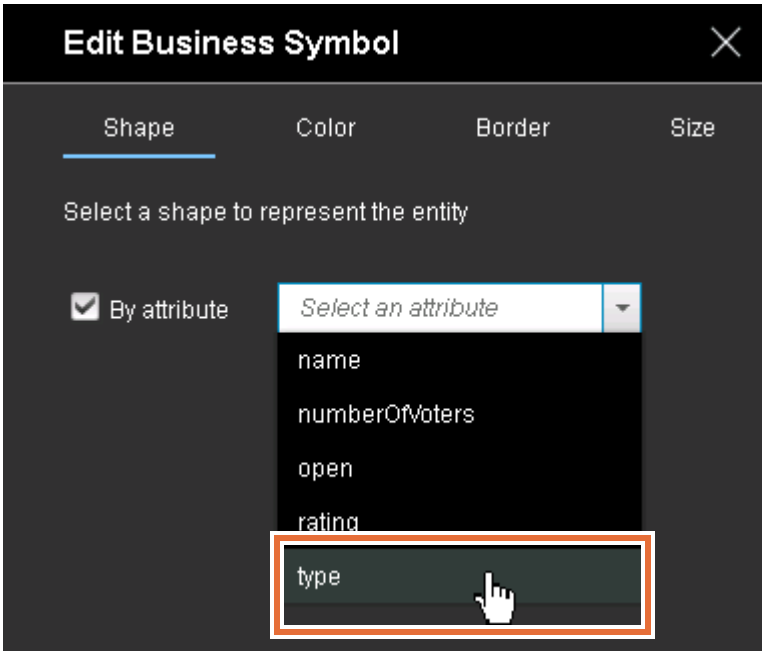
__ 3. To define a shape by entity attribute:

__ a. In the Edit Business Symbol window, click the **By attribute** check box.



The Shape options change from icons to an attribute menu.

__ b. Select the **type** attribute from the menu.



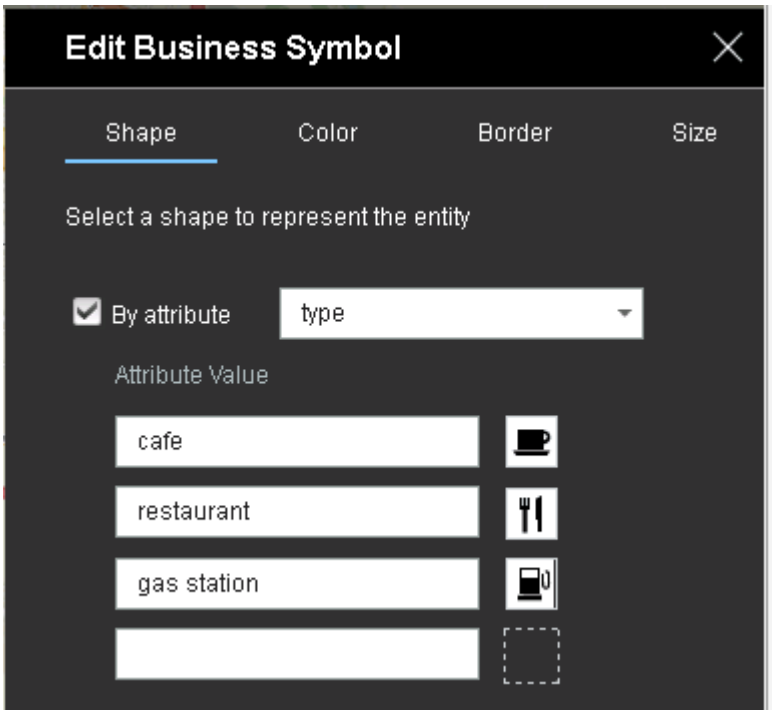
__ c. In the **Attribute Value** fields, enter:

- `cafe`
- `restaurant`
- `gas station`

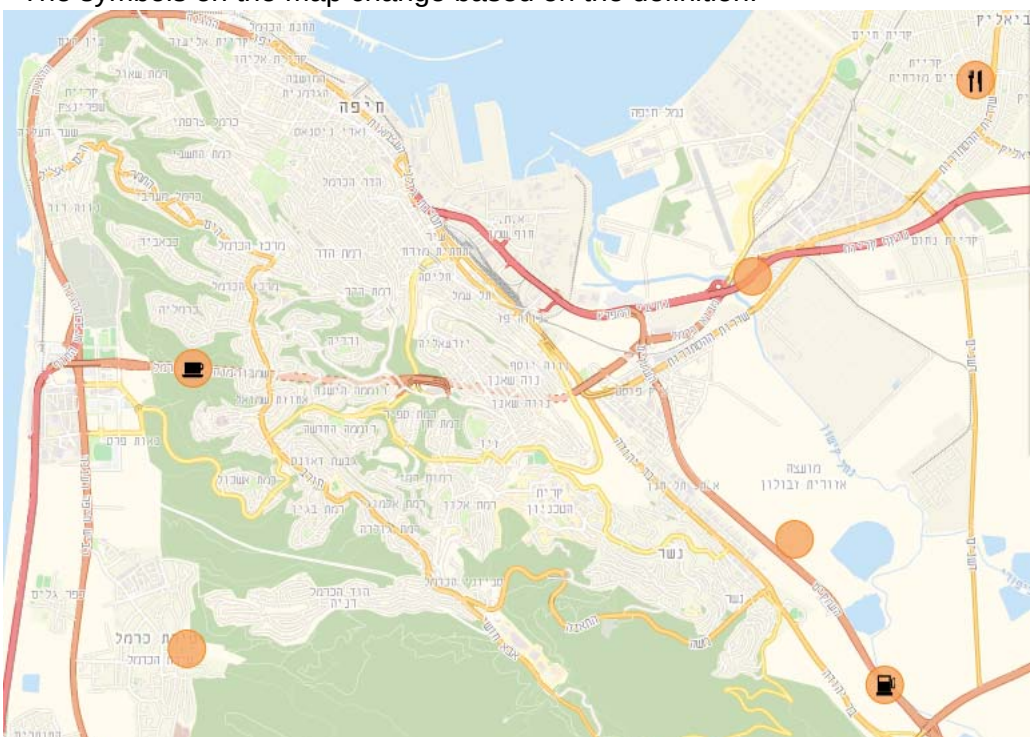
__ d. For each value, choose corresponding symbols by clicking the square next to the values field.

- cafe: Select the **coffee cup** symbol.
- restaurant: Select the **fork and knife** symbol.

- gas station: Select the **gas pump** symbol.

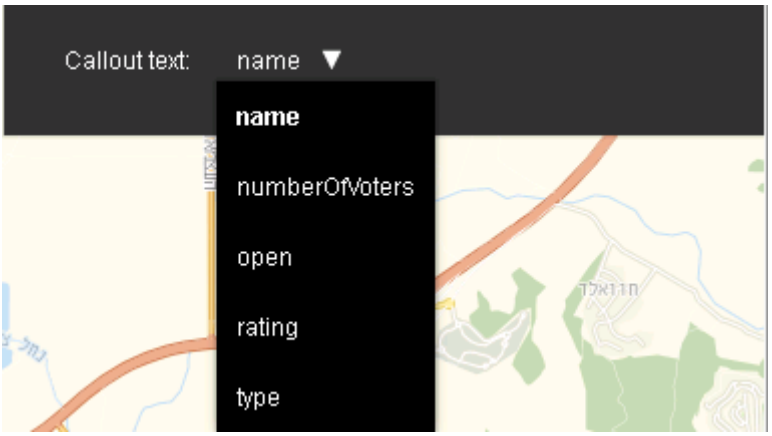


The symbols on the map change based on the definition.



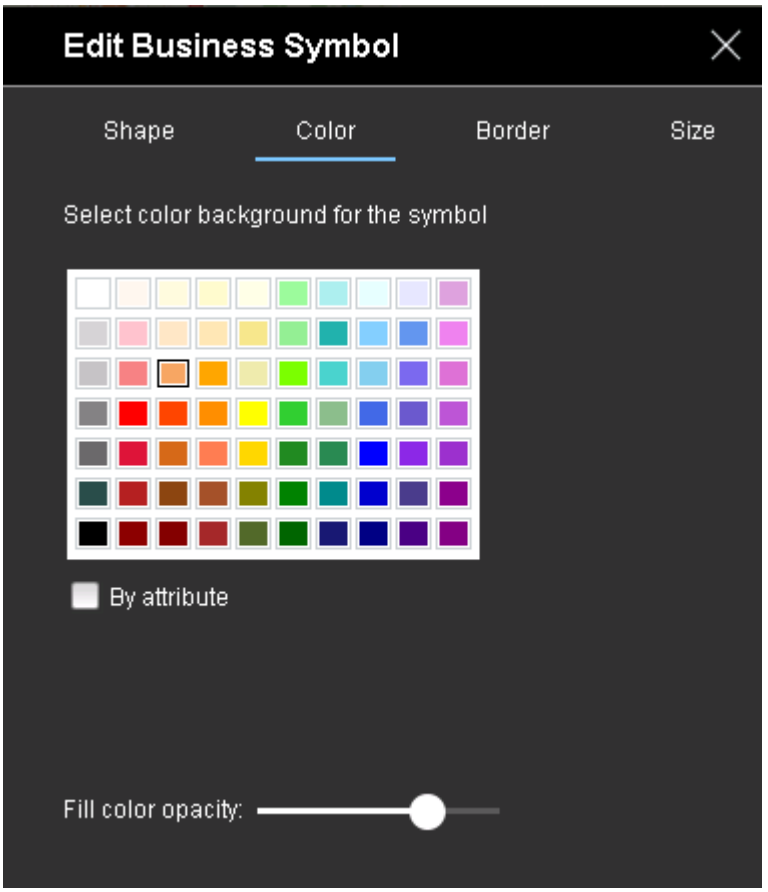
Information

You can use the **Callout text** menu at the bottom of the Edit Business Symbol window to choose the text that is shown when you hover the mouse pointer over the entity. By default, it displays the entity name.



__ 4. Click the **Color** tab.

You can replace the default orange color of the entities with a different color.



You can select a different color by clicking a color in the palette, or by setting a color by attribute.

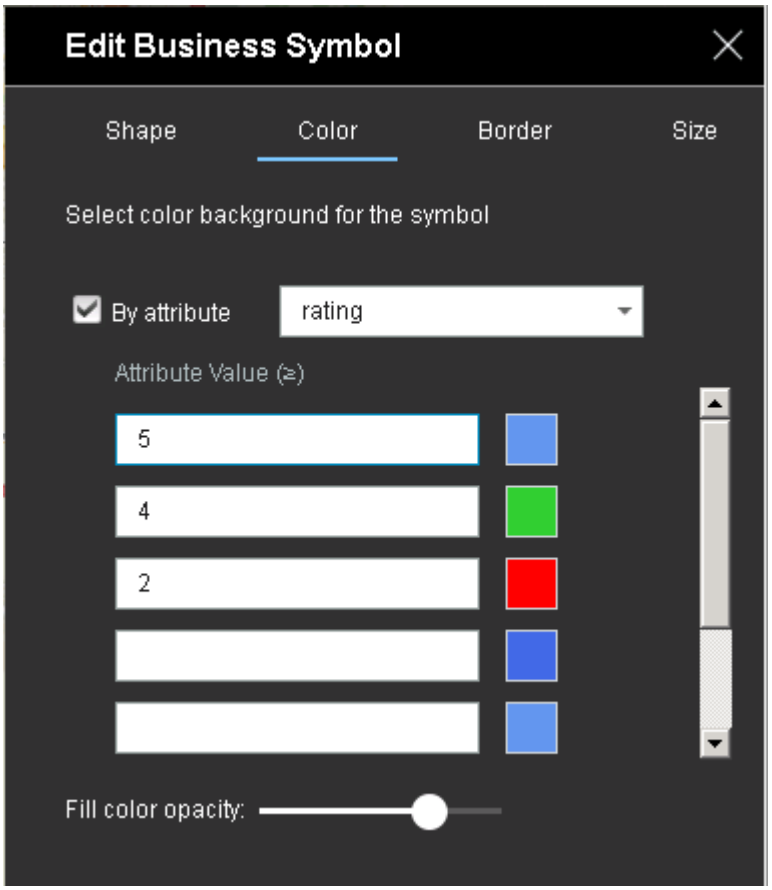
__ 5. To set color based on one of the attributes values:

__ a. Select the **By attribute** check box.

__ b. From the attribute selection menu, click **rating**.

__ c. In the **Attribute Value** fields, set the following values and colors:

Value	Color
5	Blue
4	Green
2	Red



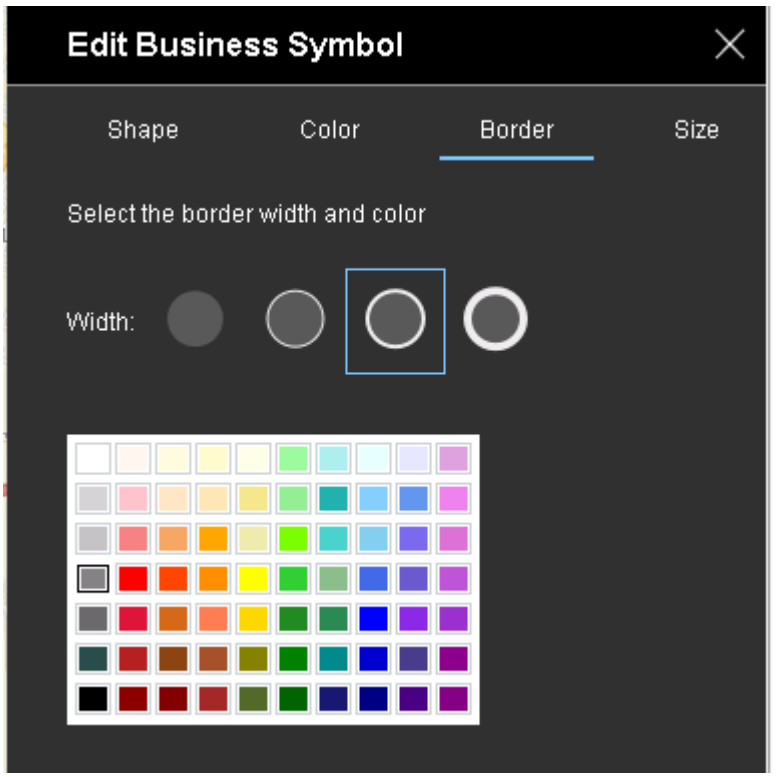
Information

You can also change the opacity of the fill color by using the **Fill color opacity** slider.

Troubleshooting

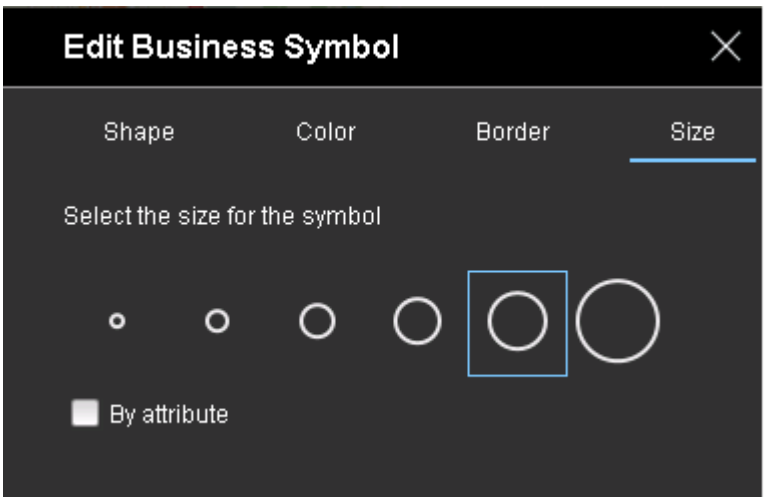
The color changes that you defined are not applied to the business entities on the map until you run the rate setting later in this exercise.

__ 6. Click the **Border** tab to select a border width and color of your choosing.



__ 7. Click the **Size** tab and select a size for all the symbols.

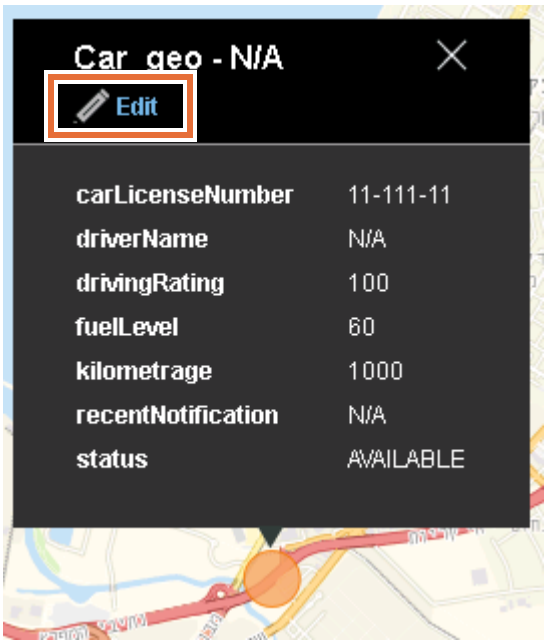
You can also define the symbol size based on a specific attribute value by selecting the **By attribute** check box.



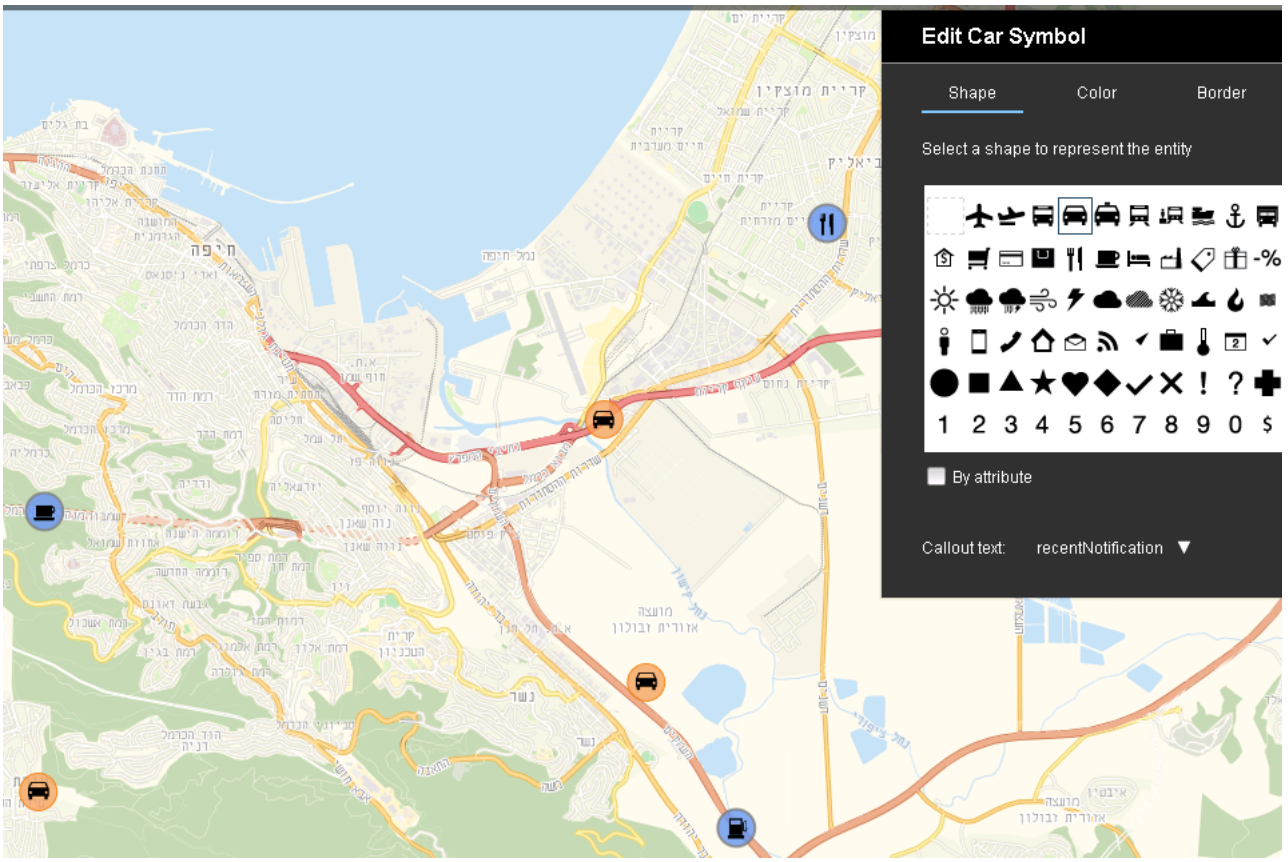
__ 8. Close the Edit Business Symbol window by clicking the **X**.

__ 9. Customize the car entity.

__ a. Click one of the cars on the map and click **Edit**.



__ b. In the Edit Car Symbol window, click the **Shape** tab and select a **car** icon.

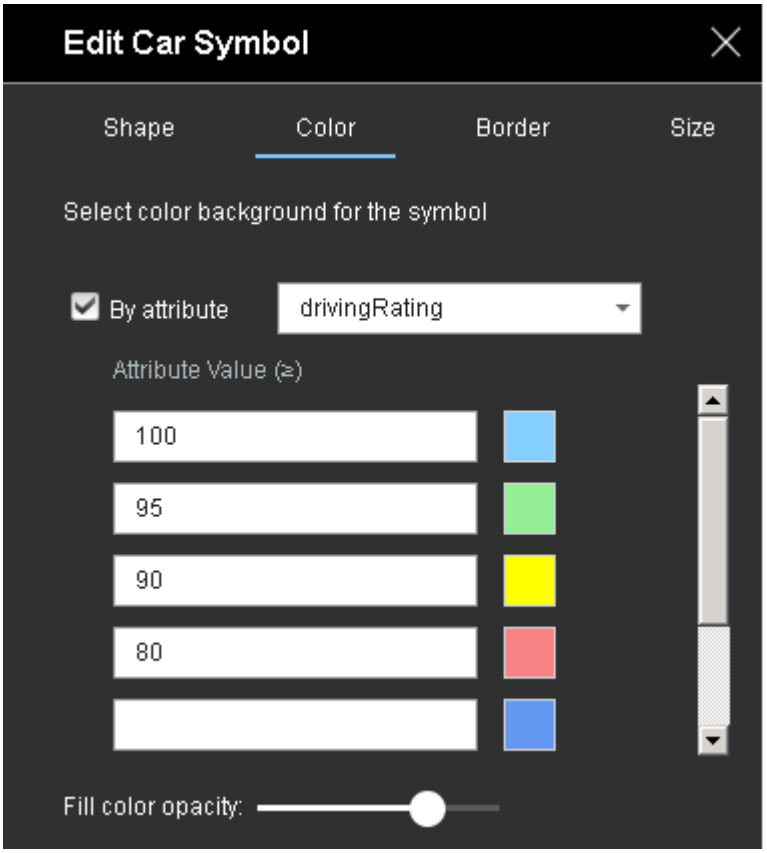


__ c. In the Edit Car Symbol window, click the **Color** tab and select **By attribute**.

__ d. Click the **Select an attribute** menu, and select the **drivingRating** attribute.

__ e. Enter the following rating range and the corresponding colors, which are based on the test scenario:

Value	Color
100	Blue
95	Green
90	Yellow
80	Red



Important

The range values must be entered in **descending** order. The evaluation of the criteria is done top-down, and the first matching pair wins.

__ f. (*Optional*) You can change the border and the size of the car symbol by selecting the different options on the **Border** and **Size** tabs.

__ g. Click the **X** to close the Edit Car Symbol window and save your changes.

You are now finished with customizing how the entities display on the map.

Section 4. Testing the solution

4.1. Run the rate setting

__ 1. Go back to Insight Designer, and make sure that you are in the Java perspective.

__ 2. If needed, click the **Package Explorer** to access the `car-rental-test` project.

__ 3. Right-click the **runRatings()** method of the `CarRentalTest` class and click **Run As > JUnit Test**.

__ 4. Wait for the test to complete.

__ 5. Switch back to the browser to see the different business ratings, which are marked by different colors:

- BBQ Restaurant rating: 5
- Awesome Gas rating: 4.5
- Insight Café rating: 2.5

4.2. Run the solution scenario

__ 1. Go back to Insight Designer and make sure that you are in the Java perspective.

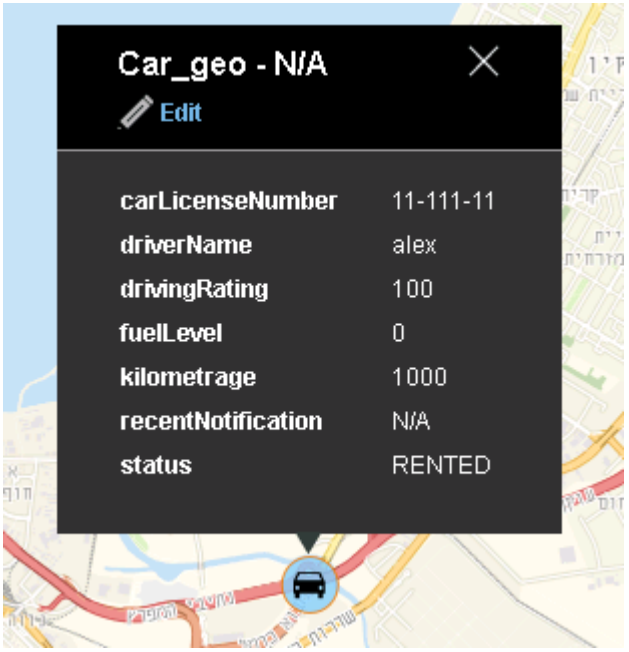
__ 2. If needed, click the **Package Explorer** to access the `car-rental-test` project.

__ 3. Right-click the **runDemo()** method of the `CarRentalTest` class, and click **Run As > JUnit Test**.

__ 4. Switch to the browser.

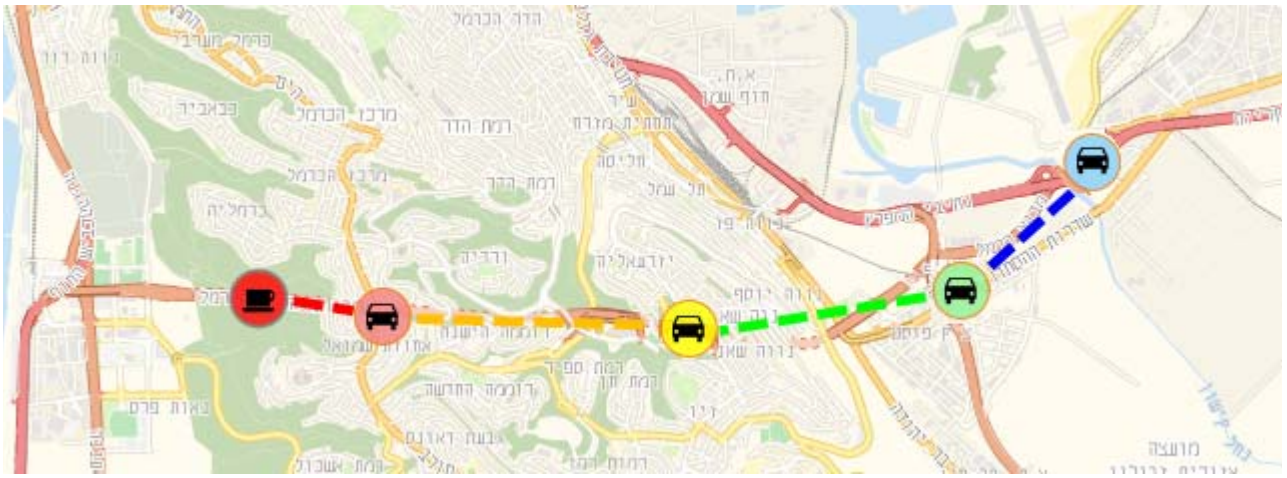
__ 5. Wait until you see that one of the cars, which has the license number **11-111-11**, is rented, and starts to move towards Insight Coffee.

__ 6. Click the moving car to see its attribute details.



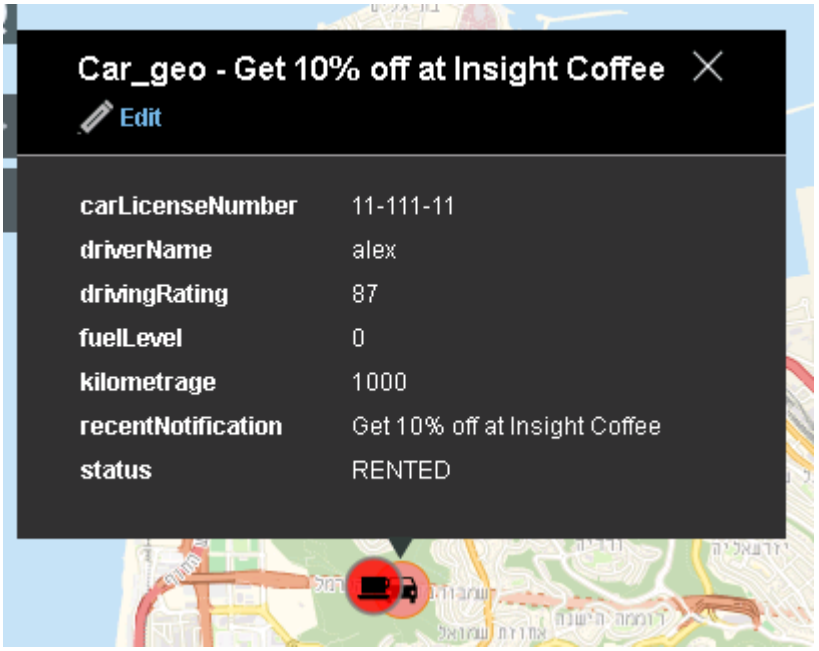
The `drivingRating` attribute is based on the speed of the car, and it is calculated while the car moves.

__ 7. Notice that the color of the car symbol changes when the value of the `drivingRating` attribute changes.

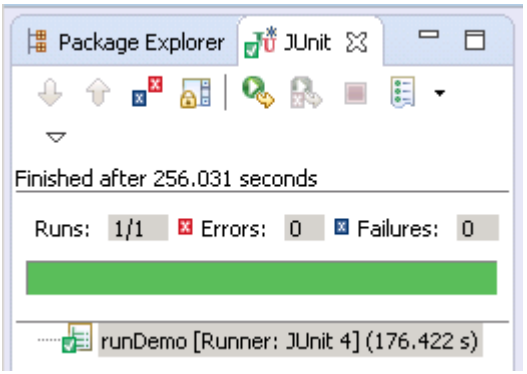


__ 8. Also, notice that when the car approaches Insight Coffee, a coupon is sent as a recent notification, which reads:

`Get 10% off at Insight Coffee`



__ 9. Go back to Insight Designer, and notice in the JUnit view that the demonstration ran successfully.



__ 10. (*Optional*) Run the test again.

__ a. Run the `prepare()` JUnit test again to reset everything to its initial state.

__ b. Rerun the `runRatings()` and `runDemo()` JUnit tests.

__ 11. After you finish testing, close the browser.

End of exercise

Exercise review and wrap-up

The exercise demonstrated how to work with Insight Map Viewer to better visualize moving geometries.

Exercise 12.Defining connectivity

Estimated time

00:30

Overview

This exercise shows how to define connectivity for a solution. In a later exercise, you deploy and test the connectivity.

Objectives

After completing this exercise, you should be able to:

- Configure inbound and outbound endpoints
- Generate and validate connectivity configurations

Introduction

In this exercise, you define connectivity for a solution.

This exercise includes these sections:

Requirements

This exercise has no requirements.

Section 1. Setting up your workspace

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

__ 1. In Insight Designer, switch to a new workspace and import the projects for this exercise:

__ a. From the **File** menu, click **Switch Workspace > Other**.

__ b. When prompted in the Workspace Launcher for a workspace, type a workspace path, such as:

`C:\labfiles\workspaces\connectivity`

__ c. Click **OK** to close the Workspace Launcher.

__ 2. Import the start 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 archive file** and click **Browse**.

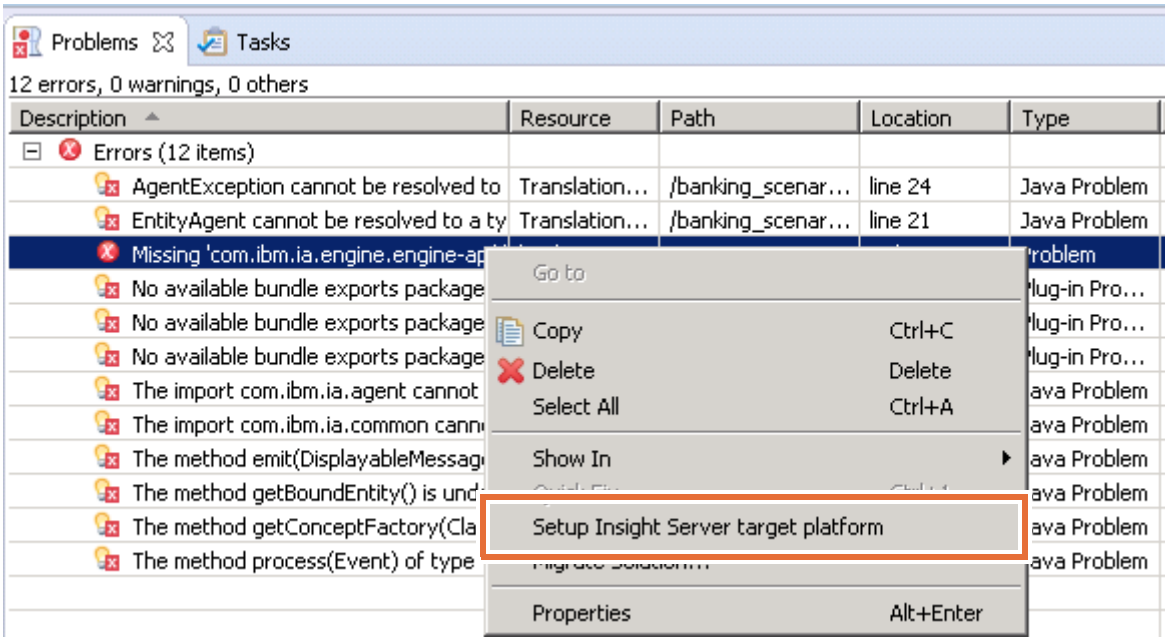
__ d. Go to the ***<LabfilesDir>*** and select the `workspace-final.zip` file and click **Open**.

__ e. Click **Finish** and wait for the workspace to build.

Your workspace now contains all the required projects.

__ 3. Resolve project errors.

__ a. In the Problems view, right-click any of the errors and click **Setup Insight Server target platform**.



__ b. Wait for the project to rebuild completely.

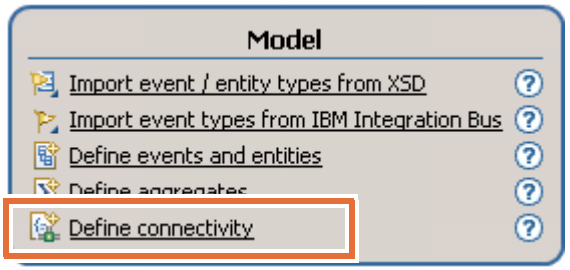
You should not see errors after the workspace is rebuilt.

Section 2. Creating connectivity definitions for the solution

In this section, you define the inbound and outbound endpoints and connectivity definitions for a solution.

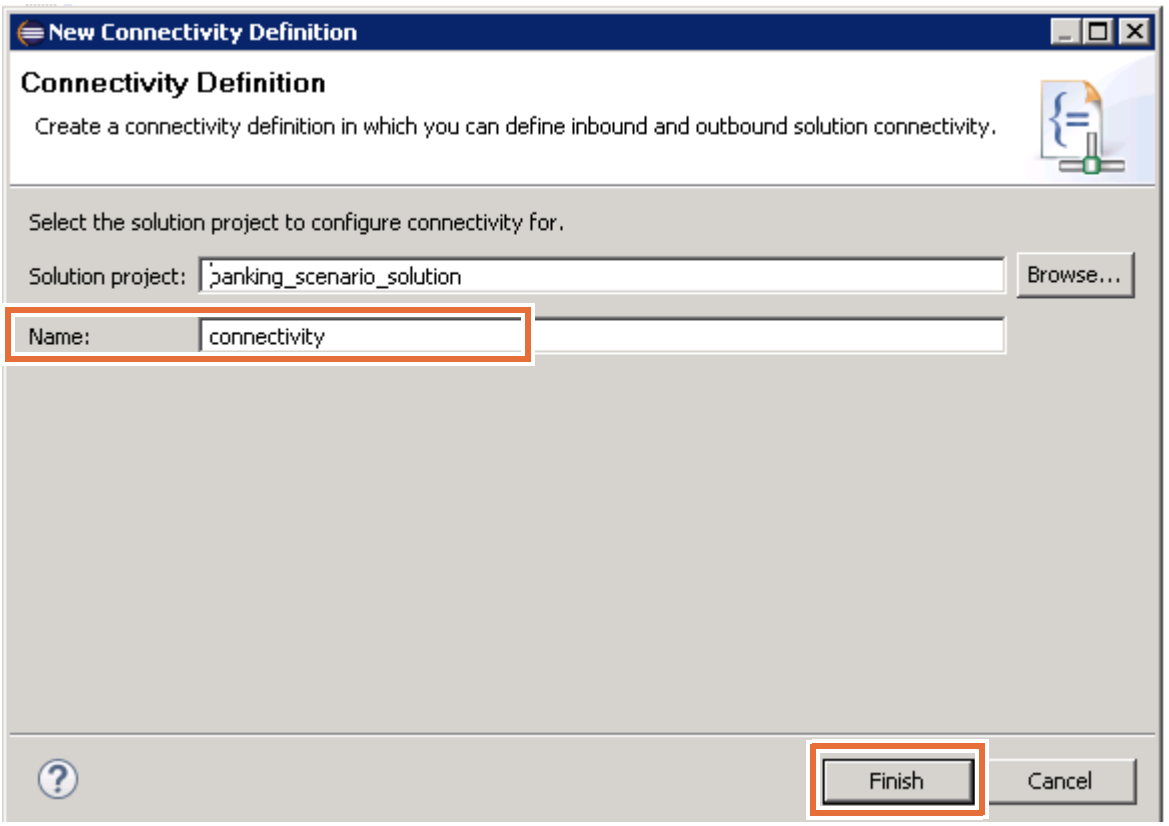
__ 1. In Solution Explorer, click **banking_scenario_solution** to open the **Solution Map** view.

__ 2. In the Model goal, click the **Define connectivity** link.

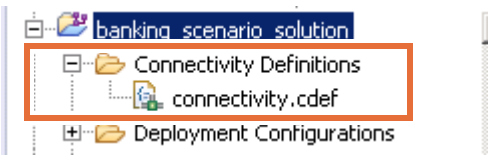


The New Connectivity Definition wizard opens.

__ 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 .
```

__ b. Starting on a new line, 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".
```

__ c. Starting on a new line, type the following text to define the outbound binding and endpoint.

```
define outbound binding 'message'
  with
    description "Outbound displayable message" ,
  using
    message format application/xml ,
    protocol HTTP ,
  delivering events :
    - displayable message .
define outbound HTTP endpoint 'notificationToClient'
  with
    description "Outbound message" ,
  using
    binding 'message' ,
    url "http://localhost:8081" .
```

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 ***<LabfilesDir>*\code** folder. After you paste the text, press Ctrl+Shift+F to format it.

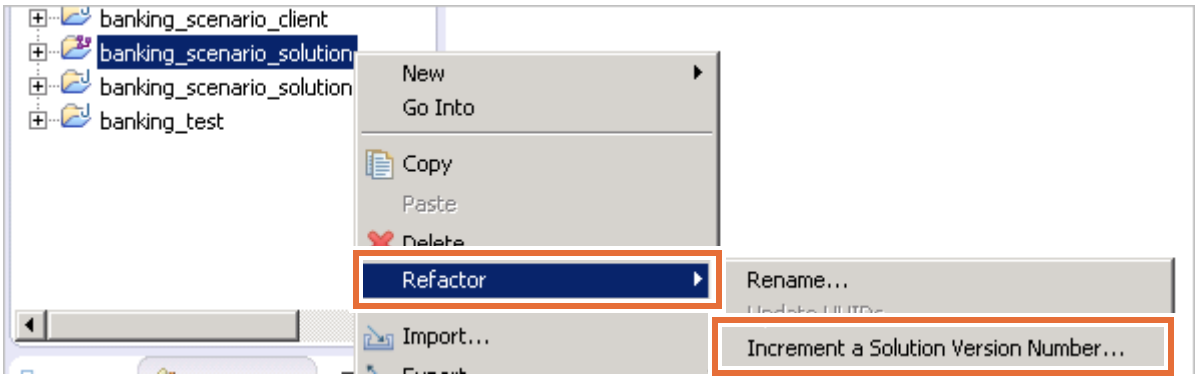
__ d. Save your work and close the file.

Section 3. Exporting a solution for deployment

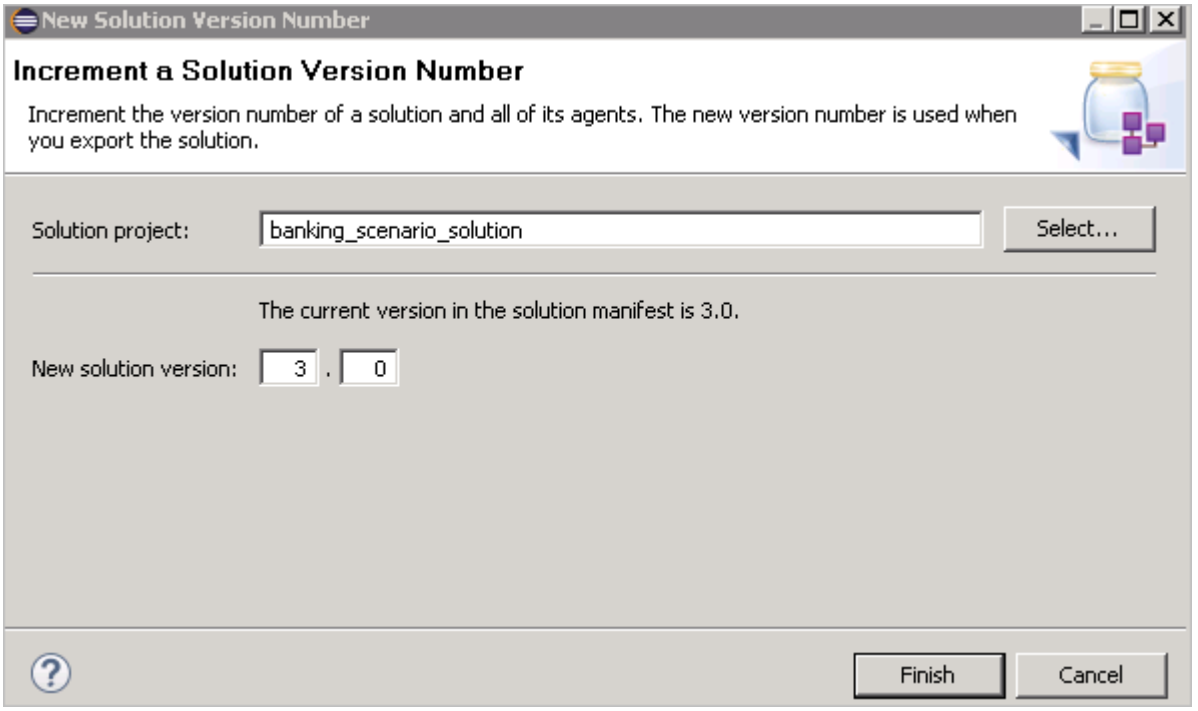
In this section, you export the solution archive so that it can be managed for deployment.

3.1. Increment the version number

__ 1. In Solution Explorer, right-click **banking_scenario_solution**, and click **Refactor > Increment a Solution Version Number**.



__ 2. In the **New solution version** field, increment the major version. For example, if the version was **2.0**, change it to **3.0**.



Note

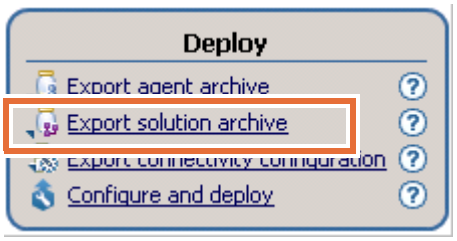
Your version number might be different from the screen capture.

__ 3. Click **Finish** and wait for the workspace to rebuild.

3.2. Export the solution archive

__ 1. In the Decision Insight perspective, make sure that the `banking_scenario_solution` project is selected with your mouse and open the **Solution Map** view.

__ 2. In the **Deploy** part of the Solution Map, click the **Export solution archive** link.

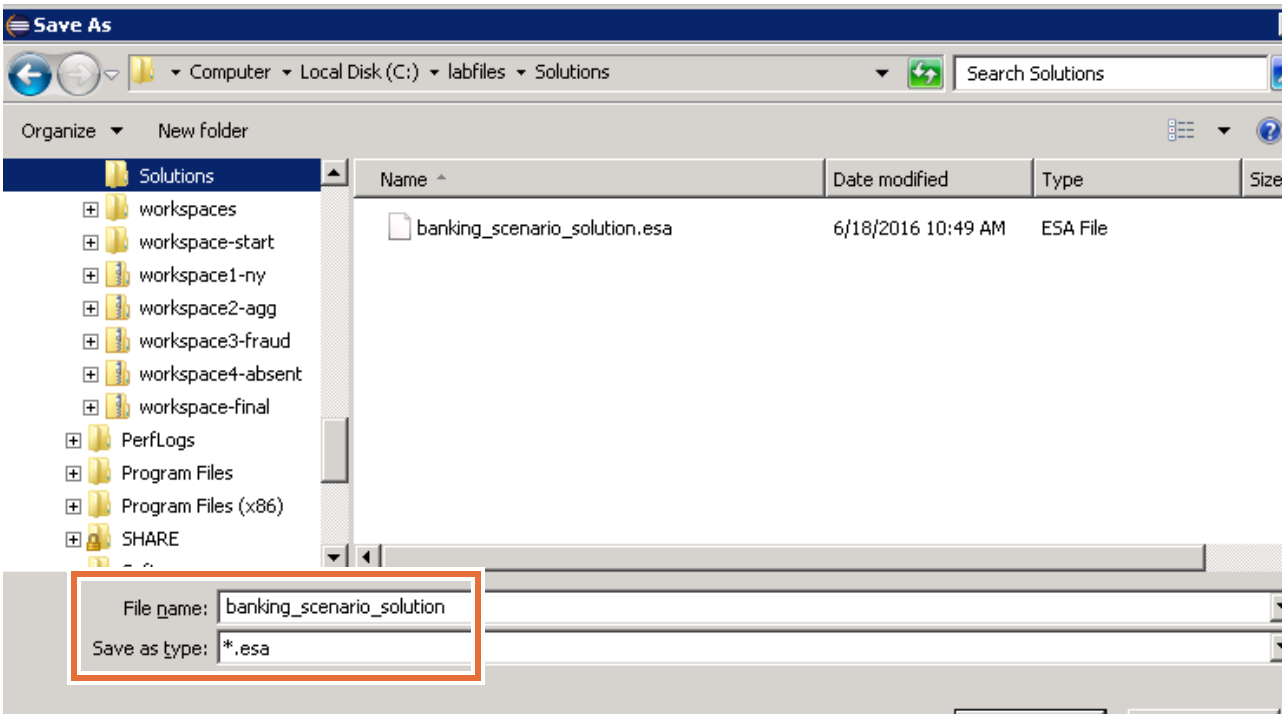


__ 3. In the Solution Archive Export wizard, click **Browse** to set the **Output file** field to:

`<LabfilesDir>\Solutions`

Where `<LabfilesDir>` is by default: `C:\labfiles`

__ 4. Remove the version number from the archive 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.

__ 5. Click **Finish**.

Section 4. Generating connectivity configurations

In this section, you use the `connectivityManager` script to generate and validate connectivity configurations.

4.1. Preparing the inbound application and configuration

__ 1. Generate an inbound application EAR.

__ a. Make sure that you are in the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ b. Type the following `connectivityManager` command.

```
connectivityManager generate application
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking_scenario_solution-inbound.ear
```

Hint

You can copy and paste the command lines from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder.

After generation finishes, you see a "Successfully generated" message.

```
C:\IBM\ODMInsights88\runtime\ia\bin>connectivityManager generate application C:
labfiles\Solutions\banking_scenario_solution.esa C:\labfiles\Solutions\banking_
cenario_solution-inbound.ear
CWMBE1146I: Reading the input file: C:\labfiles\Solutions\banking_scenario_solu
ion.esa
CWMBE1148I: Writing to the output file: C:\labfiles\Solutions\banking_scenario_
olution-inbound.ear
CWMBE1474I: Successfully generated the solution inbound connectivity applicatio
 file: C:\labfiles\Solutions\banking_scenario_solution-inbound.ear
```

__ 2. Generate the XML configuration file for deploying inbound connectivity by typing this command.

```
connectivityManager generate config
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-inbound-config.xml
--inboundEndpoints="*"
```

After generation finishes, you see a “Successfully generated” message.

```
C:\IBM\ODMInsights88\runtime\ia\bin>connectivityManager generate config C:\labf
les\Solutions\banking_scenario_solution.esa C:\labfiles\Solutions\banking-serve
-inbound-config.xml --inboundEndpoints="*"
CWMBE1146I: Reading the input file: C:\labfiles\Solutions\banking_scenario_solu
ion.esa
CWMBE1491I: Generated a template for the missing resource "banking1" used by th
 endpoint "banking1" of the solution "banking_scenario_solution" in the file "C
\labfiles\Solutions\banking-server-inbound-config.xml".
CWMBE1494I: Successfully generated  template solution connectivity configurati
n file "C:\...g-server-inbound-config.xml" for the soluti
n "banking_scenario_solution".
```

__ 3. Edit the newly generated configuration file to edit it.

__ a. In the `C:\labfiles\Solutions` directory, right-click the `banking-server-inbound-config.xml` file and open it with Notepad++.

__ b. Uncomment the application section by deleting the comments around the application definition at the top of the file.

__ c. Configure the `iaEventSubmitter` role to map to the admin user by replacing this line:

```
<security-role name="iaEventSubmitter"/>
```

With these lines:

```
<security-role name="iaEventSubmitter">
  <user name="admin"/>
</security-role>
```

__ d. At the bottom of the file, uncomment the HTTP endpoint definition.

The endpoint configuration is now ready for deployment.

```
<?xml version="1.1" encoding="utf-8"?><server>
<!--Application definition for inbound connectivity application for solution:

  <application location="banking_scenario_solution-inbound.ear"
      <application-bnd>
          <security-role name="iaEventSubmitter">
              <user name="admin"/>
          </security-role>
      </application-bnd>
  </application>

<!--Generated configuration for endpoint: banking1-->

  <ia_inboundHttpEndpoint endpoint="banking_scenario_solution/banking1" />
</server>
```

__ 4. Save and close the file.

4.2. Preparing the outbound configuration

__ 1. Generate a configuration to deploy on the outbound server by returning to the command prompt and typing this command.

```
connectivityManager generate config
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-outbound-config.xml
--outboundEndpoints="*"
```

Hint

You can copy and paste the command lines from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder.

__ 2. Edit the newly generated configuration file to edit it.

__ a. In the <*LabfilesDir*>\Solutions directory, open the banking-server-outbound-config.xml file (with Notepad++).

__ b. Uncomment the HTTP endpoint definition by deleting the comments around the application definition at the top of the file.

```
<!--Generated configuration for endpoint: notificationToClient-->

    <ia_outboundHttpEndpoint endpoint=
    "banking_scenario_solution/notificationToClient" />
</server>
```

__ c. Save and close the file.

4.3. Validating the connectivity configurations

Hint

You can copy and paste the command lines from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder.

__ 1. Validate the inbound configuration by returning to the command prompt and typing the following command:

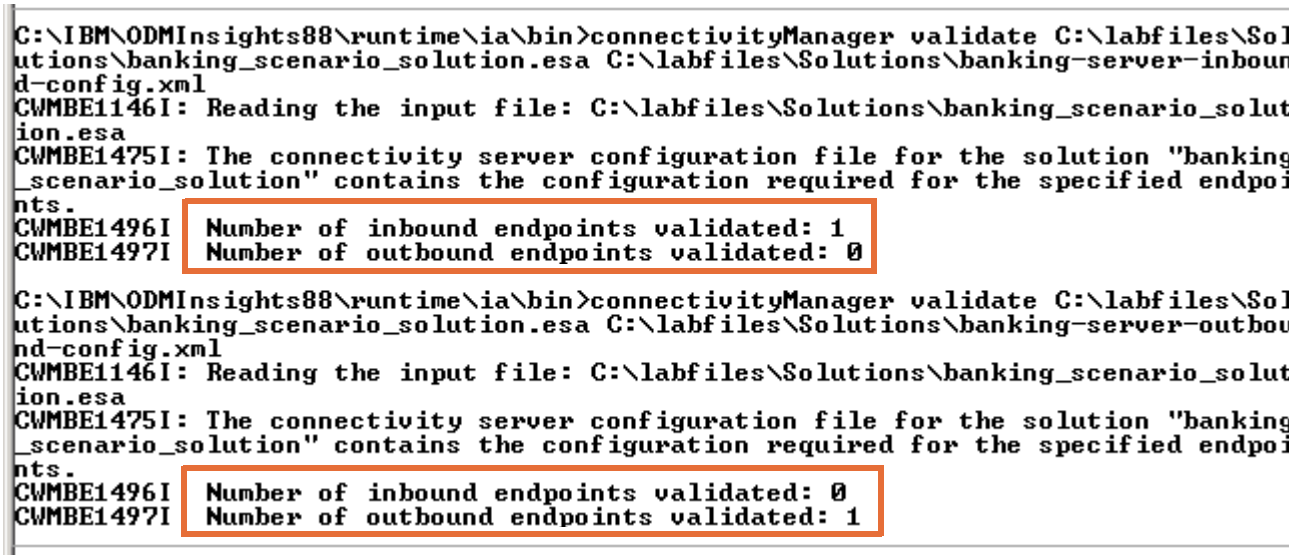
```
connectivityManager validate
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-inbound-config.xml
```

The result shows a validated inbound endpoint.

__ 2. Validate the outbound configuration by typing the following command:

```
connectivityManager validate
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-outbound-config.xml
```

After running the validation commands, the result shows one validated inbound endpoint and one validated outbound endpoint.



End of exercise

Exercise review and wrap-up

In this exercise, you defined connectivity for a solution and validated the connectivity configurations.

Exercise 13.Installing Decision Server Insights

Estimated time

01:00

Overview

In this exercise, you learn how to install Decision Server Insights on multiple hosts.

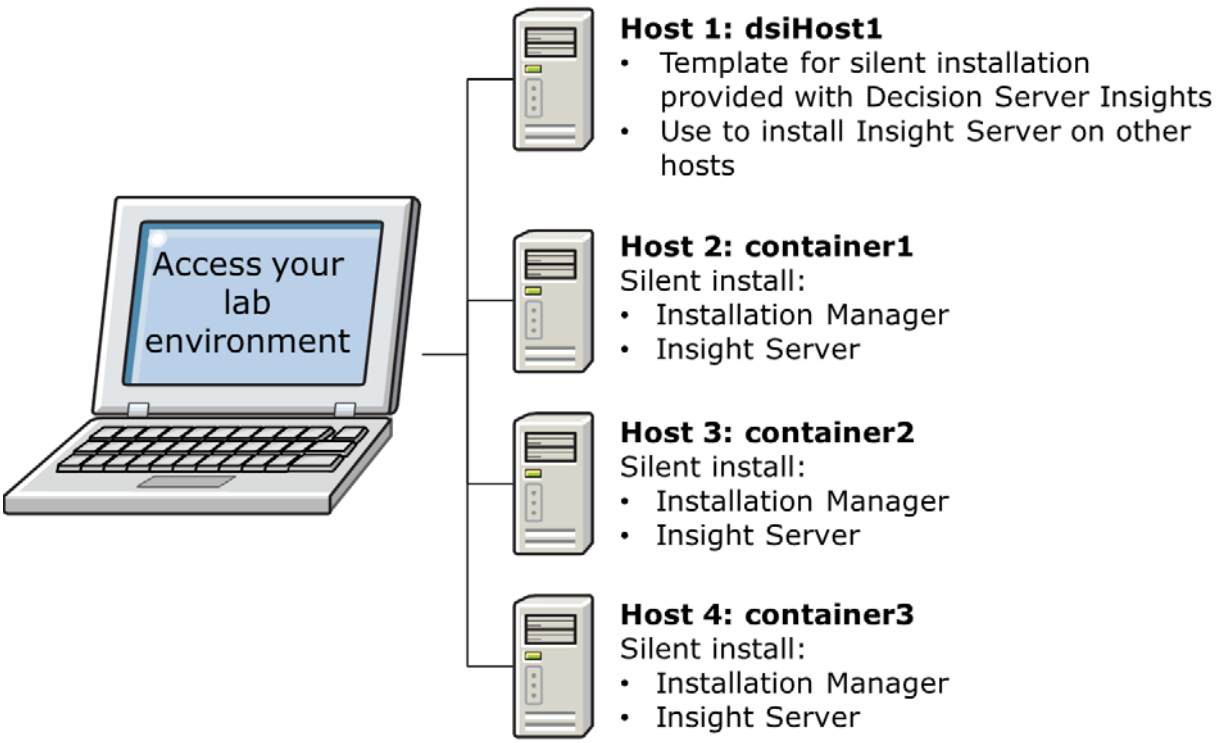
Objectives

After completing this exercise, you should be able to:

- Install IBM Installation Manager on every machine in the lab environment where Insight Server should run
- Use Installation Manager to install Decision Server Insights on every machine in the lab environment where Insight Server should run

Introduction

In this exercise, you install Decision Server Insights on each of the hosts that are in your environment.



Attention

The default host names are: **dsiHost1**, **container1**, **container2**, and **container3**. If you are in a classroom setting with multiple sets of hosts for students, your hosts might be renamed to other unique host names.

Make sure that you know the host names of the virtual images that you are using and that you use the ***actual*** host name *or* the IP address during the exercises.

This exercise includes these sections:

Requirements

This exercise requires that:

- All other servers must be stopped (make sure that the sample server is not running)
- Decision Server Insights must be installed on the main host (dsiHost1)
- Decision Server Insights installation files must be on all the hosts

For this exercise, you start on dsiHost1. You also work with the other three workstations: container1, container2, and container3.

Attention

The exercise instructions assume that your machines use the default host names. However, if your hosts use other unique names, make sure that you use the ***actual*** host name during the exercises.

Section 1. Preparing the Decision Server Insights template for silent installation on multiple hosts

You can run a silent installation of Decision Server Insights with IBM Installation Manager. Before you run the silent installation, you must prepare the installation template that is provided for Decision Server Insights.

Information

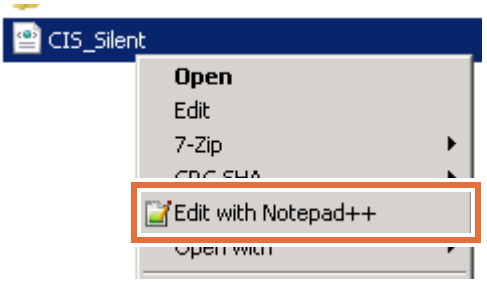
For this exercise, you installed Decision Server Insights on Host 1: dsiHost1. After you install Decision Server Insights on a host, the template for silent installation is provided in the `<InstallDir>\doc\silent` directory.

To prepare the CIS_Silent.xml template

__ 1. Open the `C:\IBM\ODMInsights88\doc\silent` directory.

__ 2. Copy the `CIS_Silent.xml` file to the `C:\SHARE` directory.

__ 3. Right-click the `CIS_Silent.xml` file, and click **Edit with Notepad++** to open the installation template.



Note

If you get update messages for Notepad++, you can ignore them.

In the template, you define the following settings:

Template placeholders	Description
`'!CIS_REPOSITORY!'`	Directory location where the installation files are stored.
`'!CIS_HOME!'`	Directory location to use as the installation path.

The placeholders are delimited with the exclamation (`!`) character. You replace them with the actual value.

Here you see the placeholders that are highlighted in the template.

```
<!--
################################################################################
All repositories are listed here.
A repository can be either a local location or a live repository.
################################################################################
-->
<server>
    <repository location='!CIS_REPOSITORY!'/>
</server>

<!--
################################################################################
This profile node defines where Operational Decision Manager will be installed
################################################################################
-->
<profile id='Decision Server Insights V8.8.0' installLocation='!CIS_HOME!'>
    <data key='eclipseLocation' value='!CIS_HOME!'/>
    <data key='cic.selector.nl' value='fr, es, it, en, de, nl, pl, pt_BR, ru, ja, ko, zh, zh_TW'/>
    <data key='user.prod.cis' value='true'/>

</profile>

<!--
################################################################################
This installation node directs the IM installer to install IM-based offerings.
################################################################################
-->
<install modify='false'>
    <offering id='com.ibm.websphere.cis.ia.v88' profile='Decision Server Insights V8.8.0' features=
    'base,com.ibm.cis.runtime.feature' installFixes='none'/>
```

__ 4. In the <server> section of the template, set the CIS_REPOSITORY value to the location of the **disk5** (Decision Server Insights) installation folder, as you see here.

```
<server>
  <repository
location='C:\labfiles\ODMinstallers\disk5\DecisionServerInsights'/>
</server>
```

__ 5. Set the two instances of the CIS_HOME value to the directory where you want to install Insight Server, as you see here.

```
installLocation='C:\IBM\ODMInsights88'>

<data key='eclipseLocation' value='C:\IBM\ODMInsights88'/>
```

__ 6. Save the file as CIS_Silent.xml in the C:\SHARE directory.

Note

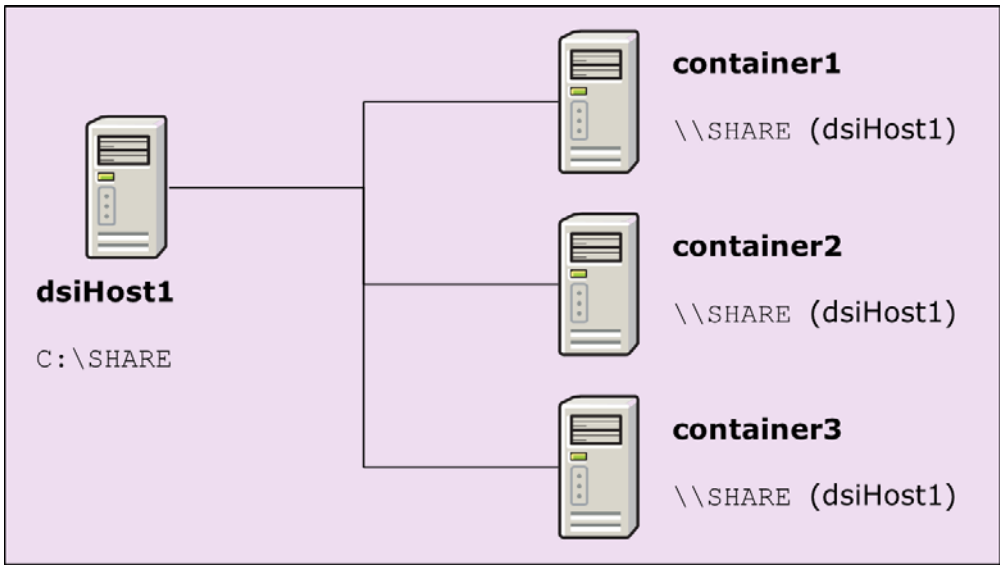
Make sure that you set all the placeholders. You can use the CIS_Silent.xml file in the ***<LabfilesDir>*\code** directory to compare your settings for this exercise.

Section 2. Verifying access to the remote hosts

Before running the silent installation on the remote hosts, you must copy the Decision Server Insights silent installation template to each host.

To transfer files from one virtual machine to another, you use the shared directory: `C:\SHARE`

The `C:\SHARE` folder is on your main host (dsiHost1). The other machines have a drive that is mapped to the **SHARE** folder.



Attention

The default host names are: **dsiHost1**, **container1**, **container2**, and **container3**.

Make sure that you know and use the host names and IP addresses that are assigned to your virtual images during the exercises.

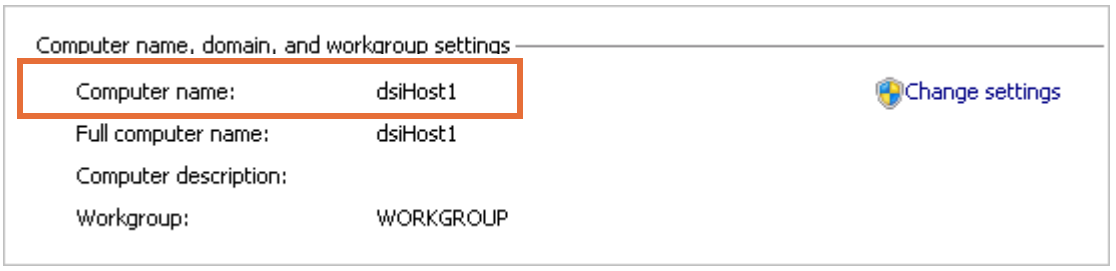
You can use the Appendix A, "Host names and IP addresses" as a reference for the host names and IP addresses that are assigned to your host.

2.1. Verifying your host name

To verify the host name:

__ 1. On the desktop, double-click the **Computer** icon and click **System properties**.

__ 2. In the **Computer name, domain, and workgroup settings** section, note the value for **Computer name**.



Stop

The default host name for the “main” host is **dsiHost1**. Your “main” host might have a different name. If the computer name is not dsiHost1, take note of the actual computer name and use that name whenever the exercise instructions require the host name.

Troubleshooting

If you run into issues with the host name, you can also rename your host to a unique name. Changing host names also requires remapping drives on the remote hosts to the shared directory.

For the steps to make these changes, see Appendix B, "Changing host names and mapped drives".

Troubleshooting

If you are unable to access the `\\SHARE` directory on the container host, you might need to remap a drive to your main host.

For the steps to map a drive, see Appendix B, "Changing host names and mapped drives".

Section 3. Running the silent installation on the remote hosts

In this section, you use command lines to install Installation Manager. After Installation Manager is installed, you use Installation Manager to silently install Insight Server.

For this exercise, you install both Installation Manager and Decision Server Insights on one machine at a time. The commands that you use here can be added to a script for further automation.

Note

For this section, if your environment can support opening multiple hosts, you can run the installation steps simultaneously on the container hosts. Otherwise, you can install each host in sequence.

__ 1. Switch to the container1 workstation (or the unique name for your “container 1” host.)

Important

If your environment can support running multiple hosts, you can also open the container2 and container3 workstations.

__ 2. Copy the `CIS_Silent.xml` file from the `\\SHARE` directory to the `C:\labfiles` directory.

__ 3. Install Installation Manager.

__ a. Open a command prompt window and change the current directory to: `C:\labfiles\ODMinstallers\disk1\IM64`

```
cd C:\labfiles\ODMinstallers\disk1\IM64
```

Note

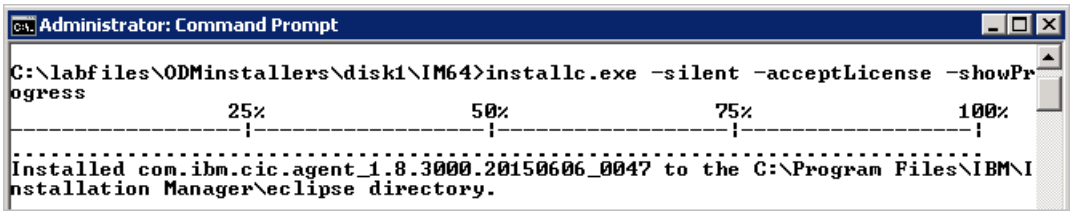
You can copy and paste the text for the command lines in this exercise from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder on the main host.

To make the `dsi.txt` file accessible to the remote hosts, you can place it in the **`C:\SHARE`** folder.

__ b. Run the silent installation of Installation Manager by typing this command.

```
installc.exe -silent -acceptLicense -showProgress
```

After the Installation Manager is installed on each machine, you see a message that identifies the installation path.



__ 4. Install Insight Server on each machine.

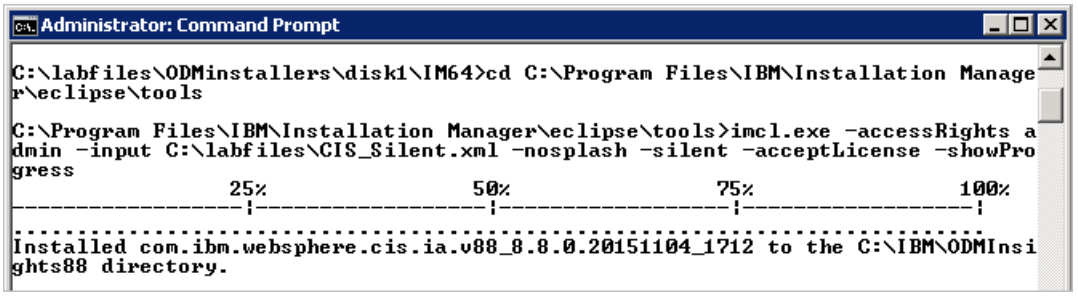
__ a. On container1, in the command prompt window, open the directory where you installed Installation Manager: `C:\Program Files\IBM\Installation Manager\eclipse\tools`

`cd C:\Program Files\IBM\Installation Manager\eclipse\tools`

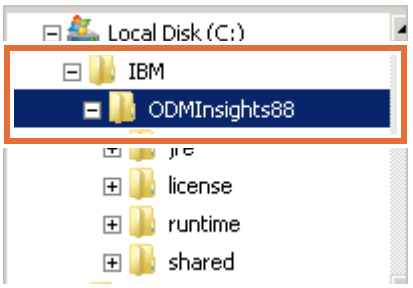
__ b. To install the Decision Server Insights Insight Server, type the following command:

`imcl.exe -accessRights admin -input C:\labfiles\CIS_Silent.xml -nosplash -silent -acceptLicense -showProgress`

After the silent installation is complete, you see a message that identifies the installation path.



__ 5. Refresh the view in Windows Explorer to verify that Insight Server was installed in the directory that you specified in the installation template.



__ 6. Repeat the steps in this section to install Installation Manager and Decision Server Insights Server this section on container2 and container3.

End of exercise

Exercise review and wrap-up

In this exercise, you installed the Decision Server Insights Insight Server on multiple hosts.

Exercise 14. Configuring Decision Server Insights

Estimated time

03:00

Overview

In this exercise, you learn how to configure Insight Servers on multiple hosts to create a grid.

Objectives

After completing this exercise, you should be able to:

- Create and configure catalog, container, and inbound and outbound servers

Introduction

This exercise includes these sections:

Requirements

This exercise requires that Decision Server Insights be installed on dsiHost1, container1, container2, and container3. You must also have a mapped drive from the container hosts to dsiHost1.

For this exercise, you start on your main host (dsiHost1). You also work on the container hosts.

Attention

The default host names are: **dsiHost1**, **container1**, **container2**, and **container3**. If you are in a classroom setting with multiple sets of hosts for students, your hosts might be renamed to other unique host names.

Make sure that you know the host names of the virtual images that you are using and that you use the ***actual*** host name during the exercises.

Section 1. Creating catalog servers

In this section, you create and customize catalog servers. After you create the server prototype, you modify the `bootstrap.properties` file and the `server.xml file` for each of the catalog servers. You customize the ports that are used by each server and you make sure that each server is aware of the other servers. You also enable majority quorum.

This section includes these steps:

1.1. Creating the catalog servers

__ 1. Make sure that you are on dsiHost1.

__ 2. In the command prompt window, type the following command to change directories to the `wlp\runtime` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 3. Create the first catalog server, cisCatalog1, by typing this command:

```
server create cisCatalog1 --template=cisCatalog
```

__ 4. Create cisCatalog2:

```
server create cisCatalog2 --template=cisCatalog
```

__ 5. Create cisCatalog3:

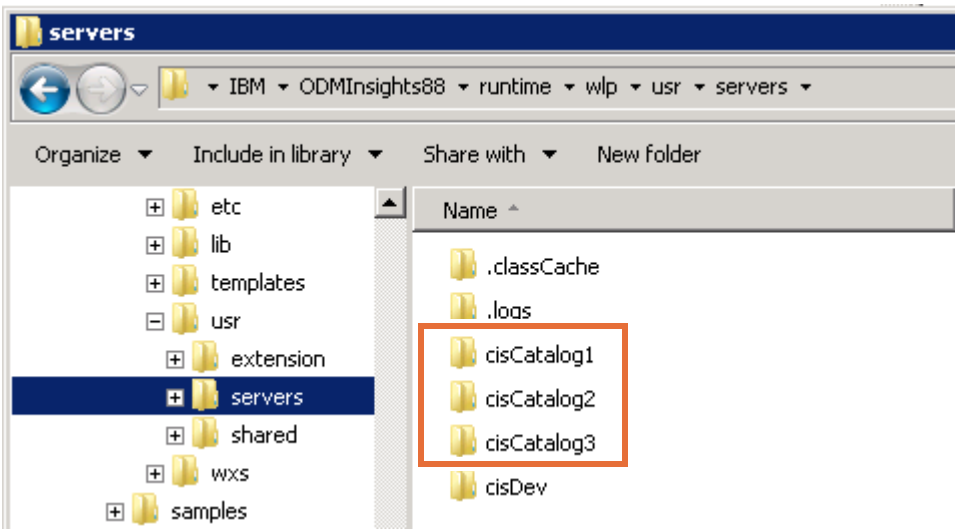
```
server create cisCatalog3 --template=cisCatalog
```

```
C:\IBM\ODMInsights88\runtime\wlp\bin>server create cisCatalog1 --template=cisCat
alog
Server cisCatalog1 created.

C:\IBM\ODMInsights88\runtime\wlp\bin>server create cisCatalog2 --template=cisCat
alog
Server cisCatalog2 created.

C:\IBM\ODMInsights88\runtime\wlp\bin>server create cisCatalog3 --template=cisCat
alog
Server cisCatalog3 created.
```

1.2. Defining the catalog cluster endpoints

__ 1. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory.

You see folders for each of the catalog servers that you created.



__ 2. Edit the cisCatalog1 bootstrap properties.

__ a. Expand the **cisCatalog1** folder, right-click the `bootstrap.properties` file and click **Edit with Notepad++**.

__ b. In the `bootstrap.properties` file, find the `ia.clusterEndpoints` property and replace it with the following definition:

```
ia.clusterEndpoints=localhost-cisCatalog1:localhost:6600:6601,localhost-cisCatalog2:localhost:6602:6603,localhost-cisCatalog3:localhost:6604:6605
```

```
25 # The server names, host names and peer connection ports of the catalog servers
26 # e.g. catalogHost01-cisCatalog:catalogHost01:6600:6601,catalogHost02-cisCatalog
27 #
28 ia.clusterEndpoints=${ia.serverName}:${ia.host}:6600:6601
29 #
```

Note

You can copy and paste this value from the `dsi.txt` file in the **C:\labfiles\code** folder. You use this value to set the endpoints for all the catalogs of the cluster so that the catalogs are aware of each other.

__ c. Replace the port values in the file to match these values.

- `http.port=9081`
- `https.port=9444`
- `ia.listenerPort=2810`

__ d. Save the file and close it.

__ 3. Edit the cisCatalog2 bootstrap properties.

__ a. Expand the **cisCatalog2** folder, right-click the `bootstrap.properties` file and click **Edit with Notepad++**.

__ b. In the `bootstrap.properties` file, find the `ia.clusterEndpoints` property and replace it with the following definition:

```
ia.clusterEndpoints=localhost-cisCatalog1:localhost:6600:6601,localhost-cisCatalog2:localhost:6602:6603,localhost-cisCatalog3:localhost:6604:6605
```

__ c. Replace the port values in the file to match these values.

- `http.port=9082`
- `https.port=9445`
- `ia.listenerPort=2811`

Note

Because the catalogs are on the same host for this exercise, you must modify the ports to avoid conflicts.

__ d. Save the file and close it.

__ 4. Edit the cisCatalog3 bootstrap properties.

__ a. Expand the **cisCatalog3** folder, right-click the `bootstrap.properties` file and click **Edit with Notepad++**.

__ b. In the `bootstrap.properties` file, find the `ia.clusterEndpoints` property and replace the value with the following value:

```
ia.clusterEndpoints=localhost-cisCatalog1:localhost:6600:6601,localhost-cisCatalog2:localhost:6602:6603,localhost-cisCatalog3:localhost:6604:6605
```

__ c. Replace the port values in the file to match these values.

- `http.port=9083`
- `https.port=9446`
- `ia.listenerPort=2812`

__ d. Save the file and close it.

1.3. Configuring security and roles

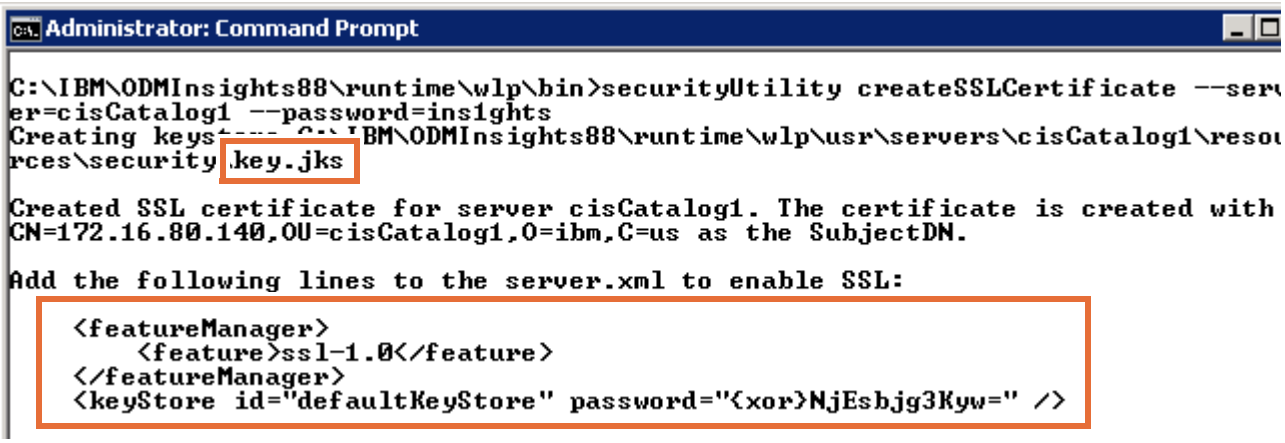
__ 1. In a command prompt window, make sure that you are in the `runtime\wlp\bin` directory.

You can type the following command to change directories to the `runtime\wlp\bin` directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 2. Configure the security for cisCatalog1 by typing this command:

```
securityUtility createSSLCertificate --server=cisCatalog1
--password=ins1ghts
```



An SSL certificate is created for cisCatalog1 in the **cisCatalog1\resources\security** folder. The response in the command prompt window shows the encrypted password that you must add to the `server.xml` file.

__ 3. Leave the command prompt window open.

__ 4. Verify that the **resources** folder with the `key.jks` file was generated for cisCatalog1.

__ a. Open Windows Explorer and go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\server\cisCatalog1` directory.

__ b. Expand the newly created **resources\security** folder.

This folder contains the `key.jks` file. For this course, you share this file with the other servers.

__ 5. Copy the **resources** folder from the **cisCatalog1** folder to the **cisCatalog2** and **cisCatalog3** folders.

__ 6. Copy the **resources** folder from the **cisCatalog1** folder to the `C:\SHARE` directory to share with the remote hosts.

Stop

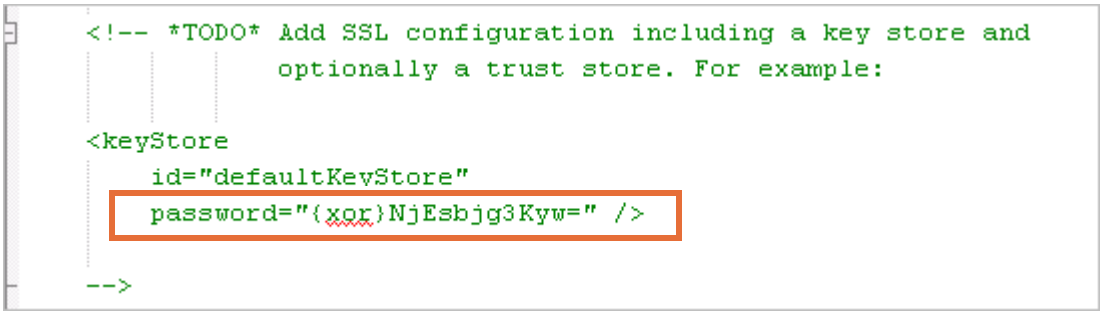
You must copy the **resources** folder from the **cisCatalog1** folder to the `C:\SHARE` directory and to the other catalog folders ***before*** you start the catalog.

__ 7. Add the security and roles to the server files for the catalogs.

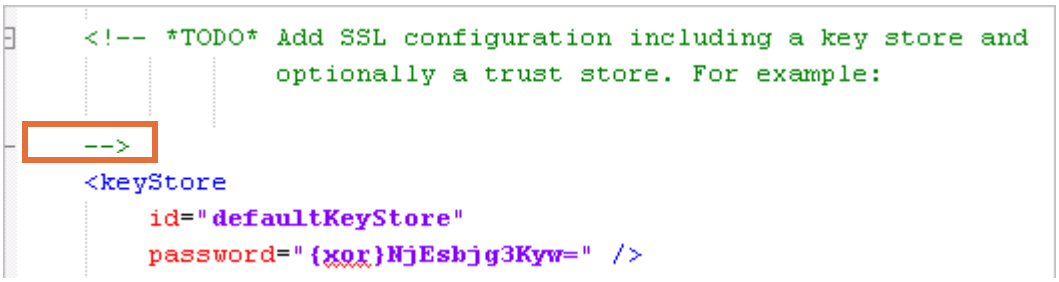
__ a. In the **cisCatalog1** folder, open the `server.xml` file (with Notepad++) and look for the “TODO” sections.

__ b. In the keystore section, copy the encrypted password that was returned from the command prompt.

```
<keyStore id="defaultKeyStore" password="{xor}NjEsbjg3Kyw=" />
```



__ c. Move the closing comment line (-->) before the keystore section so that the keystore is not commented out.



__ d. In the basicRegistry section, replace the section with the following lines:

```
<basicRegistry id="basic" realm="DWRealm">
   <user name="admin" password="ins1ghts"/>
   <group name="DWGroup">
   <member name="admin"/>
   </group>
</basicRegistry>
```

__ e. Move the basicRegistry entry outside of the commented section.

__ f. In the administrator-role section, insert the group name: DWGroup

```
<administrator-role>
  <group>DWGroup</group>
</administrator-role>
```

__ g. Move the administrator-role entry outside of the commented section.

Troubleshooting

Make sure that all your new entries to the server.xml file are outside the commented sections. Otherwise, you cannot access your servers later by using REST.

__ h. Save the file and close it.

__ 8. Copy and replace the server.xml file from **cisCatalog1** folder to the **cisCatalog2** and **cisCatalog3** folders.

All the catalog servers can share the `server.xml` file.

1.4. Enabling quorum

To enable quorum, you edit two files for each catalog server:

- `server.xml`
- `jvm.options`

__ 1. Set the `enableQuorum` property in the `server.xml` file.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory and expand the **cisCatalog1** folder.

__ b. Open the `server.xml` file (with Notepad++) and locate the `xsServer` entry.

```
<xsServer
  catalogServer="true"
  catalogClusterEndpoints="${ia.clusterEndpoints}"
  listenerPort="${ia.listenerPort}"
  transport="XIO"
  serverName="${ia.serverName}"/>
```

__ c. Append this line before the closing bracket (/>):

```
enableQuorum="true"
```

```
<xsServer
    catalogServer="true"
    catalogClusterEndpoints="${ia.clusterEndpoints}"
    listenerPort="${ia.listenerPort}"
    transport="XIO"
    serverName="${ia.serverName}" enableQuorum="true"/>
```

__ d. Save the file and close it.

__ 2. Set the `com.ibm.websphere.objectgrid.server.catalog.majority.quorum` property.

__ a. In the **cisCatalog1** folder in the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory, open the `jvm.options` file (with Notepad++).

__ b. After the first list of properties that start with “-D”, append this line:

```
-Dcom.ibm.websphere.objectgrid.server.catalog.majority.quorum=true
```

```
-Djava.endorsed.dirs=../../../wxs/lib/endorsed
-Dorg.osgi.framework.bootdelegation=com.ibm.wsspi.runtime
-Djava.library.path=.
-Dorg.apache.xml.dtm.DTMManager=org.apache.xml.dtm.ref.DTMManagerDefault
-DX10_OSGI=LATESTVERSION
-Dcom.ibm.xs.xio.transport.disableSSL=true
-Dcom.ibm.websphere.objectgrid.server.catalog.majority.quorum=true
```

__ c. Save the file and close it.

__ 3. Repeat Step 1 and Step 2 for cisCatalog2.

__ 4. Repeat Step 1 and Step 2 for cisCatalog3.

Next, you start the servers and verify that they are running and that quorum is enabled.

1.5. Starting the catalog servers

__ 1. Open three command prompt windows, and in each, go to the `wlp\runtime\bin` directory.

You can type the following command to change directories to the `wlp\runtime\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

Note

You open three command prompts because you must start all the catalogs together. The catalogs are configured to be aware of each other. If you start one while the others are not yet started, an error is produced.

__ 2. Start all the servers.

__ a. In command window 1, type the following command and press Enter:

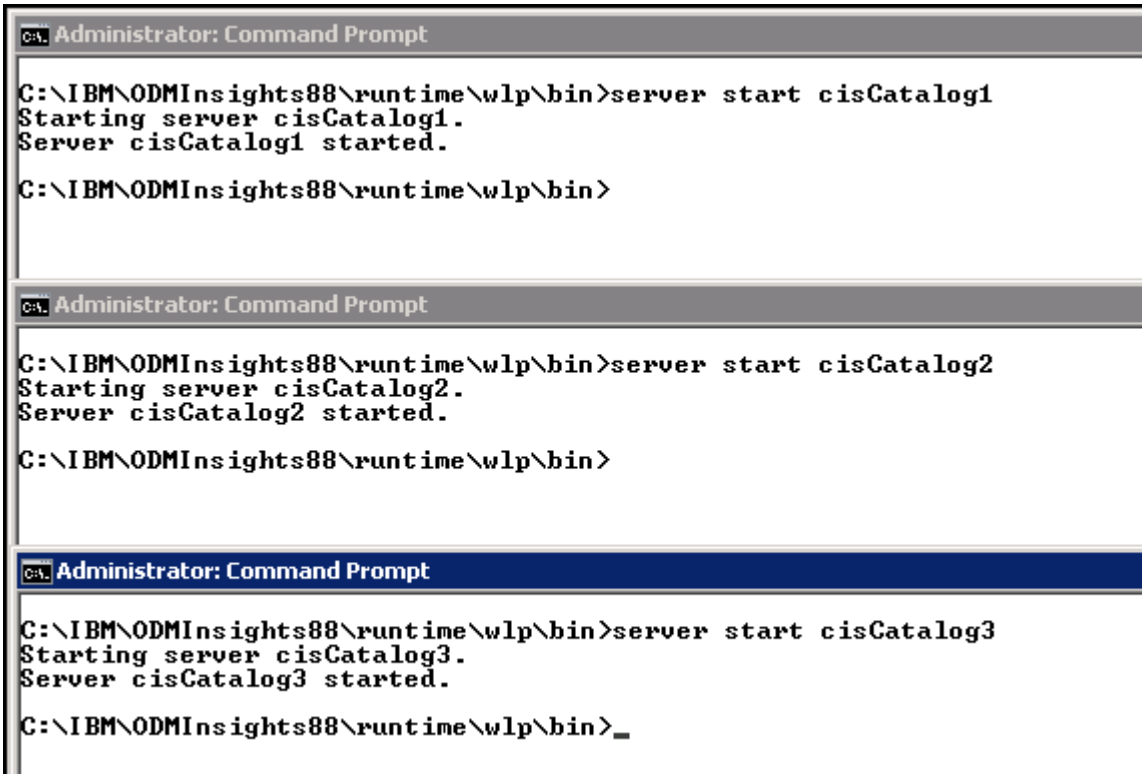
```
server start cisCatalog1
```

__ b. In command window 2, type the following command and press Enter:

```
server start cisCatalog2
```

__ c. In command window 3, type the following command and press Enter:

```
server start cisCatalog3
```



Troubleshooting

If you are unable to start a catalog server, you might need to check the logs for errors or problems. The **logs** folder for each catalog is in the catalog server folder. For example, to see the log for the cisCatalog1 server, open the `messages` file in the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisCatalog1\logs` directory.

1.6. Using REST to verify that the servers are running

__ 1. Open a browser to use the REST API to check that the catalog servers are running.

__ a. In a browser, type:

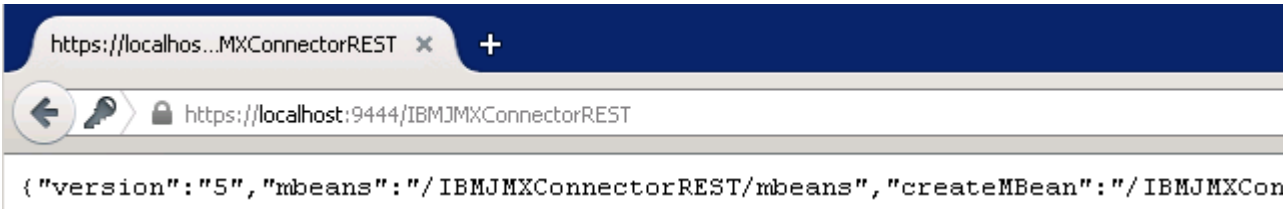
```
https://localhost:9444/IBMJMXConnectorREST
```

__ b. If you get a security warning, confirm the exception and continue.

__ 2. When prompted to sign in, use:

- **User name**: `admin`
- **Password**: `ins1ghts`

The browser returns a message that confirms the connection.



1.7. Checking the logs to verify quorum

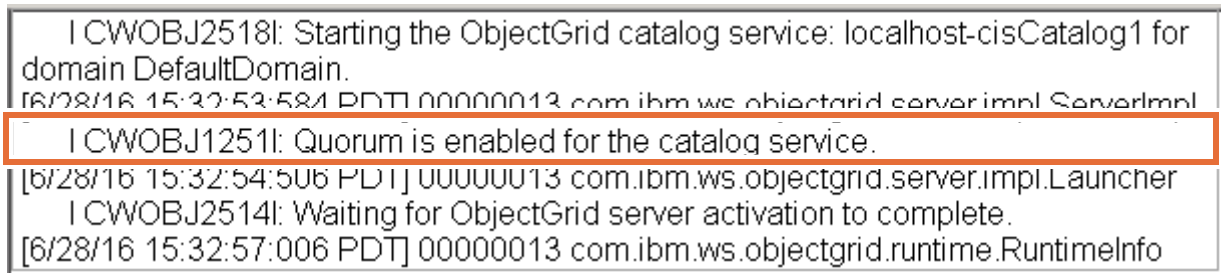
__ 1. Open the log file for the cisCatalog1 server.

__ a. In the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisCatalog1` directory, expand the **logs** folder.

__ b. Double-click the **messages** file to open the log.

__ 2. Search for “quorum” to find the following confirmation message:

```
Quorum is enabled for the catalog service.
```

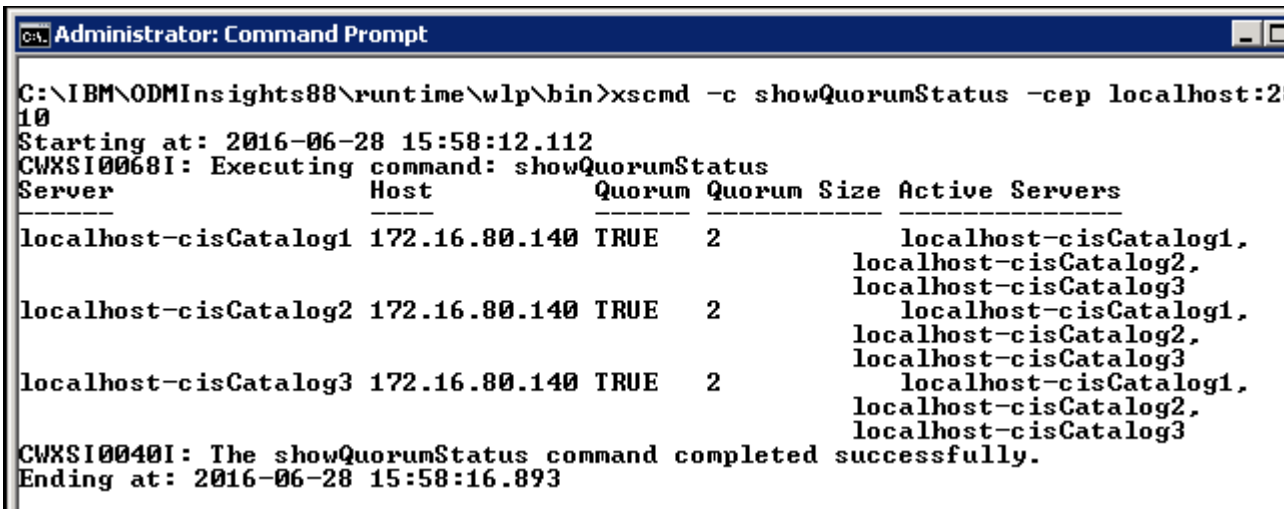


__ 3. Close the file.

1.8. Using WebSphere eXtreme Scale xscmd to check your catalog status

__ 1. Check the quorum status of the catalogs by typing this command:

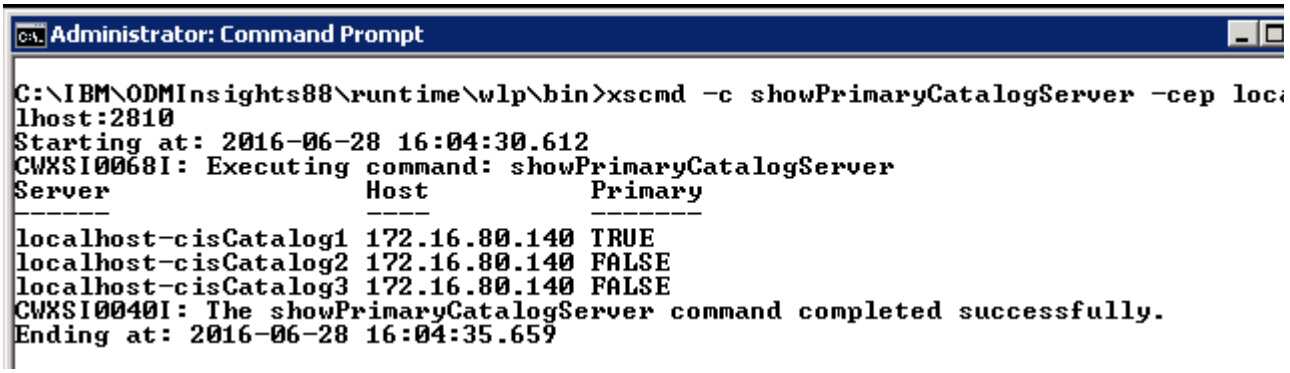
```
xscmd -c showQuorumStatus -cep localhost:2810
```



The quorum status is enabled (TRUE) for all the catalogs.

__ 2. Show the primary catalog by typing this command.

```
xscmd -c showPrimaryCatalogServer -cep localhost:2810
```



The “primary” status for cisCatalog1 server is set to TRUE to show that it is the master catalog server.

Section 2. Creating the container servers

In this section, you create and configure the container servers on the remote hosts.

2.1. Create cisContainer1

__ 1. Switch to the container1 host.

Stop

The default host name for the “container 1” host is **container1**. Your “container 1” host might have a different name.

__ 2. Open a new command prompt window and change to this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

Hint

You can copy and paste the command lines from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder on your other hosts. To do so, copy the `dsi.txt` file to the shared directory to make the file available to the remote hosts.

On the dsiHost1 host, copy the `dsi.txt` file from the ***<LabfilesDir>*\code** folder to the `C:\SHARE` directory.

__ 3. Type the following command to create the container server

```
server create cisContainer1 --template=cisContainer
```

2.2. Customize the container

__ 1. Modify the endpoints for the grid by editing the `bootstrap.properties` file for the container.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory, and expand the **cisContainer1** folder.

__ b. Open the `bootstrap.properties` file (with Notepad++) and locate the `ia.bootstrapEndpoints` property.

__ c. Set the property to the listener ports of the catalog servers in your topology.

```
ia.bootstrapEndpoints=dsiHost1:2810,dsiHost1:2811,dsiHost1:2812
```

```
#
# The host names and client listener ports of the catalog servers
# e.g. catalogHost01:2809,catalogHost02:2809,...
#
ia.bootstrapEndpoints=dsiHost1:2810,dsiHost1:2811,dsiHost1:2812
```

Stop

The default host name for the main host is **dsiHost1**.

If your host has a different name, replace `dsiHost1` in the property value with the ***actual*** host name for your main host.

__ d. Save the file and close it.

__ 2. Reduce the heap size to ensure that memory does not grow beyond the physical memory of the machine.

__ a. In the **cisContainer1** folder, and open the `jvm.options` file (with Notepad++) and locate the `-Xms` and `-Xmx` properties.

The `-Xms` property specifies the minimum Java heap and the `-Xmx` property specifies the maximum heap.

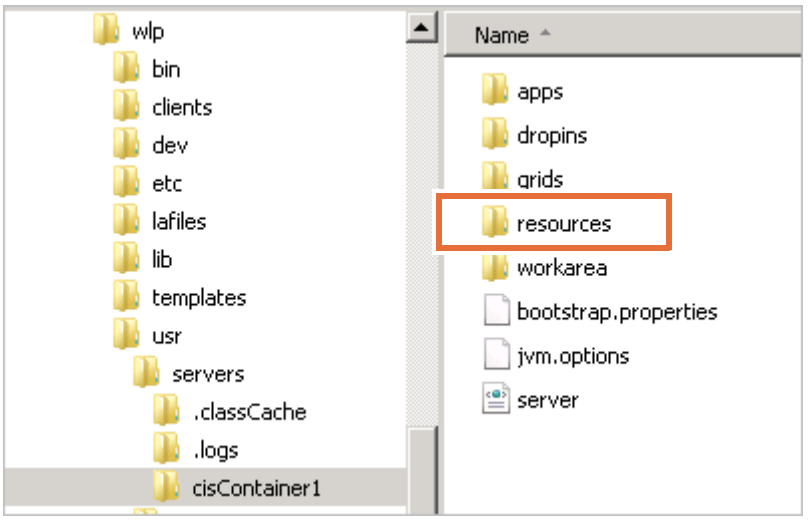
__ b. Set both properties to `3g` instead of 28g.

```
21 -Xgcpolicy:balanced
22 -Xms3g
23 -Xmx3g
24 # bypass blueprint EBA transaction interceptor
```

__ c. Save the file and close it.

__ 3. Define the security and roles for the container.

__ a. On container1, go to the mapped `SHARE(\\dsiHost1)` directory and copy the **resources** folder to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer1` directory.



__ 4. In the `server.xml` file, configure the security and roles.

__ a. Open the `server.xml` file (with Notepad++) and locate the “TODO” section.

__ b. Uncomment the keystore section and replace it with the following text:

```
<keyStore id="defaultKeyStore" password="{xor}NjEsbjg3Kyw=" />
```

Hint

If you want to copy and paste from the `dsi.txt` file that is in the ***<LabfilesDir>*\code** folder on your main host (dsiHost1), you can move that file to the `C:\SHARE` directory.

__ c. Uncomment the user registry section and replace the `basicRegistry` entries to match this text:

```
<basicRegistry id="basic" realm="DWRealm">
  <user name="admin" password="ins1ghts"/>
  <group name="DWGroup">
  <member name="admin"/>
  </group>
</basicRegistry>
```

__ d. Uncomment the **`administrator-role`** entry and replace it with the following text:

```
<administrator-role>
  <group>DWGroup</group>
</administrator-role>
```

__ e. Uncomment the authorization for the REST section and replace it with the following text:

```
<authorization-roles id="iaAuthorization">
      <security-role name="iaRESTWriter">
         <group name="DWGroup" />
      </security-role>
      <security-role name="iaRESTReader">
         <group name="DWGroup" />
      </security-role>
   </authorization-roles>
```

__ f. Verify that each of the sections that you edited are not enclosed within commented text.

```
<!-- *TODO* Add SSL configuration including a key store and
            optionally a trust store. For example:
-->
<keyStore
    id="defaultKeyStore"
    password="{xor}NjEsbjg3Kyw=" />

<!-- *TODO* Add basic or LDAP user registry configuration.
            For example:
-->
<basicRegistry id="basic" realm="DWRealm">
    <user name="admin" password="insights"/>
    <group name="DWGroup">
    <member name="admin"/>
    </group>
</basicRegistry>

<!-- *TODO* Configure authorization roles for server administration.
            For example:
-->
<administrator-role>
    <group>DWGroup</group>
</administrator-role>

<!-- *TODO* Configure authorization roles for the CIS REST API
            For example:
-->
<authorization-roles id="iaAuthorization">
    <security-role name="iaRESTWriter">
        <group name="DWGroup" />
    </security-role>
    <security-role name="iaRESTReader">
        <group name="DWGroup" />
    </security-role>
</authorization-roles>
```

__ g. Save the file and close it.

__ 5. Copy the `server.xml` file from the `cisContainer1` directory to the mapped `SHARE` drive. You can reuse this `server.xml` file for the other containers.

2.3. Start the container to verify that it is accessible

__ 1. In a command prompt window, make sure that you are in the `runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 2. Start the server by typing this command:

```
server start cisContainer1
```

Note

The server can take a few minutes to start. When the server is started, you see the message:

```
Server cisContainer1 started.
```

__ 3. Switch to the main host (dsiHost1).

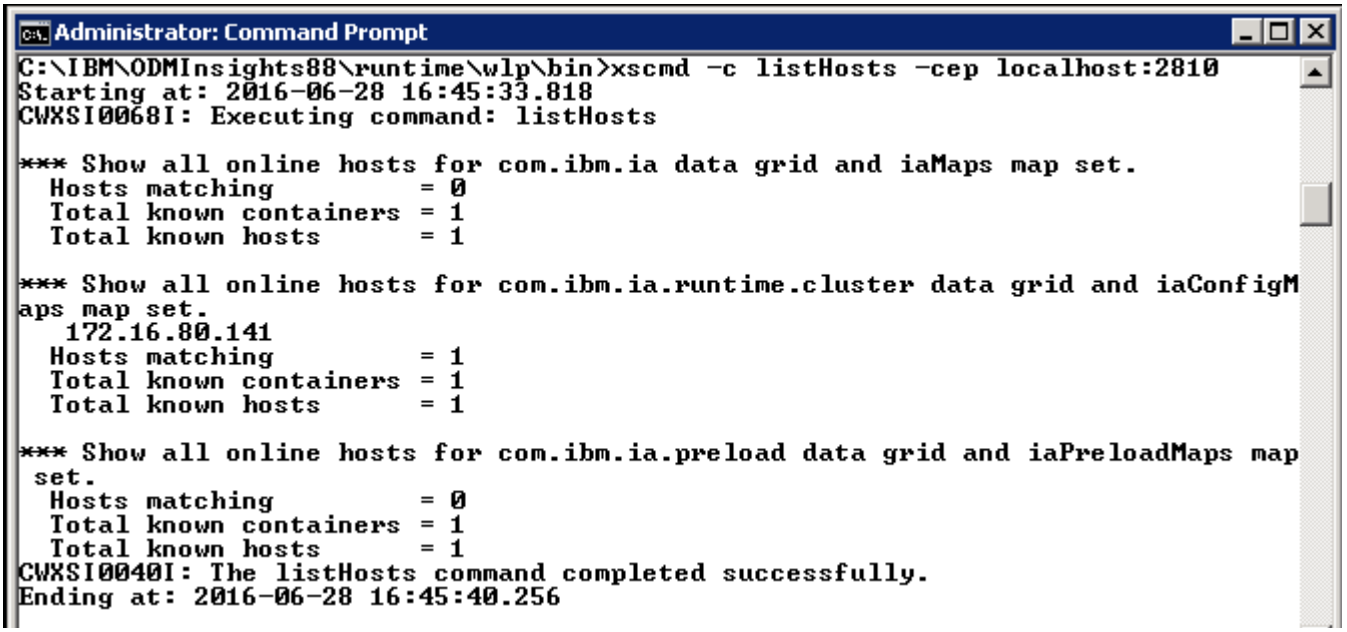
__ 4. Make sure that your main host can access container1.

__ a. In a command prompt window, make sure that you are in this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ b. Type the following command:

```
xscmd -c listHosts -cep localhost:2810
```



This command should find that one container is running, your newly created cisContainer1.

__ 5. Open a browser and use the REST API to check access to the container server.

__ a. In a browser, type:

```
https://<hostname>:9443/IBMJMXConnectorREST
```

where <hostname> is the unique name for your container1 host. For example:

```
https://container1:9443/IBMJMXConnectorREST
```

__ b. When you see the security warning, confirm the exception and continue.

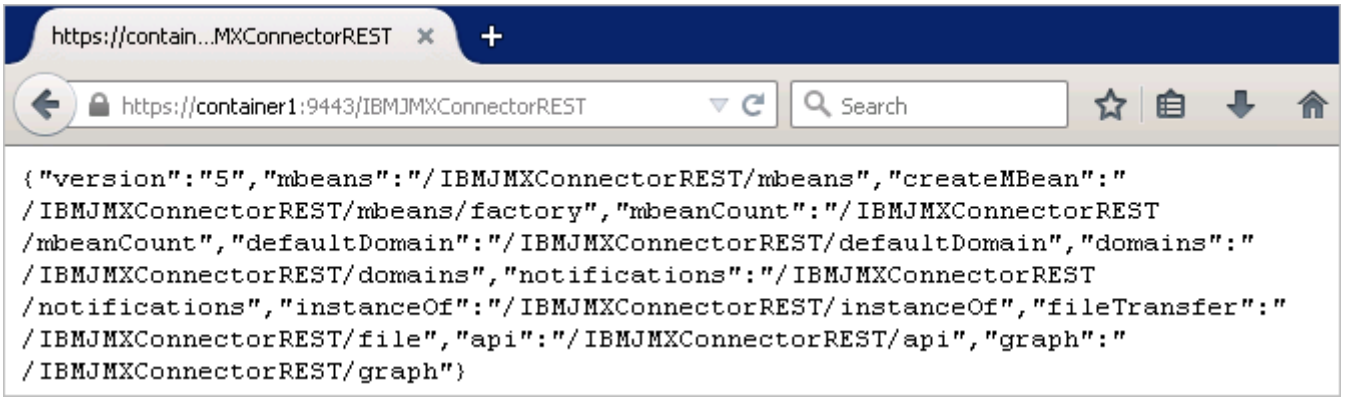
Note

The security exception is already an indication that your host is accessible. If you prefer, you can skip signing in.

__ c. When prompted to sign in, use:

- **User name**: `admin`
- **Password**: `ins1ghts`

The browser returns a message that confirms the connection.



Troubleshooting

If you are unable to access your container through REST, you must verify that you configured the container correctly. If you need to re-create the container:

__ 1. Stop cisContainer1.

```
server stop cisContainer1
```

__ 2. Delete the container.

__ a. Go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory.

__ b. Delete the **cisContainer1** folder.

__ 3. Repeat Section 2.1, "Create cisContainer1", Section 2.2, "Customize the container", and Section 2.3, "Start the container to verify that it is accessible".

2.4. Create and configure cisContainer2

__ 1. Switch to the container2 host.

Stop

The default host name for the “container 2” host is **container2**. Your “container 2” host might have a different name.

__ 2. Create the cisContainer2 server.

__ a. Open a new command prompt window and change to this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ b. Type the following command to create the container server

```
server create cisContainer2 --template=cisContainer
```

__ 3. Configure security by reusing the **resources** folder and the `server.xml` file from mapped `SHARE` drive.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory, and expand the **cisContainer2** folder.

__ b. In a separate Windows Explorer window, open the `SHARE` directory.

__ c. Copy the **resources** folder from the `SHARE` directory to the `cisContainer2` directory.

__ d. Copy the `server.xml` file from the `SHARE` directory to the `cisContainer2` directory to replace the existing file.

__ 4. Configure the bootstrap endpoints.

__ a. In the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer2` directory, open the `bootstrap.properties` file (with Notepad++).

__ b. Locate the `ia.bootstrapEndpoints` property and replace the entry with the following text:

```
ia.bootstrapEndpoints=dsiHost1:2810,dsiHost1:2811,dsiHost1:2812
```

Stop

The default host name for the main host is **dsiHost1**.

If your host has a different name, replace `dsiHost1` in the property value with the ***actual*** host name for your main host.

__ c. Save the file and close it.

__ 5. Configure the JVM heap size.

__ a. In the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer2` directory, open the `jvm.options` file (with Notepad++).

__ b. Locate the `-Xms` and `-Xmx` properties.

__ c. Set both properties to `3g` instead of 28g.

__ d. Save the file and close it.

__ 6. Verify that cisContainer2 is correctly configured by following the steps in Section 2.3, "Start the container to verify that it is accessible".

2.5. Creating and configuring cisContainer3

__ 1. Switch to the container3 host.

Stop

The default host name for the “container 3” host is **container3**. Your “container 3” host might have a different name.

__ 2. Create the cisContainer3 server.

__ a. Open a new command prompt window and change to this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ b. Type the following command to create the container server

```
server create cisContainer3 --template=cisContainer
```

__ 3. Configure security by reusing the **resources** folder and the `server.xml` file from mapped `SHARE` drive.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory, and expand the **cisContainer3** folder.

__ b. In a separate Windows Explorer window, go to the `SHARE` directory.

__ c. Copy the **resources** folder from the `SHARE` directory to the `cisContainer3` directory.

__ d. Copy the `server.xml` file from the `SHARE` directory to the `cisContainer3` directory to replace the existing file.

__ 4. Configure the bootstrap endpoints.

__ a. In the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer3` directory, open the `bootstrap.properties` file (with Notepad++).

__ b. Locate the `ia.bootstrapEndpoints` property and replace the entry with the following text:

```
ia.bootstrapEndpoints=dsiHost1:2810,dsiHost1:2811,dsiHost1:2812
```

Stop

The default host name for the main host is **dsiHost1**.

If your host has a different name, replace `dsiHost1` in the property value with the ***actual*** host name for your main host.

__ c. Save the file and close it.

__ 5. Configure the JVM heap size.

__ a. In the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer3` directory, open the `jvm.options` file (with Notepad++).

__ b. Locate the `-Xms` and `-Xmx` properties.

__ c. Set both properties to `3g` instead of 28g.

__ d. Save the file and close it.

__ 6. Verify that cisContainer3 is correctly configured by following the steps in Section 2.3, "Start the container to verify that it is accessible".

2.6. Using WebSphere eXtreme Scale xscmd to check your container status

__ 1. Switch to the main host (dsiHost1).

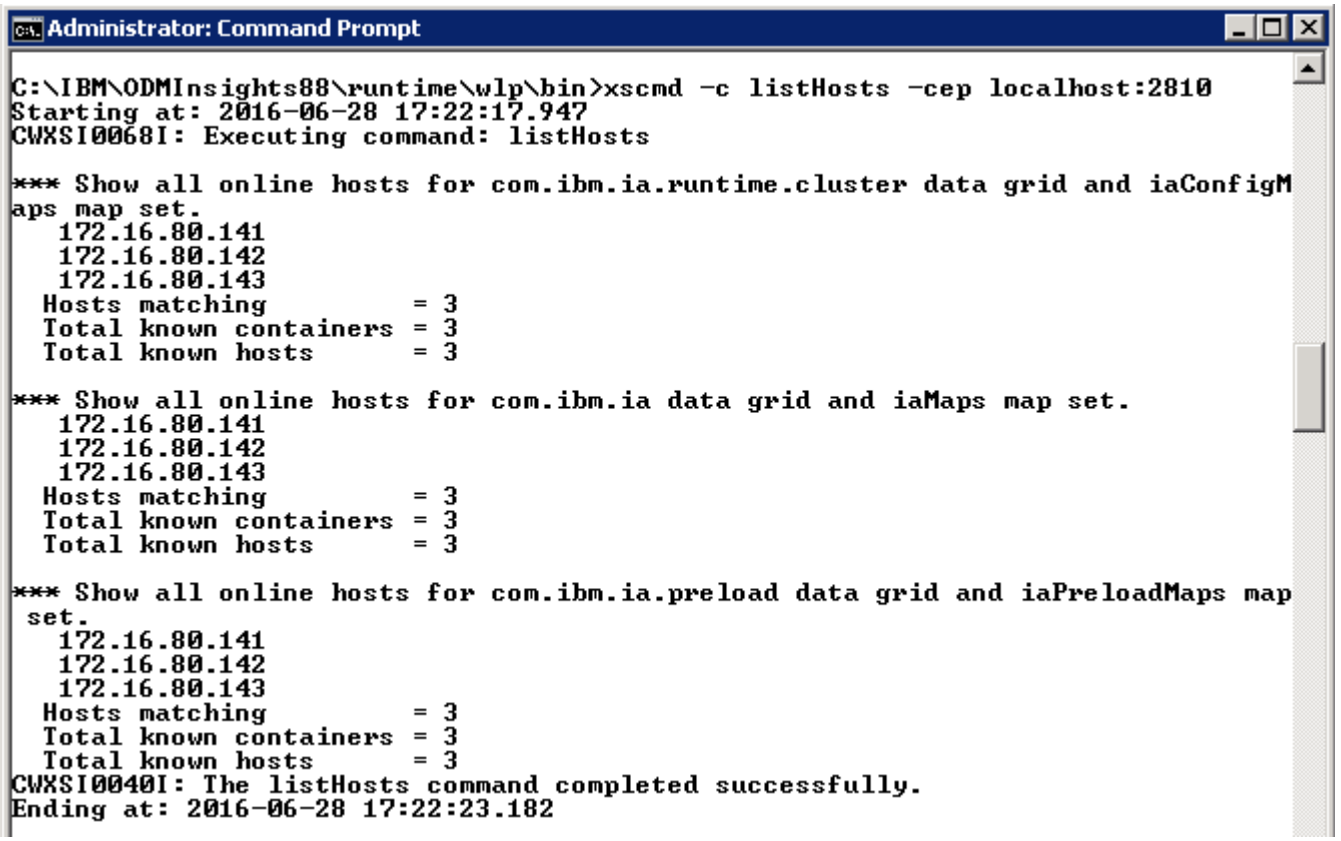
__ 2. Make sure all the containers are running and accessible to the catalogs.

__ a. In a command prompt window, make sure that you are in this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ b. Type the following command:

```
xscmd -c listHosts -cep localhost:2810
```



All the grid container servers and their IP addresses are listed.

Section 3. Creating the inbound and outbound servers

In this section, you configure the inbound and outbound servers on dsiHost1.

3.1. Creating the inbound and outbound servers

__ 1. Make sure that you are in dsiHost1.

__ 2. Open a command prompt window and change to this directory:

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 3. Type the following command to create the inbound server

```
server create cisInbound1 --template=cisInbound
```

__ 4. Type the following command to create the outbound server.

```
server create cisOutbound1 --template=cisOutbound
```

3.2. Customizing the inbound and outbound servers

__ 1. Copy the `key.jks` from the cisCatalog1 server to the cisInbound1 and cisOutbound1 servers.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers` directory.

__ b. Expand the `cisCatalog1` directory, and copy the **resources** folder.

__ c. Expand the `cisInbound1` directory and paste the **resources** folder.

__ d. Expand the `cisOutbound1` directory and paste the **resources** folder.

__ 2. Modify the ports and endpoint properties in the `bootstrap.properties` file.

__ a. In the `servers\cisInbound1` directory, open the `bootstrap.properties` file (with Notepad++) and set these properties for the **cisInbound1** server:

```
http.port=9084
https.port=9447
ia.bootstrapEndpoints=localhost:2810,localhost:2811,localhost:2812
```

__ b. Save the file and close it.

__ c. In the `servers\cisOutbound1` directory, open the `bootstrap.properties` file (with Notepad++) and set the properties for the **cisOutbound1** server.

```
http.port=9085
https.port=9448
ia.bootstrapEndpoints=localhost:2810,localhost:2811,localhost:2812
```

__ d. Save the file and close it.

Stop

Make sure that the ports are set correctly in the `bootstrap.properties` file. Port conflict causes later exercises to fail.

__ 3. Edit the security information for the inbound server to match cisCatalog1.

__ a. In the `servers\cisInbound1` directory, open the `server.xml` file (with Notepad++) and find the “TODO” section.

__ b. In the `keystore` section, replace `*INSERT_ENCODED_PASSWORD*` with the encrypted password.

```
<keyStore id="defaultKeyStore" password="{xor}NjEsbjg3Kyw=" />
```

Hint

You can copy and paste from the `dsi.txt` file.

__ c. Edit the basic registry section to use these values:

```
<basicRegistry id="basic" realm="DWRealm">
  <user name="admin" password="ins1ghts"/>
  <group name="DWGroup">
  <member name="admin"/>
  </group>
</basicRegistry>
```

__ d. Edit the role section by replacing `*INSERT_GROUP_NAME*` with this group value: `DWGroup`

```
<administrator-role>
  <group>DWGroup</group>
</administrator-role>
```

__ e. Make sure each of the modified sections are not enclosed within commented text.

```
<!-- *TODO* Add SSL configuration including a key store and
        optionally a trust store. For example:
-->
<keyStore
    id="defaultKeyStore"
    password="{xor}NjEsbjg3Kyw=" />

<!-- *TODO* Add basic or LDAP user registry configuration.
        For example:
-->
<basicRegistry id="basic" realm="DWRealm">
    <user name="admin" password="insights"/>
    <group name="DWGroup">
    <member name="admin"/>
    </group>
</basicRegistry>

<!-- *TODO* Configure authorization roles for server administration.
        For example:
-->
<administrator-role>
    <group>DWGroup</group>
</administrator-role>
```

__ f. Save the `server.xml` file and close it.

__ 4. Repeat Step 3 for the outbound server in the `servers\cisOutbound1` directory.

3.3. Starting the inbound and outbound servers

__ 1. In the command prompt window, make sure that you are in the `runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 2. Run the start command for the outbound server.

```
server start cisOutbound1
```

__ 3. Run the start command for the inbound server.

```
server start cisInbound1
```

__ 4. Use REST to verify access to the inbound and outbound servers.

__ a. In a browser, type:

- `https://localhost:9447/IBMJMXConnectorREST` (for the inbound server)
- `https://localhost:9448/IBMJMXConnectorREST` (for the outbound server)

__ b. When you get a security warning, confirm the exception and continue.

Note

The security exception is already an indication that your host is accessible. If you prefer, you can skip signing in.

__ c. When prompted to sign in, use:

- **User name**: `admin`
- **Password**: `ins1ghts`

The browser returns a message that confirms the connection.

End of exercise

Exercise review and wrap-up

In this exercise, you configured the various Decision Server Insights server types on multiple hosts.

Exercise 15.Deploying solutions

Estimated time

01:30

Overview

In this exercise, you learn how to deploy solutions to a grid. You also learn how to deploy and test connectivity for a grid environment.

Objectives

After completing this exercise, you should be able to:

- Use solutionManager to deploy solutions
- Manage deployment and connectivity for a grid environment

Introduction

In this exercise, you deploy a solution to the grid containers. You also deploy the connectivity to the inbound and outbound servers to handle inbound and outbound events.

This exercise includes these sections:

Requirements

This exercise requires that you complete Exercise 12, "Defining connectivity" and Exercise 14, "Configuring Decision Server Insights".

For this exercise, you work on your main host and the container hosts.

Attention

The default host names are: **dsiHost1**, **container1**, **container2**, and **container3**.

Make sure that you know the host names of the virtual images that you are using and that you use the ***actual*** host name during the exercises.

Section 1. Defining connection property files

In this section, you deploy the solution that you used in Exercise 12, "Defining connectivity". You deploy from dsiHost1 to the remote containers.

To help you deploy to each server, you can create connection property files that contain all the parameters that are expected by the command-line scripts. Connection properties files are stored in the **<*InstallDir*>\runtime\ia\etc** folder.

Important

If you run into issue during this exercise that you cannot solve with the instructions that are provided here, see Appendix C, "Troubleshooting issues".

1.1. Creating connection property files

__ 1. Make sure that you are on your main host (dsiHost1).

Stop

All the steps in this section are performed on the main host. Check that you are on dsiHost1 (or the unique host name that is assigned to your main host).

__ 2. Create a connection properties file so you can connect remotely to the runtime server on container1.

__ a. In the `C:\IBM\ODMInsights88\runtime\ia\etc` directory, copy the `connections.properties` file, paste it in the same directory, and rename it to: `connectionC1.properties`

__ b. Edit the `connectionC1.properties` file (with Notepad++) to match these values.

server=cisContainer1
host=container1 (or the actual name or IP address of your container 1 host)
port=9443
username=admin
password=ins1ghts
trustStoreLocation=${wlp.user.dir}/servers/cisCatalog1/resources/security/key.jks
trustStorePassword=ins1ghts
sslProtocol=TLS
disableSSLHostnameVerification=true

Attention

The default container host names are: **container1**, **container2**, and **container3**.

Make sure that you use the ***actual*** host name *or* IP address to define the **host** value in the connection properties files.

__ 3. Test that your connection properties files are correct by using the `serverManager isonline` command to test access to your runtime server.

__ a. In a command prompt, switch to the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ b. Type the following command.

```
serverManager isonline --propertiesFile=../etc/connectionC1.properties
```

Troubleshooting

You know that your connection properties file is correct if you can successfully access your container1 runtime server with the `serverManager isonline` command.

If you are unable to connect:

- Confirm that your runtime server is running.
- Confirm that your connection.properties file is edited correctly.

You can use the connection properties files that are in the ***<LabfilesDir>*\code** folder. However, if you use those files, make sure that make sure that the host names match the host names in your computer lab environment.

__ 4. Create a connection properties file so you can connect remotely to the runtime server on the container2.

__ a. Copy the `connectionC1.properties` and rename it: `connectionC2.properties`

__ b. Edit the following values in the `connectionC2.properties` file (with Notepad++).

- **server**=`cisContainer2`
- **host**=`container2` (or the actual name of your container 2 host)

__ 5. Create a connection properties file so you can connect remotely to the runtime server on container3.

__ a. Copy the `connectionC1.properties` and rename it: `connectionC3.properties`

__ b. Edit the following values in the `connectionC3.properties` file (with Notepad++).

- **server**=`cisContainer3`
- **host**=`container3` (or the actual name or IP address of your container 3 host)

__ 6. Create a `connection.properties` file for the inbound server.

__ a. In the `C:\IBM\ODMInsights88\runtime\ia\etc` directory, copy the `connectionsC1.properties` file and rename it: `connectionIn1.properties`

__ b. Edit the `connectionIn1.properties` file (with Notepad++) to match these values.

```
server=cisInbound1
host=localhost
port=9447
username=admin
password=ins1ghts
trustStoreLocation=${wlp.user.dir}/servers/cisCatalog1/resources/security
/key.jks
trustStorePassword=ins1ghts
sslProtocol=TLS
disableSSLHostnameVerification=true
```

__ 7. Create a `connection.properties` file for the outbound server.

__ a. In the `C:\IBM\ODMInsights88\runtime\ia\etc` directory, copy the `connectionIn1.properties` file and rename it: `connectionOut1.properties`

__ b. Edit the server name and port properties in the `connectionOut1.properties` file (with Notepad++) to match these values.

```
server=cisOutbound1
port=9448
```

Section 2. Deploying the solution to the grid

__ 1. Check that catalogs and containers are communicating.

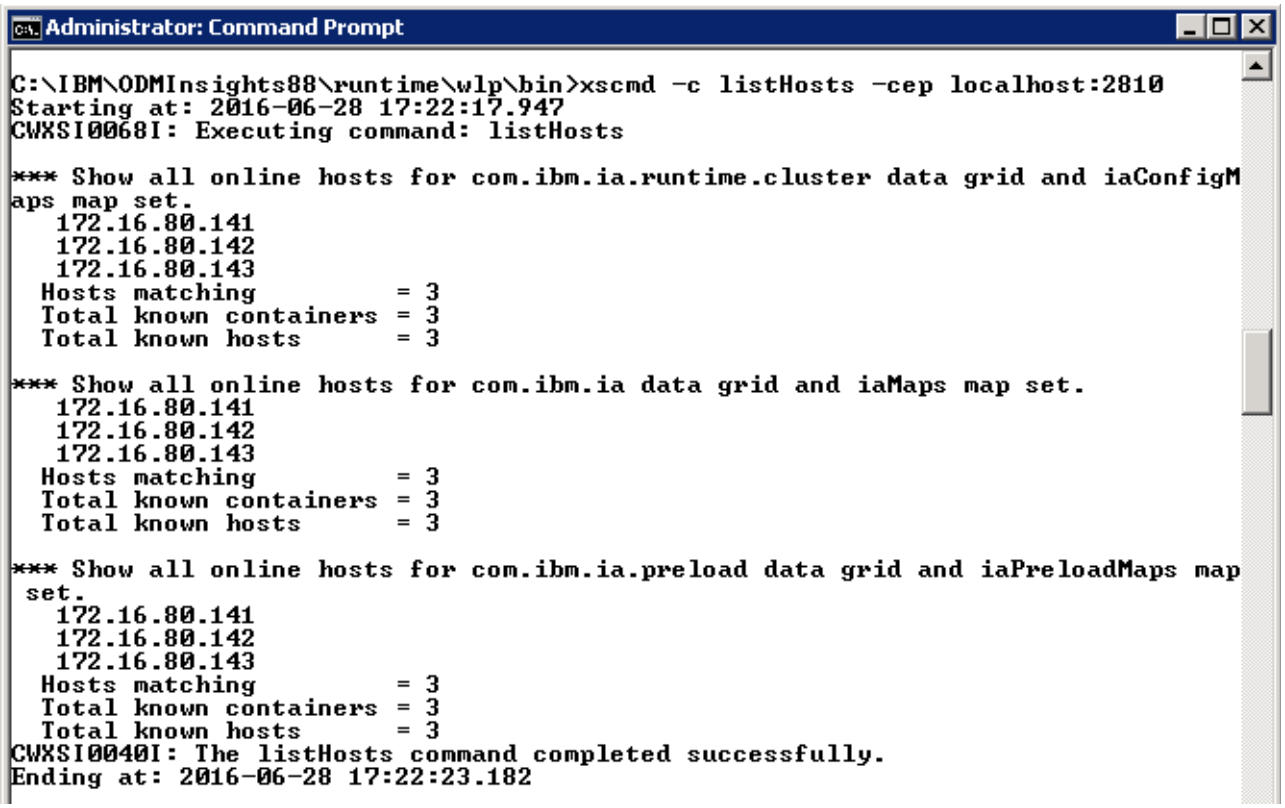
__ a. In a command prompt, switch to the `C:\IBM\ODMInsights88\runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ b. Type the following remote deployment command.

```
xscmd -c listHosts -cep localhost:2810
```

All the grid container servers and their IP addresses are listed.



Troubleshooting

If your containers are not all listed, you might need to restart the container server.

If you do not see any containers listed, you might need to check whether all your catalog servers are running.

See Appendix C, "Troubleshooting issues".

__ 2. Deploy the `banking_scenario_solution` solution to the first container.

__ a. In a command prompt, switch to the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

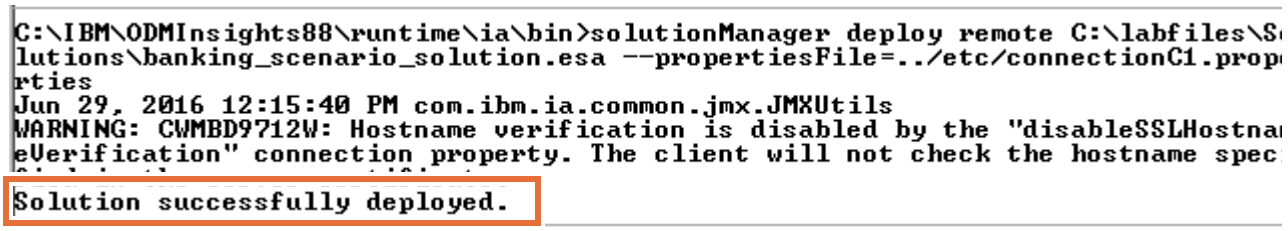
__ b. Type the following remote deployment command.

```
solutionManager deploy remote
C:\labfiles\Solutions\banking_scenario_solution.esa
--propertiesFile=../etc/connectionC1.properties
```

Hint

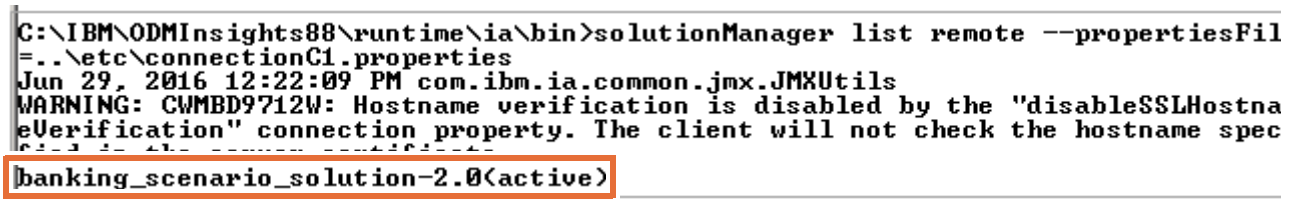
You can copy and paste the commands from the `dsi.txt` file that is in the ***<LabfilesDir>*\code** folder.

After deployment finishes, you see a message: `Solution successfully deployed.`



__ 3. Use the `solutionManager` script to verify the deployment:

```
solutionManager list remote --propertiesFile=..\etc\connectionC1.properties
```



Note

Your version of the deployed `banking_scenario_solution` might be different.

__ 4. Deploy the second container by typing this command:

```
solutionManager deploy remote
C:\labfiles\Solutions\banking_scenario_solution.esa
--propertiesFile=../etc/connectionC2.properties
```

__ 5. Deploy to the third container by typing this command:

```
solutionManager deploy remote
C:\labfiles\Solutions\banking_scenario_solution.esa
--propertiesFile=../etc/connectionC3.properties
```

__ 6. Use the `solutionManager list` command from Step 3 to verify deployment to container 2 and container 3.

__ 7. Use REST to verify the deployment to the containers.

__ a. Open a browser and type this URL to check deployment to container 1:

`http://container1:9080/ibm/ia/rest/solutions`

Attention

Make sure that you use the ***actual*** host name *or* IP address for your container in the URL.

__ b. Accept any security certificates for the browser and continue.

__ c. When prompted for authorization, use this login:

- **User name**: `admin`
- **Password**: `ins1ghts`

Note

The URL automatically switches to a secure connection (`https://container1:9443/ibm/ia/rest/solutions`).

If you do not see the solution name listed immediately, you might need to wait a few minutes and try again after the containers have time to finish load balancing.

You should see the solution and version listed.

Note

Your version of the deployed `banking_scenario_solution` might be different.

__ d. Repeat for container 2 and container 3.

Section 3. Deploying connectivity

In this section, you deploy the connectivity configurations to the inbound and outbound servers.

Hint

You can copy and paste the command lines from the `dsi.txt` file in the ***<LabfilesDir>*\code** folder.

__ 1. In a command prompt, make sure that you are in the `C:\IBM\ODMInsights88\runtime\ia\bin` directory.

__ 2. Deploy the inbound configuration by typing this command:

```
connectivityManager deploy local
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-inbound-config.xml
--propertiesFile=../etc/connectionIn1.properties
```

__ 3. Deploy the outbound configuration by typing this command:

```
connectivityManager deploy local
C:\labfiles\Solutions\banking_scenario_solution.esa
C:\labfiles\Solutions\banking-server-outbound-config.xml
--propertiesFile=../etc/connectionOut1.properties
```

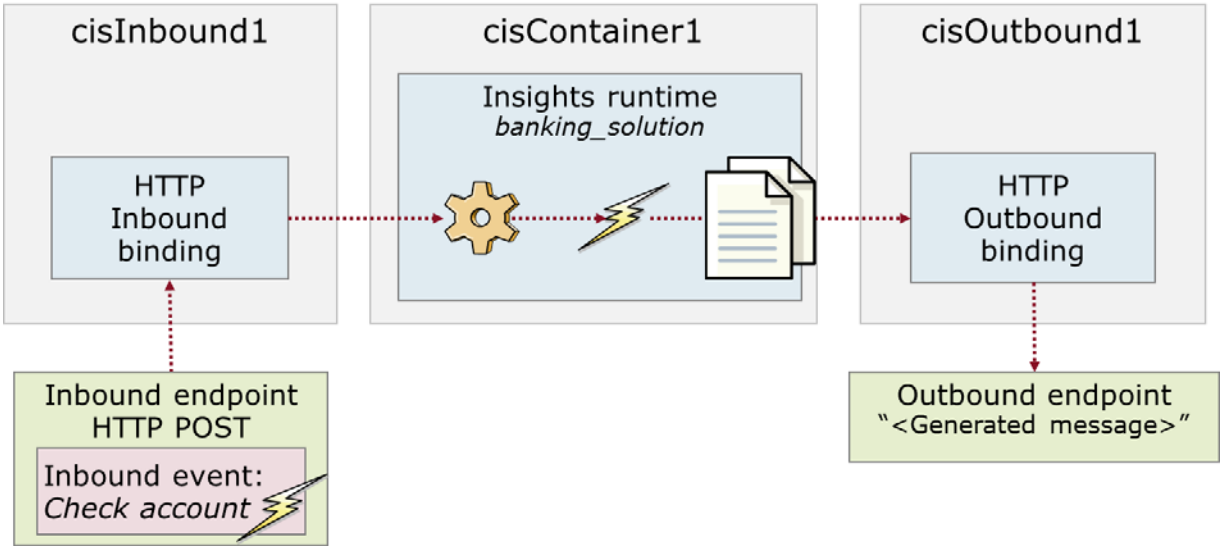
After deploying finishes, the result shows the successfully deployed and activated inbound and outbound endpoints.

```
CWMBE1452I: Successfully deployed connectivity for the solution "banking_scenari
o_solution".
CWMBE1454I: Successfully activated connectivity for the solution "banking_scenar
io_solution".
CWMBE1498I: Number of active inbound endpoints: 1
CWMBE1499I: Number of active outbound endpoints: 0
```

```
CWMBE1452I: Successfully deployed connectivity for the solution "banking_scenari
o_solution".
CWMBE1454I: Successfully activated connectivity for the solution "banking_scenar
io_solution".
CWMBE1498I: Number of active inbound endpoints: 0
CWMBE1499I: Number of active outbound endpoints: 1
```

Section 4. Testing connectivity

In this section, you test that your endpoints are correctly configured to submit and receive events and messages.



You use two additional tools:

- For the inbound event producer: HttpRequester add on to Mozilla Firefox
- For the outbound endpoint monitor: TCP/IP Monitor in Eclipse

For the event submissions, you use the text that is provided in these files of the `C:\labfiles\code\bank` directory.

- `client.txt`
- `checkAcct.txt`

4.1. Setting up TCP/IP monitoring in Eclipse

__ 1. Reopen Insight Designer and switch to a new workspace.

__ a. Go to **File > Switch Workspace > Other**.

__ b. When prompted for a workspace path in the Workspace Launcher, type:

`C:\labfiles\workspaces\connectivity`

__ 2. On the Welcome view, click **Workbench**.

__ 3. Define the TCP/IP monitoring settings from the **Window > Preferences** menu.

__ a. Click **Window > Preferences**.

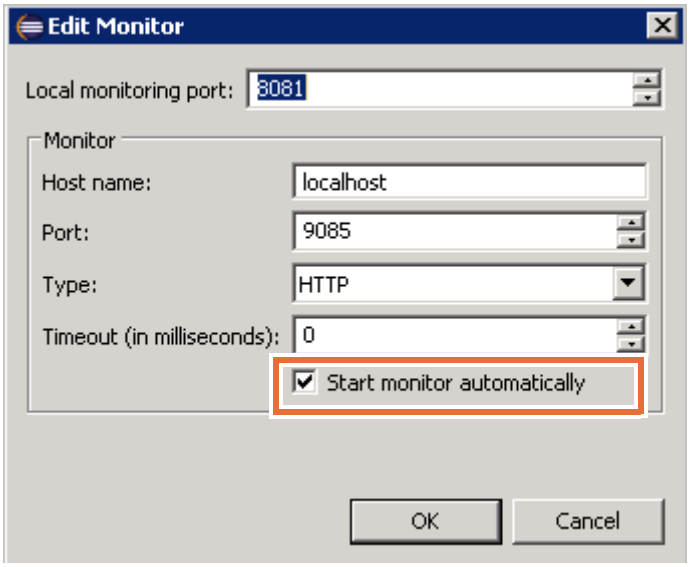
__ b. In the Preferences window, expand **Run/Debug** and click **TCP/IP Monitor**.

__ c. On the TCP/IP Monitor page, click **Add**.

__ d. Define the monitoring settings with these values:

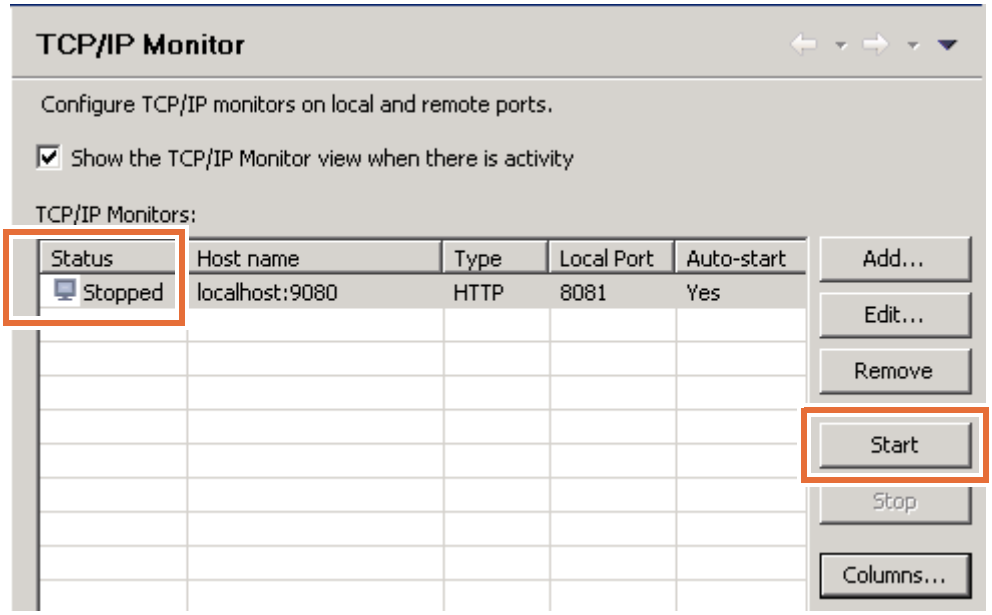
- **Local monitoring port**: `8081`
- **Host name**: `localhost`
- **Port**: `9085`
- **Type**: `HTTP`

__ e. Select **Start monitor automatically**.



__ f. Click **OK**.

__ g. If the **Status** field for the newly added monitor is “Stopped”, click **Start**.

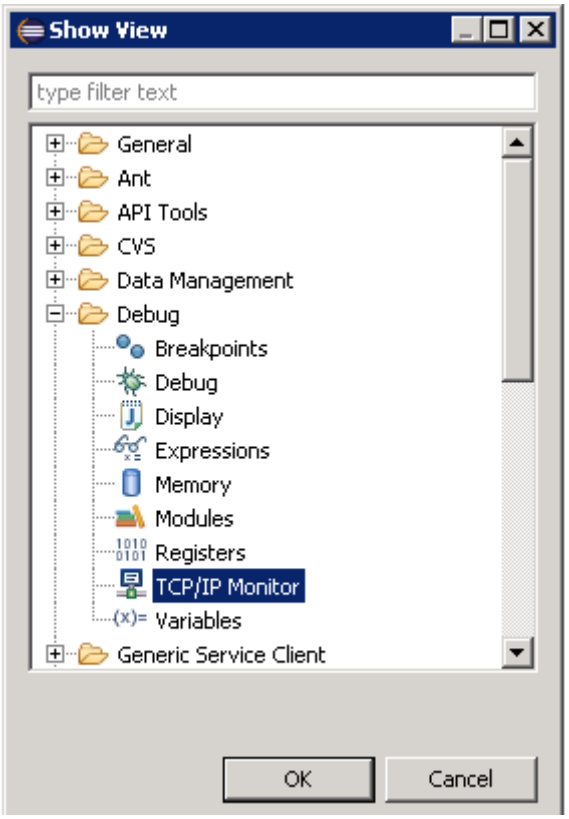


__ h. Click **OK** to close the Preferences window.

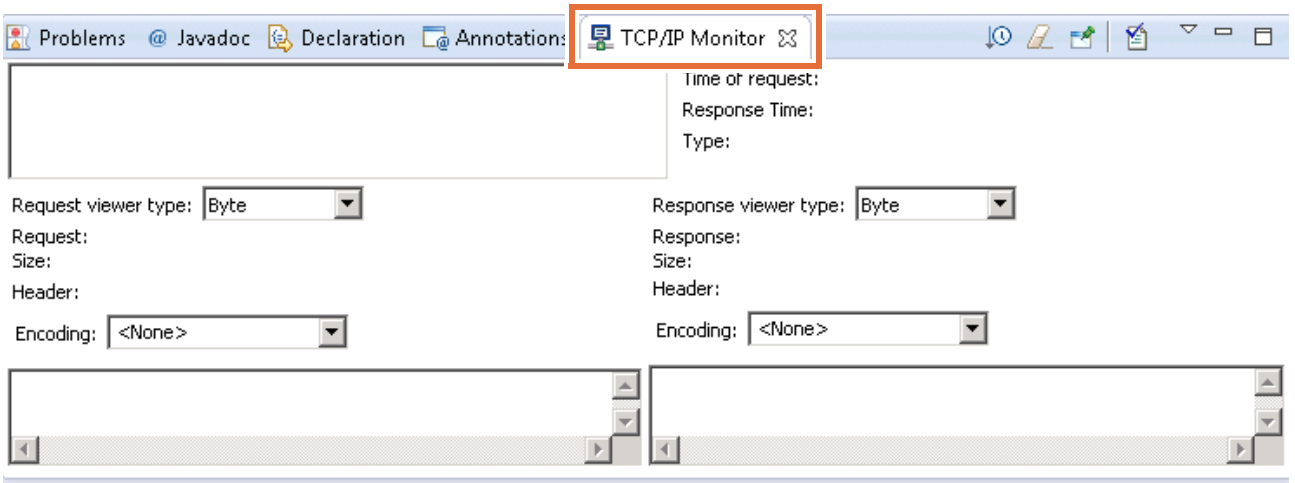
__ 4. Open the **TCP/IP Monitor** view in the Java perspective.

__ a. From the **Window** menu, click **Show View > Other**.

__ b. In the **Show View** window, select **Debug > TCP/IP Monitor** and click **OK**.



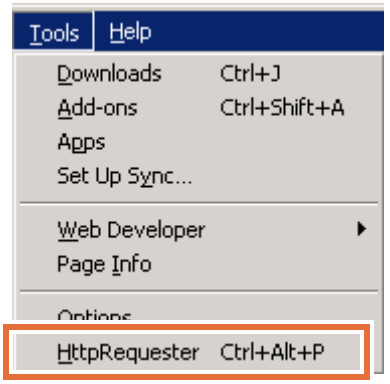
The **TCP/IP Monitor** view opens and is ready to monitor your outbound events on port 8081.



You leave Insight Designer running. Next, you use a browser to submit events.

4.2. Submitting events through the HttpRequester

__ 1. Open the Mozilla Firefox browser and on the **Tools** menu, click **HttpRequester**.



Note

You can also click the **HttpRequester** icon  on the toolbar.

__ 2. Use HttpRequester to submit a REST API POST method to create entities for the banking_scenario_solution that runs on container1.

__ a. In the **URL** field of the dialog box, type this URL:

`http://container1:9080/ibm/ia/rest/solutions/banking_scenario_solution/entity-types/banking_scenario.Client/entities`

__ b. Select **POST** from the method list.

__ c. In the **Content Type** field, select **application/xml**.

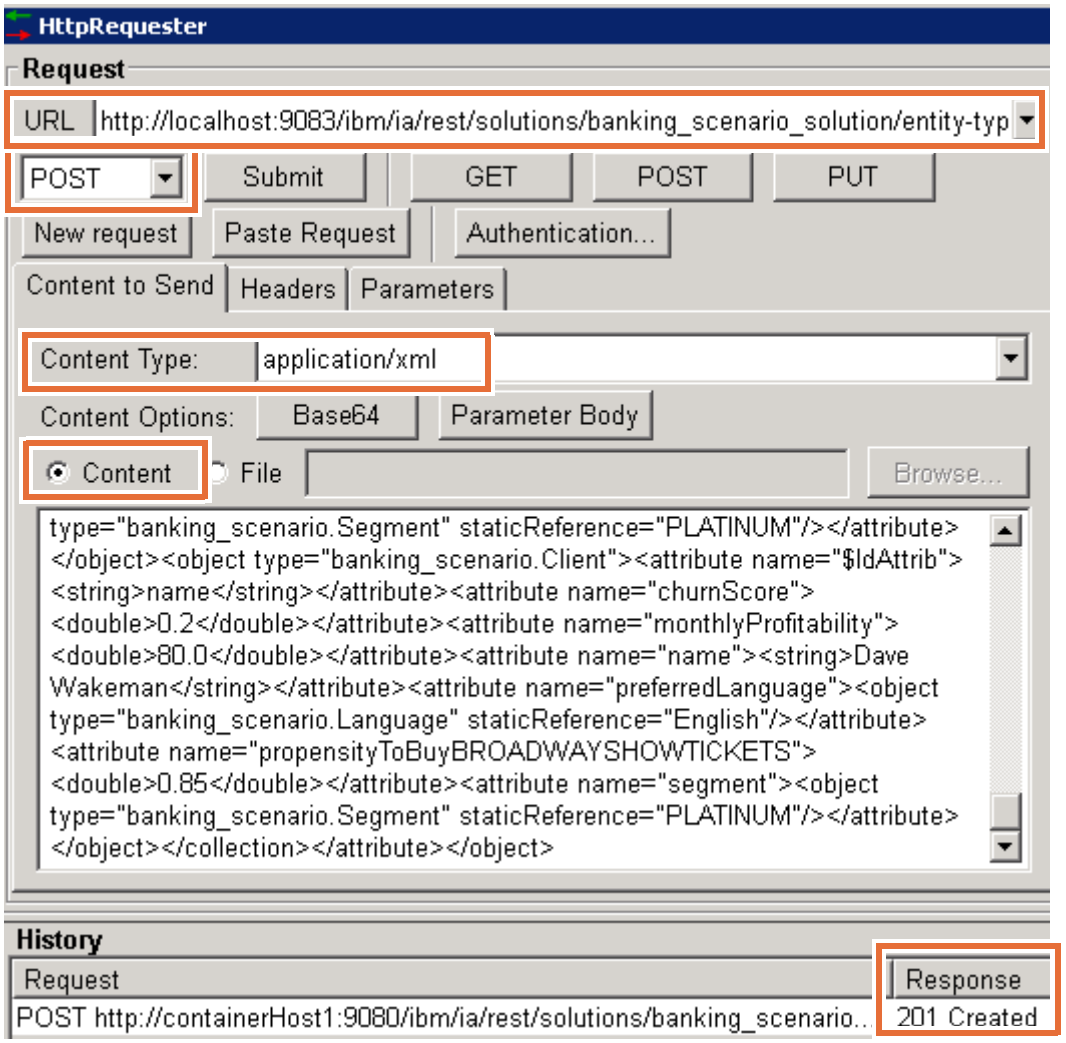
__ d. In the `C:\labfiles\code\bank` directory, double-click the `client.txt` file to open it in a text editor and copy the text (press Ctrl+A and press Ctrl+C).

__ e. Paste the text from the `client.txt` file to the **Content** field of HttpRequester (press Ctrl+V).

__ f. Click **Submit**.

__ g. When prompted for authentication, use `admin/ins1ghts` as the user name and password.

The response is returned as `201 Created`.



__ 3. Use the POST method to submit a withdrawal event to the inbound server.

__ a. In the **URL** field of the dialog box, type this URL:

`https://localhost:9447/banking/incoming`

__ b. Select **POST** in the method list.

__ c. In the **Content Type** field, select **application/xml**.

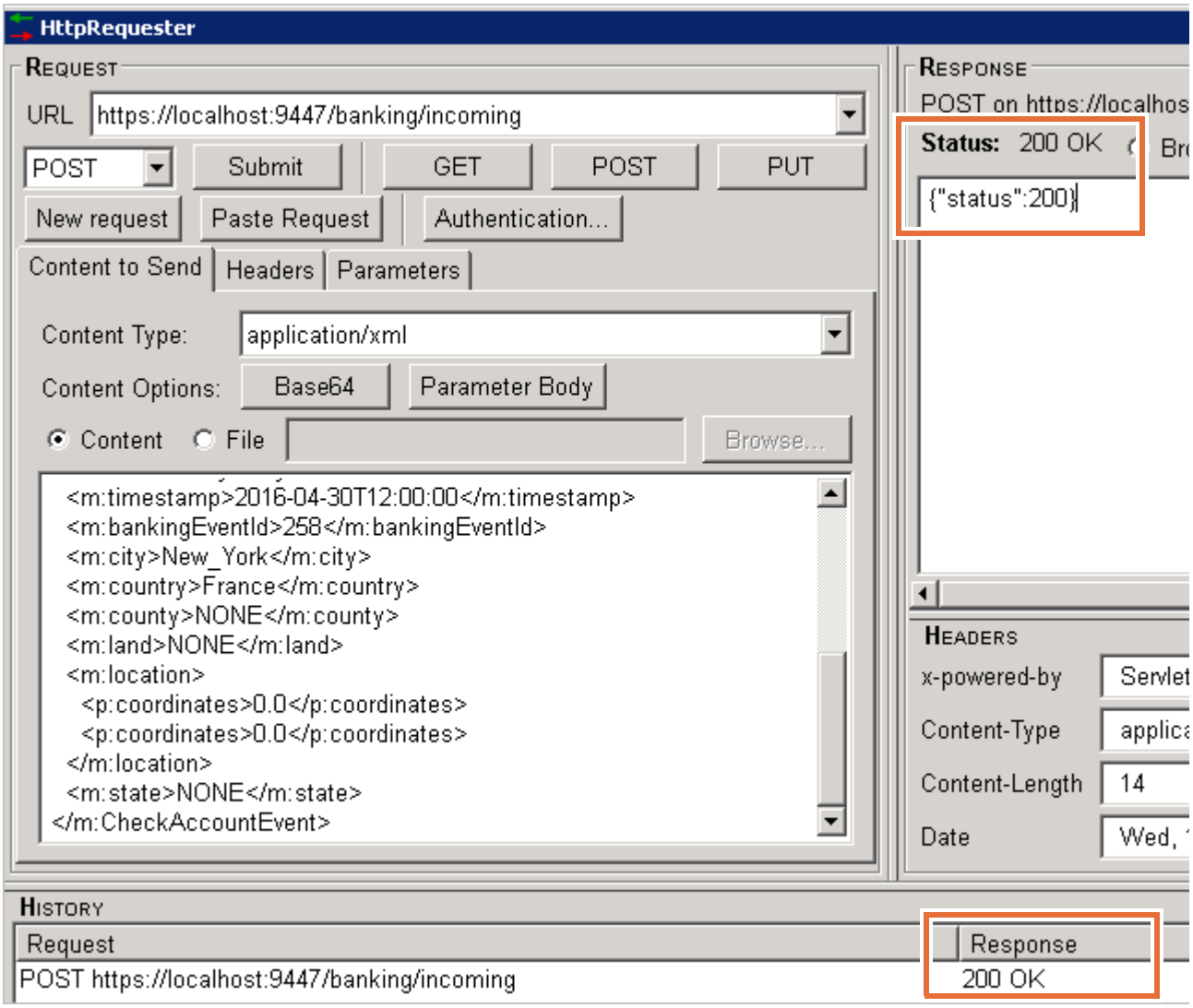
__ d. Open the `C:\labfiles\code\bank` directory, copy the text from the `checkAcct.txt` file.

__ e. Replace the test in the **Content** field with the text from the `withdraw.txt` file by pressing Ctrl+A to select all the text, and then press Ctrl+V to paste over that text.

__ f. Click **Submit**.

__ g. When prompted for authentication, use `admin/ins1ghts` as the user name and password.

The response is returned as `200 OK`.



Troubleshooting

If this test fails, make sure that all your catalogs, containers, and inbound and outbound servers are running. Also, make sure that your deployed solution is active on the container servers.

See Appendix C, "Troubleshooting issues".

Note

If you want to rerun the test, you can use the DELETE method to delete the client entities that you created with these settings:

- **URL**: `http://container1:9080/ibm/ia/rest/solutions/banking_scenario_solution/entity-types/banking_scenario.Client/entities`
- **Method**: `DELETE`
- **Content Type**: `application/xml`
- **Content**: Text that is copied from the `client.txt` file

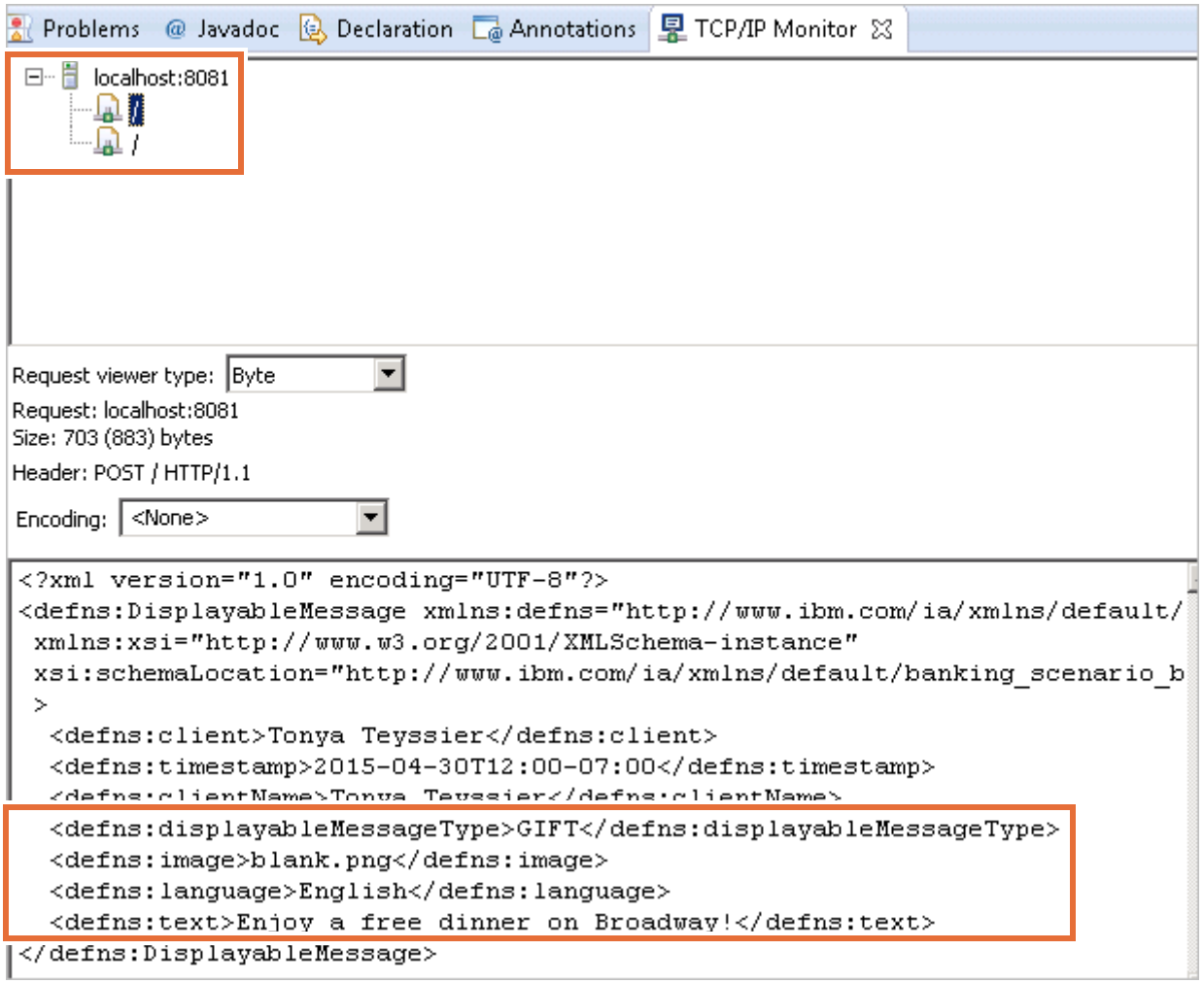
4.3. Verifying the outbound connectivity

Next, you find out whether the event submitted to the inbound server was correctly processed and generates messages that should be sent to the outbound endpoint. If the connectivity is set up correctly, the outbound server sends the messages to the outbound endpoint.

__ 1. Return to the TCP/IP Monitor window in Eclipse.

You should see two responses that are captured by the monitor.

__ 2. Select the first response and view the outbound displayable message event that is returned.



__ 3. Close Eclipse.

__ 4. Close HttpRequester.

__ 5. Close the `client.txt` and checkAcct.txt files.

End of exercise

Exercise review and wrap-up

In the first part of this exercise, you deployed a solution to the grid. You then configured and deployed connectivity to the inbound and outbound connectivity servers, and you tested your deployment.

Exercise 16.Administering Decision Server Insights

Estimated time

01:00

Overview

In this exercise, you learn how to manage security, deploy solutions, and monitor the grid.

Objectives

After completing this exercise, you should be able to:

- Monitor and manage the hosts in a Decision Server Insights grid

Introduction

This exercise includes these sections:

Requirements

This exercise requires that you complete Exercise 15, "Deploying solutions".

For this exercise, you start on your main host (dsiHost1 or the unique name that is assigned to your main host). You also work on the container hosts.

Section 1. Checking the status of your servers

In this section, you use scripts to determine server status.

__ 1. Make sure that you are on dsiHost1.

__ 2. Open a command prompt window, and change to the `C:\IBM\ODMInsights88\runtime\ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

Hint

You can copy and paste the commands for this exercise from the `dsi.txt` file that is in the *<LabfilesDir>***code** folder.

__ 3. Run the `serverManager isonline` command to check that your container servers are running.

- To check the cisContainer1 is running, type:

```
serverManager isonline --propertiesFile=../etc/connectionC1.properties
```

- To check the cisContainer2 is running, type:

```
serverManager isonline --propertiesFile=../etc/connectionC2.properties
```

- To check the cisContainer3 is running, type:

```
serverManager isonline --propertiesFile=../etc/connectionC3.properties
```

__ 4. Open another command prompt window, and change to the `C:\IBM\ODMInsights88\runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 5. Run the `server status` command to check that your container servers are running.

- To check that the inbound server is running, type:

```
server status cisInbound1
```

- To check that the outbound server is running, type:

```
server status cisOutbound1
```

Section 2. Monitoring the grid

In this section, you use the WebSphere eXtreme Scale `xscmd` utility to monitor the servers in your grid. You used some of these commands to verify your configuration during Exercise 14, "Configuring Decision Server Insights".

Troubleshooting

If you run into issues during this section, see Appendix C, "Troubleshooting issues".

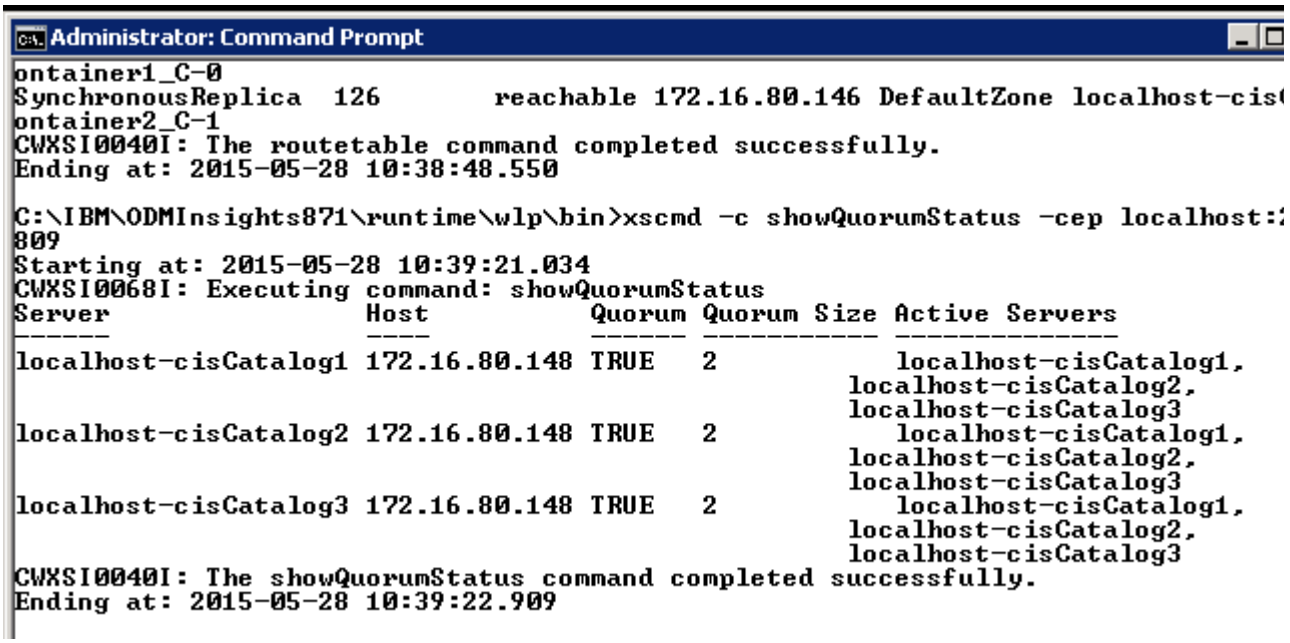
__ 1. Make sure that you are on your main host (dsiHost1).

__ 2. Open a command prompt window, and make sure that you are in the `C:\IBM\ODMInsights88\runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 3. Check the quorum status of the catalogs by typing this command:

```
xscmd -c showQuorumStatus -cep localhost:2810
```



The quorum status is enabled (`TRUE`) for all the catalogs.

__ 4. Show the primary catalog by typing this command.

```
xscmd -c showPrimaryCatalogServer -cep localhost:2810
```

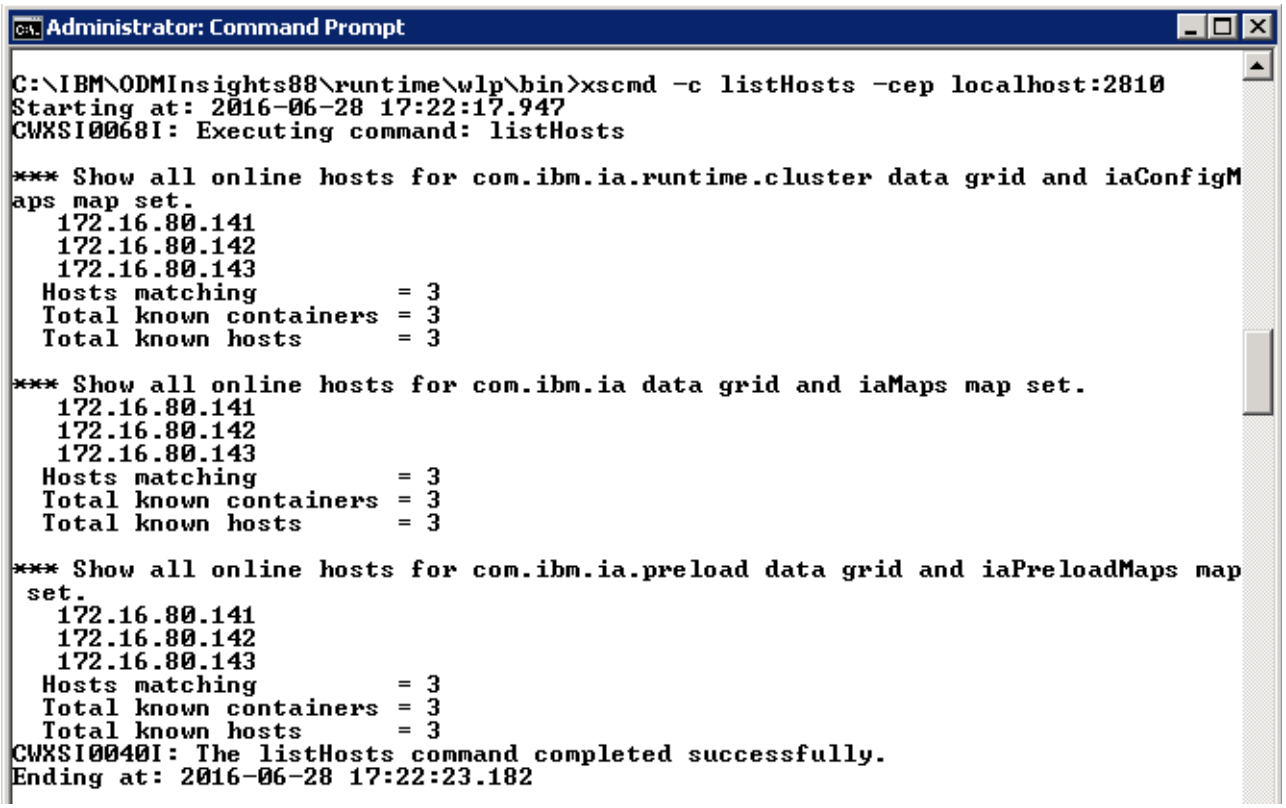
```
C:\IBM\ODMInsights871\runtime\wlp\bin>xscmd -c showPrimaryCatalogServer -cep loc
alhost:2809
Starting at: 2015-05-09 09:52:55.070
CWXSI0068I: Executing command: showPrimaryCatalogServer
Server                 Host           Primary
------                 ----           -------
localhost-cisCatalog1 172.16.80.118 TRUE
localhost-cisCatalog2 172.16.80.118 FALSE
localhost-cisCatalog3 172.16.80.118 FALSE
CWXSI0040I: The showPrimaryCatalogServer command completed successfully.
Ending at: 2015-05-09 09:52:57.054
```

The “primary” status for cisCatalog1 server is set to TRUE to show that it is the master catalog server.

__ 5. Make sure that the containers are running.

```
xscmd -c listHosts -cep localhost:2810
```

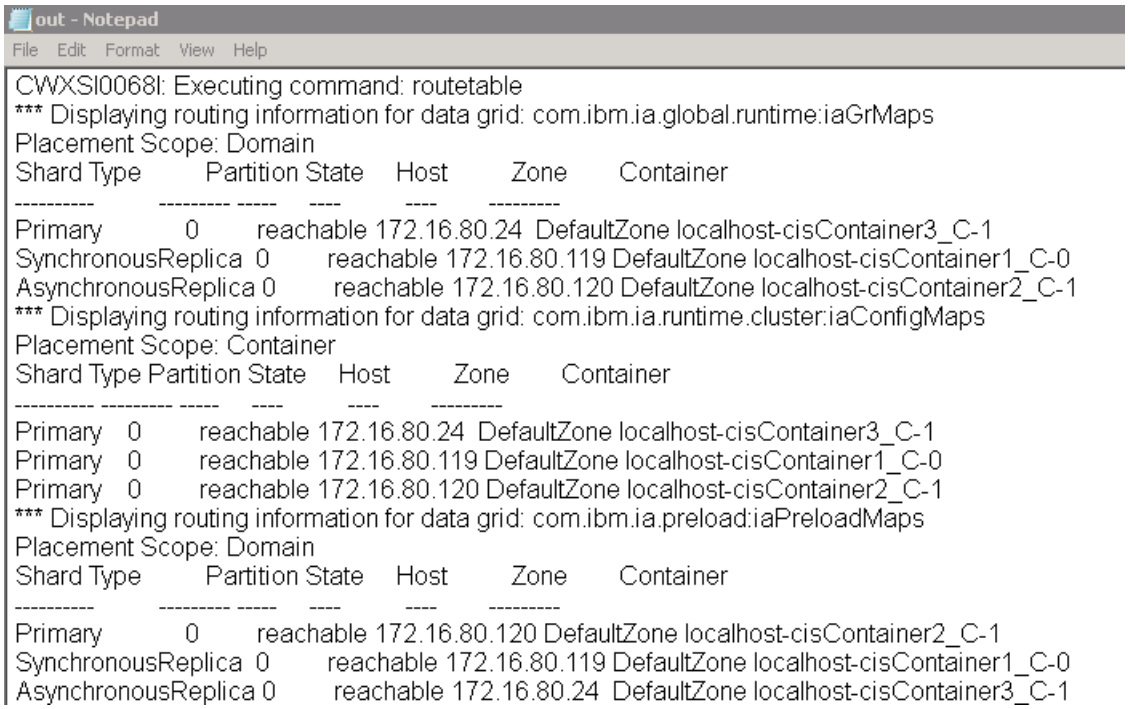
All the grid container servers and their IP addresses are listed.



__ 6. Make sure that the catalogs and containers are communicating and see all partitions on the online hosts for the data grid by typing this command:

```
xscmd -c routetable -cep localhost:2810 > out.txt
```

The result is too large to see in the command prompt window. To view the complete result, you can open the `out.txt` file in the `C:\IBM\ODMInsights88\runtime\wlp\bin` directory.



Section 3. Managing server properties

In this section, you use the `propertyManager` script to manage server properties.

__ 1. Make sure that you are on dsiHost1.

__ 2. Open a command prompt window, and make sure that you are in the `C:\IBM\ODMInsights88\runtime\ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

Hint

You can copy and paste the commands for this exercise from the `dsi.txt` file that is in the <*`LabfilesDir`*>**code** folder.

__ 3. Run the `propertyManager` to list which properties you can manage with this script.

You can use the `connectionC1.properties` file to manage the cisContainer1 server on container1.

```
propertyManager list --propertiesFile=../etc/connectionC1.properties
```

__ 4. Run the `propertyManager` script to get the current `solutionAutoStart` value for cisContainer1.

```
propertyManager get solutionAutoStart
--propertiesFile=../etc/connectionC1.properties
```

The `propertyManager` returns the value of `solutionAutoStart` as `false`.

__ 5. Set the `solutionAutoStart` property to `true` and the `LogSuppressionThreshold` property to 2.

```
propertyManager set solutionAutoStart="true" LogSuppressionThreshold="2"
--propertiesFile=../etc/connectionC1.properties
```

After you run this command, you see “Set property successful” messages with the property names, old values, and new values.

__ 6. Verify that the setting was changed in the `server.xml` file for cisContainer1.

__ a. Switch to the container1 host.

__ b. In Windows Explorer, open the `server.xml` file in the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer1` directory.

__ c. Locate the `<ia_runtime>` entry and note that the entry includes: `logSuppressionThreshold="2"`

__ d. Close the `server.xml` file.

Section 4. Creating a trace file

In this section, you review the logging properties in the `server.xml` file of the container servers. To edit these properties, you modify the `server.xml` file on each of the containers, and on the inbound and outbound servers.

4.1. Modifying the containers, inbound and outbound servers

__ 1. Make sure that you are on container1.

__ 2. Create a trace file and increase the logging values for the server.

__ a. In Windows Explorer, go to the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer1` directory.

__ b. Expand the **cisContainer1** folder, and make a backup of the `server.xml` file. (For example, you can copy and paste the file in the **cisContainer1** folder, and it is automatically renamed as "`server - Copy`".)

__ c. Open the `server.xml` file with Notepad++.

__ d. In the `server.xml` file, locate the logging entry and notice the property values:

```
<logging maxFiles="5"
traceSpecification="com.ibm.ia.*=info:com.ibm.rules.*=info:*=info"/>
```

Information

You can add or modify existing entries to the following values to get extensive trace of the solution.

For example, if you want to receive more messages, you can increase the `maxFiles` property to: `maxFiles=10`

You can also change the `traceSpecification` property to get more detailed log results for specific applications. For example, you can change `*=info` to `*=warning`.

__ e. Replace the logging entry with the following values:

```
<logging
traceSpecification="com.ibm.rules.generated.dataie.banking_scenari.*=detail:com
.ibm.ia.*=warning:
com.ibm.ia.runtime.SolutionProviderMgr=finest:com.ibm.rules.*=info:*=warning"
maxFiles="10" messageFileName="bankingSolutionMessags.log"/>
```

Hint

You can copy and paste the commands for this exercise from the `dsi.txt` file that is in the *<LabfilesDir>***code** folder.

__ f. Save the `server.xml` file and close it.

__ 3. Switch to container2 and repeat Step 2 for the cisContainer2.

__ 4. Switch to container3 and repeat Step 2 for the cisContainer3.

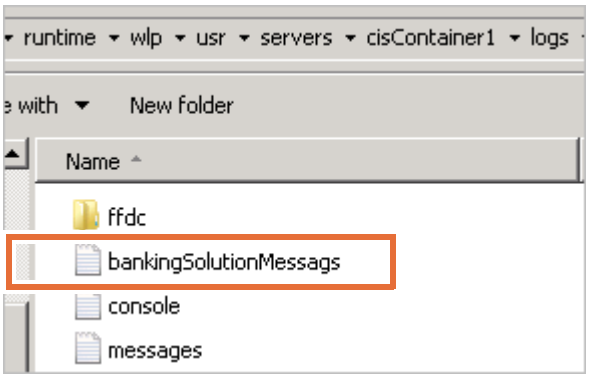
__ 5. Switch to your main host (dsiHost1) and repeat Step 2 for the cisInbound1 and cisOutbound1 servers.

Important

You do no not need to restart the servers. The changes are detected and applied automatically.

__ 6. Look for the new `bankingSolutionMessags` trace file.

__ a. Open the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisContainer1` directory, and expand the **logs** folder to see the new `bankingSolutionMessags` file.



Troubleshooting

If you do not see the `bankingSolutionMessags` file immediately, wait a few minutes and refresh the directory. Or, switch to the **logs** folder of another server. Sometimes, the file takes a few minutes to be generated.

__ b. Open the new `bankingSolutionMessags` file (in any text editor) to view the contents.

The `bankingSolutionMessags.log` file contains all messages that are written or captured by the product. This file is created only if you enable additional traces.

__ c. Close the file when you are done.

Section 5. Enable and use Insight Monitor

5.1. Enable Insight Monitor

__ 1. Make sure that you are on the main host (dsiHost1).

__ 2. Enable the Liberty admin center feature on the primary catalog server (catalog 1).

__ a. Expand the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisCatalog1` directory and open the `server.xml` file in Notepad++.

__ b. In the `<featureManager>` section, add the `adminCenter` feature tag.

`<feature>adminCenter-1.0</feature>`

```
<server description="CIS Catalog Server">

    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>restConnector-1.0</feature>
        <feature>ssl-1.0</feature>
        <feature>eXtremeScale.client-1.1.0</feature>
        <feature>eXtremeScale.server-1.1.0</feature>
        <feature>adminCenter-1.0</feature>
    </featureManager>
```

__ c. Append the `ia_admincenter` tag at the end of the `server.xml` file and include the `user` and `password` attributes for authentication.

`<ia_admincenter http.ssl.config="defaultSSLConfig" user="admin" password="ins1ghts"/>`

```
<administrator-role>
  <group>DWGroup</group>
</administrator-role>

<ia_admincenter http.ssl.config="defaultSSLConfig" user="admin"
    password="ins1ghts" />

</server>
```

__ d. Save the file.

__ 3. Enable monitoring on the other servers.

__ a. Expand the `C:\IBM\ODMInsights88\runtime\wlp\usr\servers\cisCatalog2` directory and open the `server.xml` file in Notepad++.

__ b. In the `<featureManager>` section, add the `monitor` feature tag.

`<feature>monitor-1.0</feature>`

```
<server description="CIS Catalog Server">

    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>restConnector-1.0</feature>
        <feature>ssl-1.0</feature>
        <feature>eXtremeScale.client-1.1.0</feature>
        <feature>eXtremeScale.server-1.1.0</feature>
        <feature>monitor-1.0</feature>
    </featureManager>
```

__ c. Append the `ia_admincenter` tag at the end of the `server.xml` file and include the `user` and `password` attributes for authentication.

`<ia_admincenter http.ssl.config="defaultSSLConfig" user="admin" password="ins1ghts"/>`

```
<administrator-role>
  <group>DWGroup</group>
</administrator-role>

<ia_admincenter http.ssl.config="defaultSSLConfig" user="admin"
    password="ins1ghts" />

</server>
```

__ d. Save the file.

__ 4. Repeat Step 3 on the remaining servers:

- cisCatalog3
- cisInbound1
- cisOutbound1
- cisContainer1 on the container1 host
- cisContainer2 on the container2 host
- cisContainer3 on the container3 host

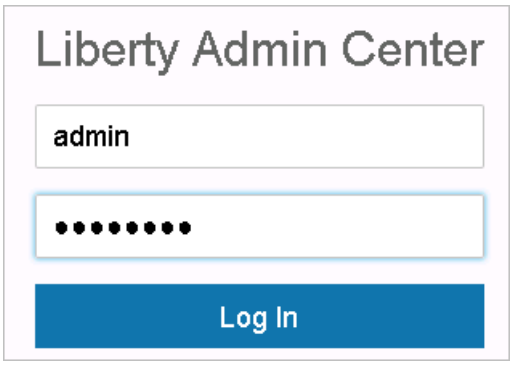
5.2. Using Insight Monitor

__ 1. Return to the main host (dsiHost1) and open a browser to this URL:

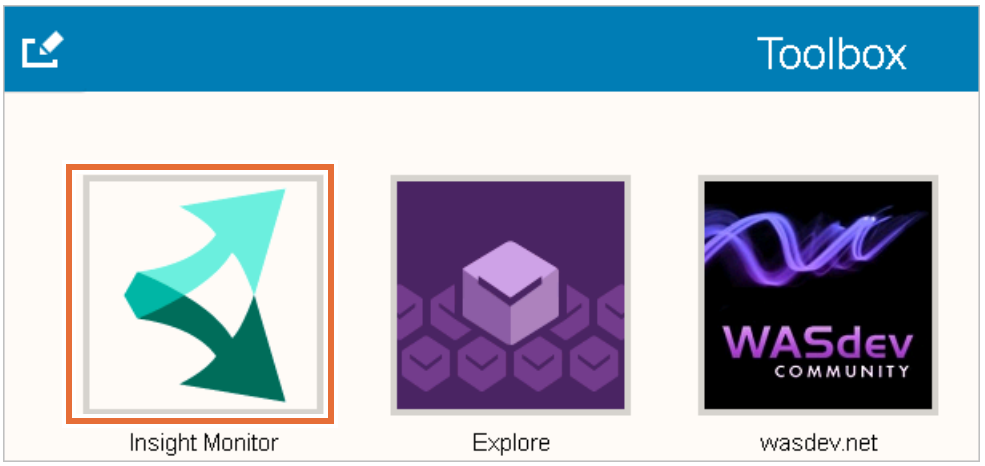
`https://localhost:9444/adminCenter`

__ 2. When prompted to log in, use the following credentials and click **Log In**:

- **User**: `admin`
- **Password**: `ins1ghts`

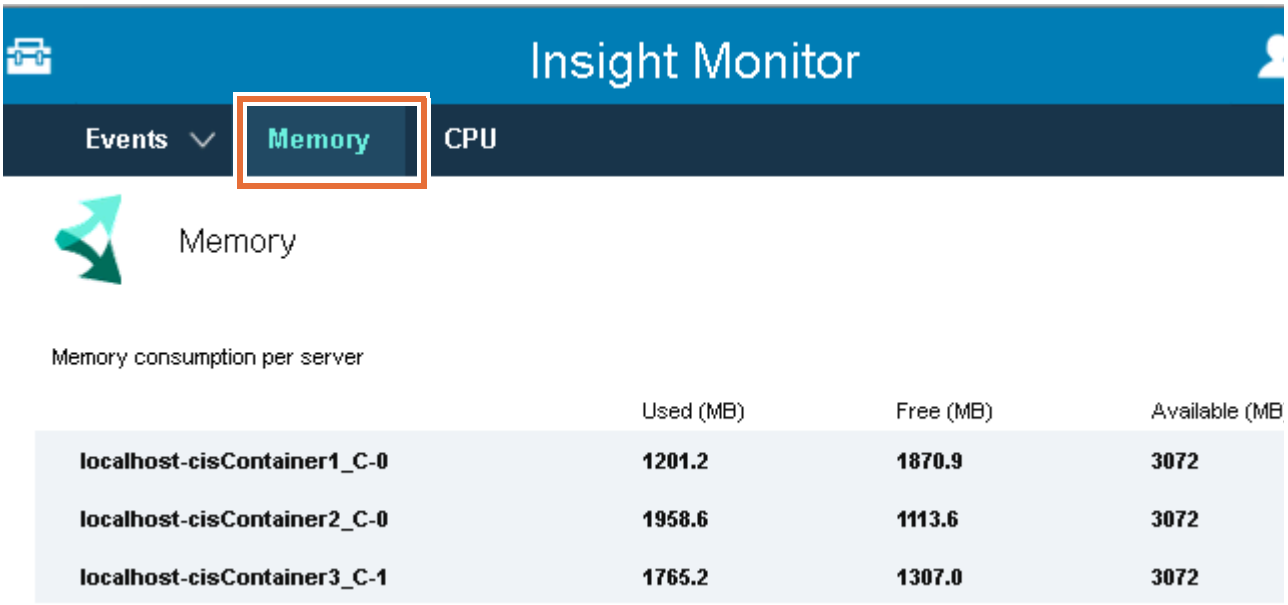


__ 3. On the Toolbox page, click the **Insight Monitor** icon.

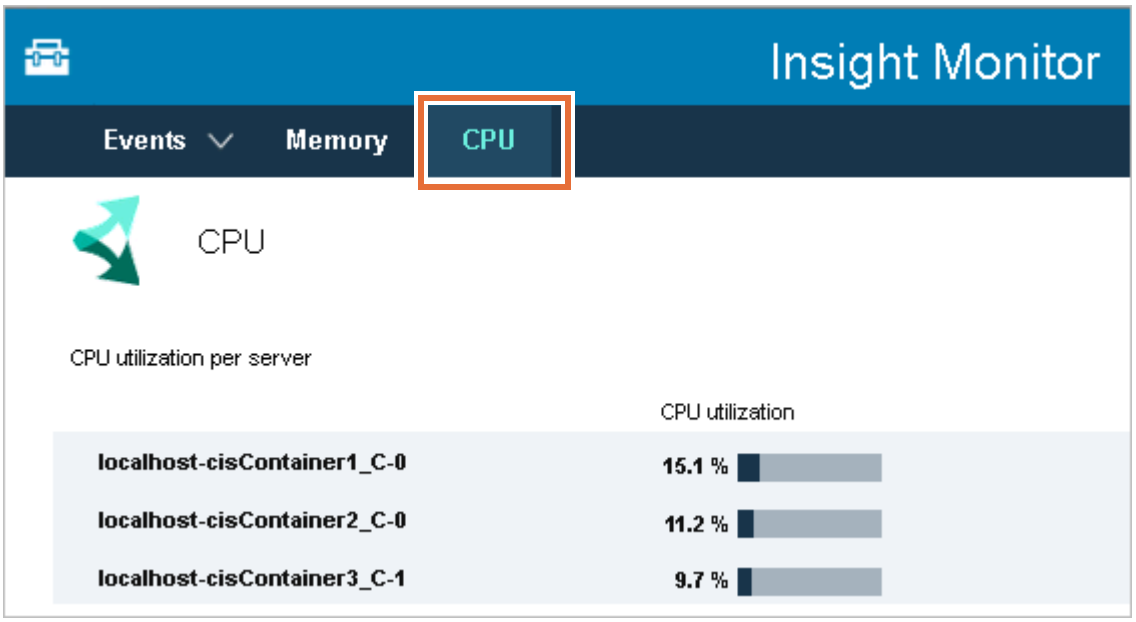


Insight Monitor opens and you see three tabs: **Events**, **Memory**, and **CPU**. On the **Events** page, you do not see any information because no events have been submitted yet since you enabled monitoring. When events are submitted to the grid, Insight Monitor Events tab is updated.

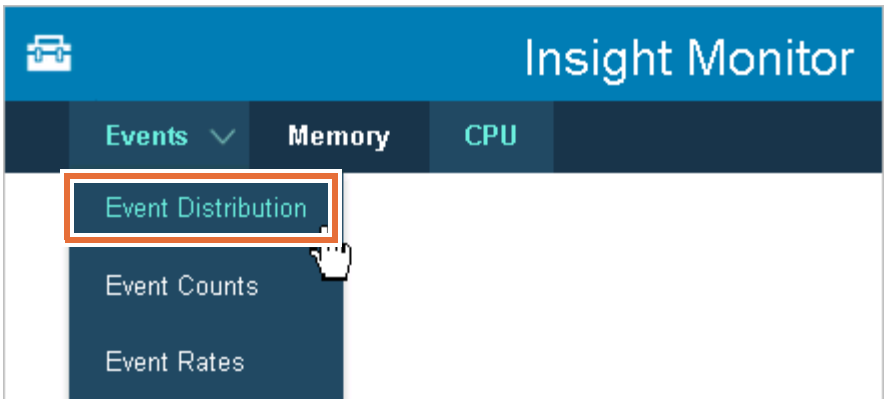
__ 4. Open the **Memory** tab and note that the memory values for each of the containers that you are monitoring.



__ 5. Open the **CPU** tab and note the utilization rates for each of the containers.



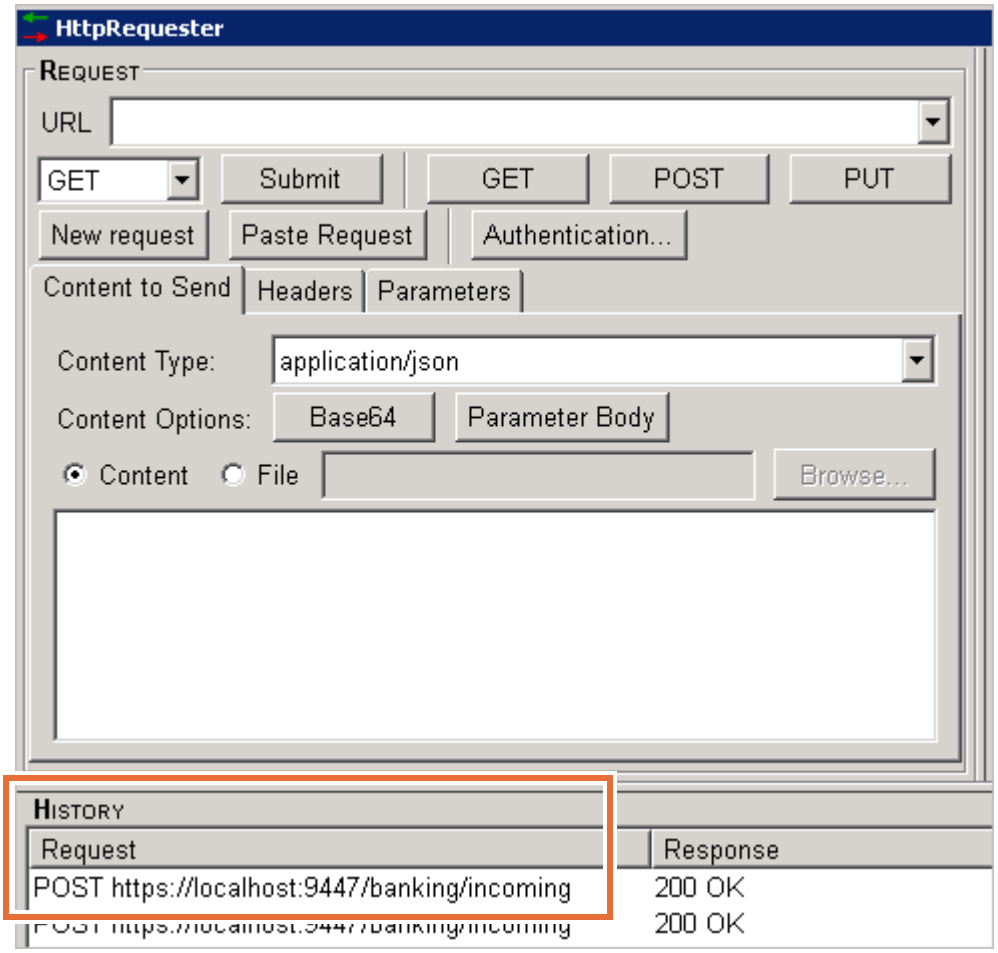
__ 6. Click the **Events** tab and click **Event Distribution** in the menu.



__ 7. Reopen HttpRequester in the Mozilla Firefox browser to resubmit events.

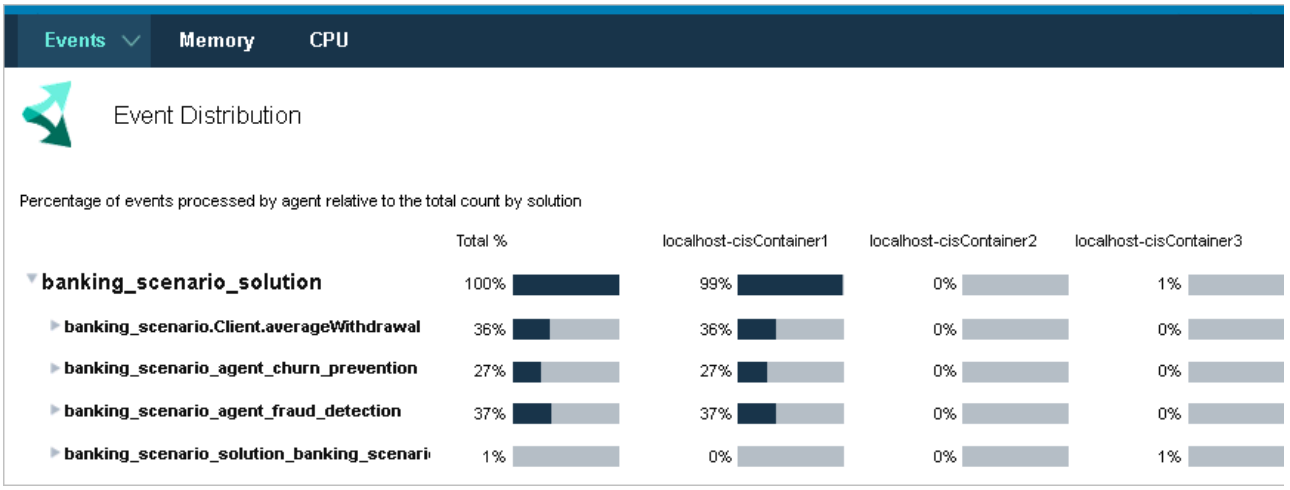
__ a. In the Firefox browser, click the **HttpRequester** icon on the toolbar.

__ b. If you still have your last POST request in the history list, you can click that request to automatically fill the request fields. Otherwise, complete the fields with the values that you used in the previous exercise. Click **Submit** repeatedly to submit the POST request



(10 or 20 times). You might need to pause slightly between clicks to make sure that you get a “200 OK” response.

__ 8. Check the **Events** page again in Insight Monitor (you might need to refresh your browser) and note how the containers are handling the event load.



__ 9. On the **Events** tab, click **Event Counts** to see the number of processed events and which events were processed.

Section 6. Undeploying solutions

This section shows you how to stop, undeploy, and delete solution files from a runtime server.

Information

This section provides useful steps that both developers and administrators should be familiar with in case of issues with deployed solutions that require the server to be cleaned and the solution to be redeployed.

6.1. Stopping and undeploying solutions

__ 1. In a command prompt, switch to the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ 2. Stop the solution on one of the containers by running the `stop` command.

```
solutionManager stop banking_scenario_solution
--propertiesFile=../etc/connectionC3.properties
```

You see a “successfully stopped” message.

```
C:\IBM\ODMInsights88\runtime\ia\bin>solutionManager stop banking_scenario_solut
on --propertiesFile=../etc/connectionC3.properties
Jun 29, 2016 1:22:19 PM com.ibm.ia.common.jmx.JMXUtils
WARNING: CWMBD9712W: Hostname verification is disabled by the "disableSSLHostna
eVerification" connection property. The client will not check the hostname spec
Solution successfully stopped: banking_scenario_solution
```

__ 3. Undeploy the solution from container 3.

```
solutionManager undeploy remote banking_scenario_solution-2.0
--propertiesFile=../etc/connectionC3.properties
```

Note

Your version of the deployed `banking_scenario_solution` might be different. If you are unsure of the version, use the `solutionManager list` command.

You see a “successfully undeployed” message.

```
C:\IBM\ODMInsights88\runtime\ia\bin>solutionManager undeploy remote banking_scen
ario_solution-2.0 --propertiesFile=../etc/connectionC1.properties
Jun 15, 2016 1:49:12 PM com.ibm.ia.common.jmx.JMXUtils
WARNING: CWMBD9712W: Hostname verification is disabled by the "disableSSLHostnam
eVerification" connection property. The client will not check the hostname speci
Solution successfully undeployed.
```

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

6.2. Deleting solution files

__ 1. Stop the runtime server on the container 3 host.

__ a. Switch to the container 3 host.

Stop

The default host name for the “container 3” host is **container3**. Your “container 3” host might have a different name.

__ b. Open a command prompt window to the `runtime\wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ c. Stop the cisContainer3 runtime server.

```
server stop cisContainer3
```

__ 2. Delete the `banking_scenario_solution` files from the container.

__ a. In a command prompt, switch to the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ b. Type the following remote deployment command.

`solutionManager delete banking_scenario_solution-3.0` (for example, to delete version 3.0)

Note

Your version of the deployed `banking_scenario_solution` might be different. Make sure that you use the correct version number.

__ 3. In the command prompt window, switch to the `wlp\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\wlp\bin
```

__ 4. Restart the runtime server with the `--clean` option to delete any remaining solution files from the `\runtime\extension\lib` and `\runtime\extension\lib\features` directories.

```
server start cisContainer3 --clean
```

__ 5. Wait until you see the message that states that the server started successfully.

__ 6. Use REST to verify the that the solution was undeployed from the container.

__ a. Open a browser and type this URL to check deployment to container 3:

```
http://container3:9080/ibm/ia/rest/solutions
```

Attention

Make sure that you use the ***actual*** host name *or* IP address for your container in the URL.

__ b. Accept any security certificates for the browser and continue.

__ c. When prompted for authorization, use this login:

- **User name**: `admin`
- **Password**: `ins1ghts`

You should not see any solution listed.

6.3. Redeploying the solution archive (.esa)

__ 1. Switch to the main host (dsiHost1).

__ 2. Deploy the `banking_scenario_solution` solution again to container 3.

__ a. In a command prompt, switch to the `ia\bin` directory.

```
cd C:\IBM\ODMInsights88\runtime\ia\bin
```

__ b. Type the following remote deployment command.

```
solutionManager deploy remote
C:\labfiles\Solutions\banking_scenario_solution.esa
--propertiesFile=../etc/connectionC3.properties
```

After deployment finishes, you see a message: `Solution successfully deployed`.

__ 3. Verify the deployment by using the `solutionManager list` command:

```
solutionManager list remote --propertiesFile=..\etc\connectionC3.properties
```

End of exercise

Exercise review and wrap-up

In this exercise, you used administration tools to manage the servers and logging properties on multiple hosts.

Appendix A. Host names and IP addresses

Write the host names and IP addresses that are assigned to the virtual machines in your environment.

Main host Dual core 16 GB RAM *Default host name: dsiHost1* Assigned host name:_____ IP: _____	**Container 1 host** Single core 8 GB RAM *Default host name: container1* Assigned host name:_____ IP: _____
Container 2 host Single core 8 GB RAM *Default host name: container2* Assigned host name:_____ IP: _____	**Container 3 host** Single core 8 GB RAM *Default host name: container3* Assigned host name:_____ IP: _____

Important

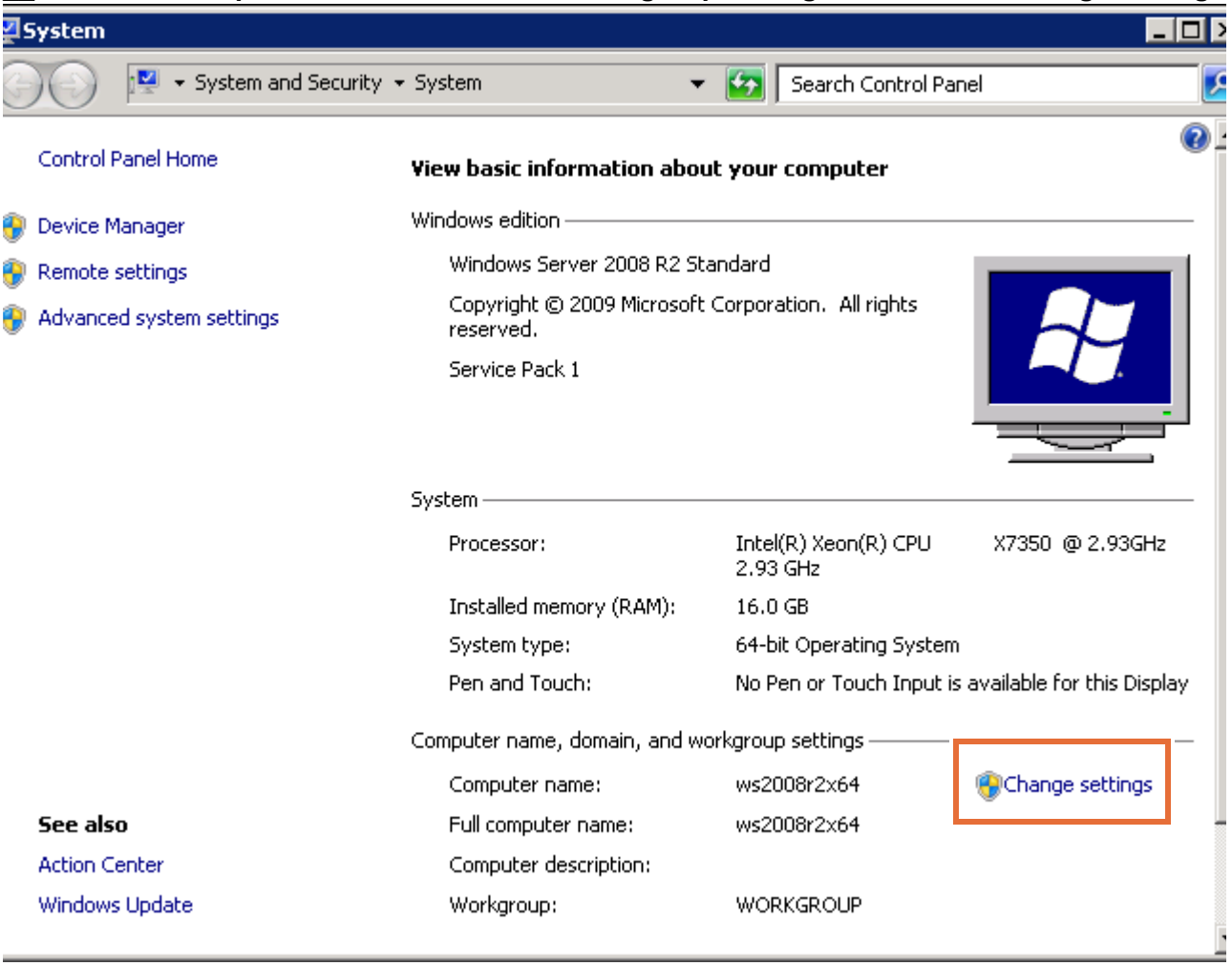
When host names are specified in the exercise steps, make sure that you use the actual host name that you noted here.

Appendix B. Changing host names and mapped drives

B.1.Renaming hosts

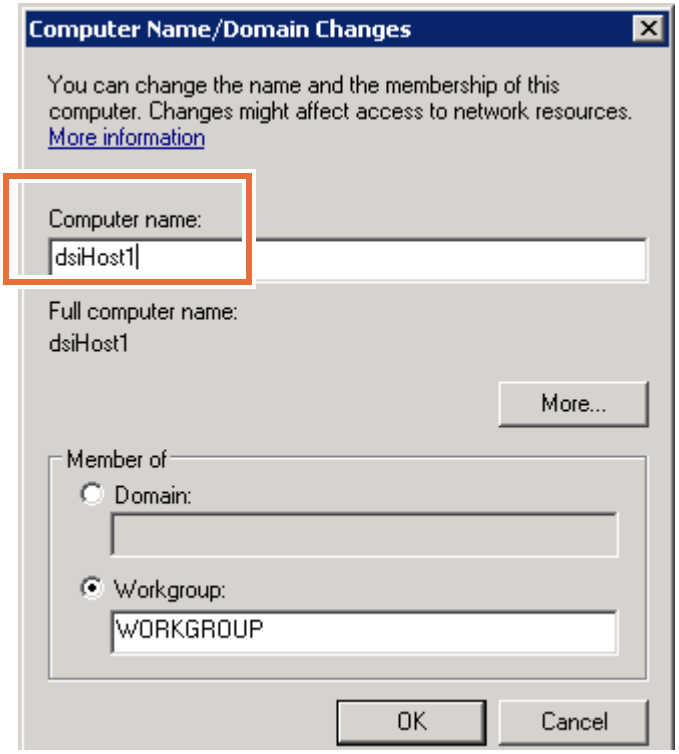
__ 1. On the desktop, right-click the **Computer** icon and click **Properties**.

__ 2. In the **Computer name, domain, and workgroup settings** section, click **Change settings**.



__ 3. In the System Properties dialog box, on the **Computer Name** page, click **Change** and click **OK**.

__ 4. In the **Computer name** field, type a unique name for your host and click **OK**.



__ 5. Accept the warnings about restarting and close the System Properties windows.

__ 6. When prompted to restart the workstation, click **Restart Now** and wait for your workstation to restart before proceeding.

Note

You use this method if you also need to rename your container hosts.

B.2.Mapped drives

If you rename your main host, you must also create a new mapped drive on each of the container hosts.

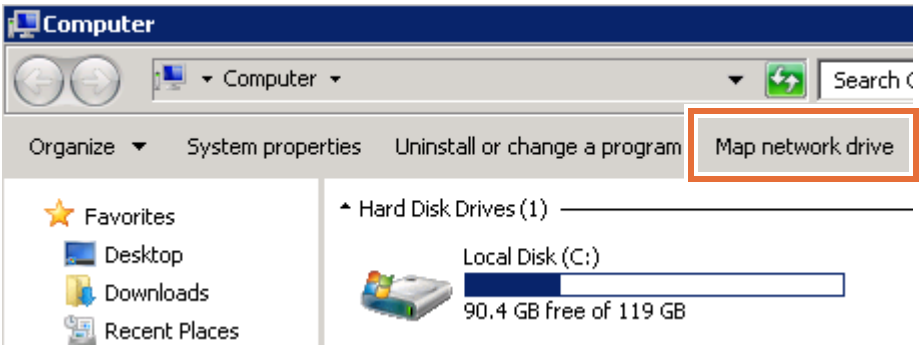
To map each container host to the shared drive:

__ 1. Go to containerHost1 (or the name of your “container 1” host).

__ 2. Map a drive to the main host.

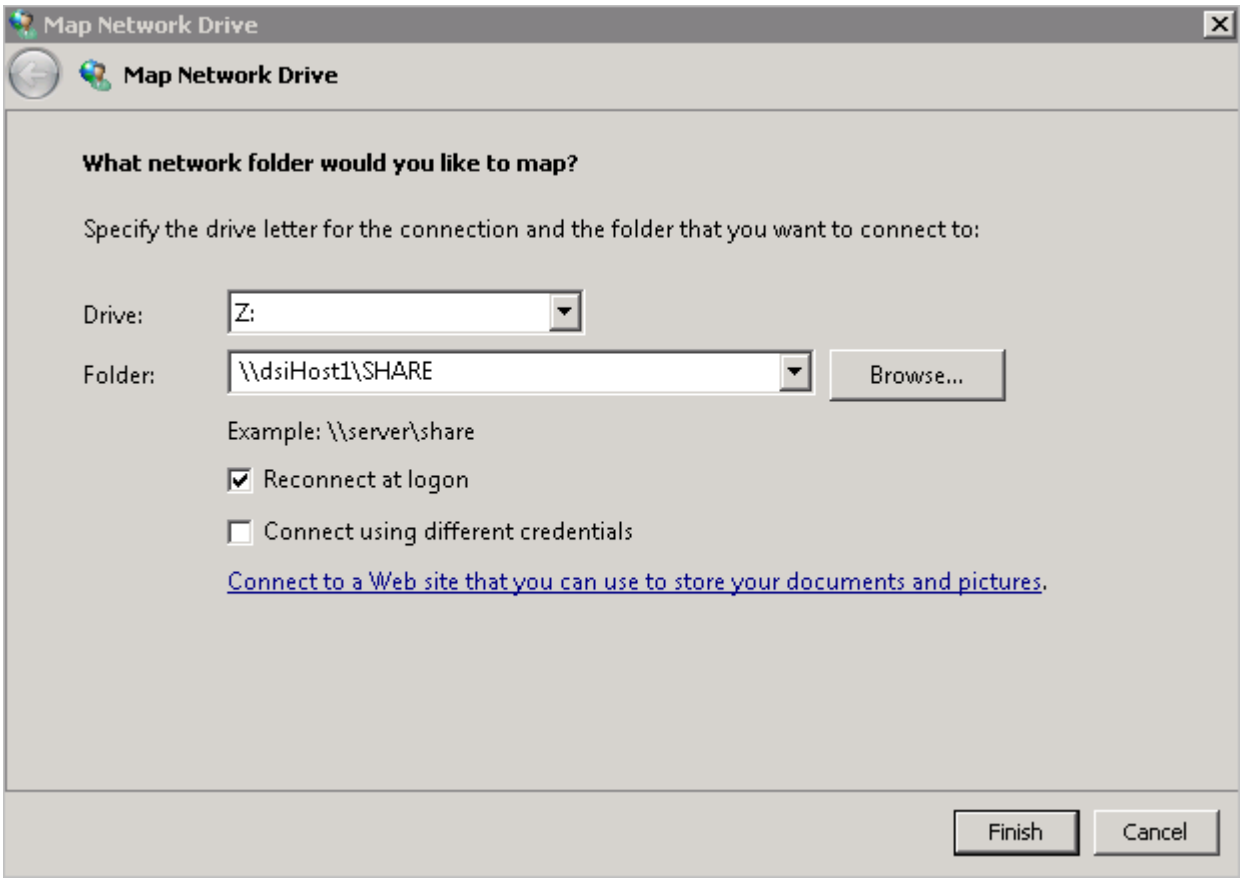
__ a. On the desktop of this host, double-click **Computer**.

__ b. Click **Map network drive**.



__ c. In the **Drive** field, you can choose a drive or keep the default drive.

__ d. In the **Folder** field, type: `\\dsiHost1\SHARE`



__ e. Click **Finish** and close Windows Explorer.

__ 3. Go to container2 (or the name of your “container 2” host) and repeat Step 2 on that host.

__ 4. Go to container3 (or the name of your “container 3” host) and repeat Step 2 on that host.

Appendix C. Troubleshooting issues

This appendix provides some suggestions to resolve issues that you might encounter during the exercises.

Container servers are not responding

If you run commands, such as deployment commands, that call your container hosts but the commands fail, you might need to verify that the servers are running.

- To check whether the catalogs and containers are communicating, switch to the `C:\IBM\ODMInsights88\runtime\wlp\bin` directory and run the `listHosts` command:

  ```
  xscmd -c listHosts -cep localhost:2810
  ```

- If only 1 or 2 of your container hosts are listed by the `xscmd` command, take note of the IP addresses for the hosts that are listed to determine which hosts are not listed.
- Check that the container not listed is running by using this command.

  ```
  serverManager isonline --propertiesFile=../etc/connectionC<Number>.properties
  ```

- If the `isonline` command fails to access the server, go to the host for that container to restart the server.
- If the container is running but was not listed, stop and restart the container server.

The catalog service is not available

- If this command fails because the catalog service is not available, check that each of your catalog servers are running by using this command.

  ```
  xscmd -c showQuorumStatus -cep localhost:2810
  ```

- If all the catalog servers are running, but not all are not listed when you run the `showQuorumStatus` command, stop and restart the catalog server that is not listed. Because quorum is enabled, it might be necessary to stop and restart all the catalogs together.
- If you must stop all the catalogs, follow the shutdown sequence for the grid as described in "Shutting down the grid" on page C-2.

Unable to stop a server

- If a server hangs or times out when you run the `server stop` command:
  - Reboot the VMware image.
  - Start the server.

The deployed solution is not responding to REST requests or HttpRequester

If you experience performance issues, such as not being able to use the REST API to access deployed solutions on your container hosts, or failure messages when you try to test the solution on the grid, you might need to redeploy your solution.

- To undeploy a solution from container servers:, follow the instructions in Exercise 16, "Administering Decision Server Insights", Section 6, "Undeploying solutions," on page 16-15.
  - From your main host (dsiHost1), stop and undeploy the solution on ***each*** of the containers. See Section 6.1, "Stopping and undeploying solutions".
  - On ***each*** container host, follow the instructions from Section 6.2, "Deleting solution files":
    - Stop the server.
    - Delete the solution files.
    - Restart the server with the `--clean` option.
  - From your main host (dsiHost1), redeploy the solution to each of the containers. See Section 6.3, "Redeploying the solution archive (.esa)".

If you also need to undeploy and redeploy connectivity, follow the steps here to undeploy.

- To undeploy connectivity:

  __ a. In a command prompt, switch to the `C:\IBM\ODMInsights88\runtime\ia\bin` directory.

  __ b. For the inbound server, type:

  ```
  connectivity Manager undeploy local banking_scenario_solution
  --propertiesFile=../etc/connectionIn1.properties
  ```

  __ c. For the outbound server, type:

  ```
  connectivity Manager undeploy local banking_scenario_solution
  --propertiesFile=../etc/connectionOut1.properties
  ```

- To regenerate and deploy connectivity configurations, follow the steps from Section 4, "Generating connectivity configurations," on page 12-8 and Section 3, "Deploying connectivity," on page 15-8.

Shutting down the grid

You might need to shut down the grid for various reasons, such as facing performance issues or an error in the grid or server configuration.

Shut down the grid by stopping the servers in this order:

- cisInbound1
- cisOutbound1
- cisContainer1, cisContainer2, cisContainer3
- cisCatalog1, cisCatalog2, cisCatalog3 (concurrently)

Restart the servers in the reverse order.

- cisCatalog1, cisCatalog2, cisCatalog3 (concurrently)
- cisContainer1, cisContainer2, cisContainer3
- cisOutbound1
- cisInbound1

IBM Training